Machine Learning Algorithms for Crime Prevention and Predictive Policing

Cardiff University School of Mathematics

Isabelle Williams

December 17, 2018

A Thesis Submitted for the Degree of Doctor of Philosophy

Abstract

Recent developments within the field of Machine Learning have given rise to the possibility of deploying these algorithms within a live policing environment. This thesis, motivated by the needs of Dyfed-Powys Police, focuses on developing a series of predictive tools that can be used directly within a live setting in order to improve efficiency across the force.

With an area of coverage that spans four socioeconomically diverse yet sparsely populated counties, Dyfed-Powys Police face a unique set of challenges in managing an increasingly limited set of resources such that offenders can be properly managed. The issue of personnel management is first addressed in the construction of a recommender system, which investigates the use of clustering techniques to exploit a stable pattern in the times at which crimes occur in various locations across the region. This is then followed with the development of a Recurrent Neural Network, which aims to predict the time to next offence within a particular narrowly-defined partition of the area.

By developing a series of tools that make use of existing data to predict which offenders within their database are most likely to reoffend, we aim to assist Dyfed-Powys in monitoring and preventing recidivism across the area. Firstly, we investigate the use of Random Forests and XGBoost algorithms, as well as Feedforward Neural Networks to predict an offender's likelihood of reoffence from a series of diverse factors. Secondly, we develop the aforementioned Random Forests algorithm into a survival model that aims to predict an offender's time to reoffence. Lastly, we develop a stacked model, which uses publicly available data to construct an Area Classification score for use as a factor within the original reoffence classification model. Insightful results are obtained, indicating a clear case for the use of many of

these techniques in a live setting.

Dedication

To Elle Woods, who taught me that like the rules of haircare, the rules of Mathematics are simple and finite. Any Cosmo girl would know!

Acknowledgements

So much changed throughout the 4 years it took to complete this thesis that it's almost impossible to thank every single person who contributed to it. Nevertheless, I'm going to try. To the staff at Dyfed-Powys police, I thank you for supporting this project and making this possible. To those at Cardiff who chose to show kindness when kindness was perhaps unwarranted, I thank you for your open-mindedness and generosity. To the staff at ERS, especially my managers Tom and Justin, I thank you for your infinite patience while mine was frequently tested. To my family, who I can't always visit as often as I'd like, I thank you for being my support to keep going even in the toughest of circumstances.

As for my husband, Aled - I don't think any amount of thanks could ever do you justice. You followed me to London, you watched me struggle to take on a full-time job, Actuarial Exams and a PhD all at once and you supported me every single day. Even when everything seemed lost, you never lost faith in me. If there's one thing that I'll keep (and treasure!) from this time of my life, it's you - and if you ever read this, now is a good time to get some tissues, because I know you're definitely crying!

Although there were many people who made significant contributions to my completion of this thesis, there were also many people along the way who didn't. Some may say that these people should go unacknowledged, but as all of us analyst-types know, sometimes the most negative experiences can lead to a positive outcome. To these people, I only have one thing to say.

In the speech I made at my wedding, I quoted Miley Cyrus. So to be in keeping with this theme within the acknowledgements of my final academic achievement, I think I'll go with Ariana.

Thank you, next.

Contents

Chapter 1

Introduction

UK police departments nationwide are facing budget freezes and deep cuts, precipitating the need for them to manage their resources more effectively while still responding to public demand for crime prevention and reduction. With financial pressure restricting the number of officers on the ground, as well as the number of officers who are available to monitor the progress of offenders post-offence, the need to produce predictive models that can assist in the allocation of these limited resources is greater than ever. For Dyfed-Powys police, a force covering one of the most sparsely populated but geographically large areas in the UK, directing their resources to the most appropriate individuals is particularly crucial. Should their officers focus their resources on the wrong offenders or the wrong locations, the large geographical area of coverage makes these mistakes far more costly to the force than they would be in a geographically smaller, more densely populated area. In particular, it is extremely important that officers are located in the right place at the right time. With little focus having been placed on applying predictive policing techniques to a rural location, or otherwise investigating the patterns of crime within an area of such great socioeconomic diversity, it cannot be assumed that current techniques as used in urban locales are necessarily applicable to an area like Dyfed-Powys. As such, it is of particular importance to this force, as well as other rural forces in the international community, that appropriate solutions to the management of their increasingly limited human resources can be found.

The primary purpose of this research is to deliver a series of predictive tools to Dyfed-Powys police, which will enable them to better manage their increasingly

limited resources. In order to achieve this objective, however, it first needs to be decided exactly which challenges we want these tools to assist with. Following input from officers in many different areas of Dyfed-Powys police, it was decided that this thesis would aim to develop tools to answer two main questions. Chapters 2 and 3 of this thesis will focus on developing a series of tools that make use of existing data to predict which offenders within their database are most likely to reoffend, while Chapter 4 will focus on the development of tools using the same data to predict where and when these offences are likely to occur. Chapter 5 will then begin to develop these further, using recent techniques and stacked models in order to provide greater insight into some of the issues faced by Dyfed-Powys police. The tools developed to answer these questions will then be brought together in the conclusion, alongside a brief series of recommendations for some possible future investigations.

In order to further explain the nature of these individual problems, we will now provide a brief introduction and rationale to the problems to be tackled in each of the four chapters of this thesis, beginning with those to be tackled in Chapters 2 and 3.

1.1 Chapters 2 and 3 - Reoffender Behaviour

When considering who is likely to reoffend within the Dyfed-Powys area, a good place to begin is to find an algorithm, or series of algorithms, that can accurately predict the future behaviour of offenders within a police database. The subject of using machine learning algorithms to predict the likelihood of recidivism following an offence, in particular, has been the subject of many articles [11, 87] and more recently, theses from both a criminology [38] and a machine learning perspective [61]. Although the efficacy of at least some of these systems versus the professional judgement of officers has been called into question [27] and it is possible that these tools may lead to biased predictions (which can lead to issues from an ethical standpoint) [21], it is still an area that is well worth looking into for the positive impact these instruments can provide for reoffences following both violent and non-violent crimes [10]. By knowing which of the offenders currently within the system are most likely to reoffend, and when these reoffences are likely to be committed, as well as the factors that are most likely to affect the offender's recidivism risk, resources can

be redirected to the individuals most likely to continue on a criminal path and away from those unlikely to cause any further issues for the police.

Being able to predict the individuals most likely to reoffend also offers further benefits to other services within the justice system as a whole. Probation services, which focus on working with and monitoring offenders to prevent further reoffences, could also benefit from the use of predictive tools like these. In the UK, where our research is based, the system currently in place to predict an offender's likelihood of recidivism based on past offence data is known as the Offender Group Reconviction Scale (OGRS) [42]. This system is currently on its third iteration, utilising a logistic regression model to predict the likelihood of recidivism within one or two years of an offence. While for practical reasons this model is based on a limited number of possible predictive factors, the current systems do not make best use of the full spectrum of data available within the police system, nor do they attempt to incorporate any sort of freely available demographic data into the models. Moreover, the current OGRS system, OGRS 3, makes no attempt to predict the likely "survival" of an offender in society once an offence has been committed by that individual. As such, there is a clear case to be made that current methods can be improved upon and made more useful for both Dyfed-Powys and forces in the wider community. While the focus of this thesis will be to predict the future offence behaviour of offenders currently registered in the Dyfed-Powys database, the methods investigated here can quite easily be used as reference for further predictive policing work in other forces.

To go beyond the scope of the systems currently in place in the UK, however, further external data will need to be incorporated into our modelling data. This external data will be comprised of data obtained from the following three sources:

1. An external dataset collected by statswales, the Welsh Index of Multiple Deprivation (WIMD) [70]. This contains several indicators of deprivation, many of which have been thought to make a significant contribution to the incidence of criminal behaviour.

2. The Cambridge Crime Harm Index [79], a recently developed index that aims to

describe the level of harm inflicted on society by an individual crime, will also be made use of within this thesis. This index has been used to great effect in many other locations in the UK and as such, it is widely believed that it may be useful as a factor predicting the incidence of crime.

3. An Urban-Rural Index [25], which describes how urban or rural a particular location is considered to be in addition to the general sparsity of the area it sits in.

These additional data items will help to enrich the data utilised by Dyfed-Powys police. The exact nature of the data to be included from each of these datasets, as well as how they will be joined to the dataset given by Dyfed-Powys, will be further discussed in Chapter 2, Section 2.

In order to keep this research as current and as easily deployable as possible, two tree-based machine learning techniques have been employed, Random Forest [15] and XGBoost [18]. These two machine learning techniques, both chosen for their lack of dependence on specific probability distributions as well as their ability to handle a large number of diverse factors, are then compared against a Feedfoward Neural Network [7]. In all three cases, predictions of recidivism will be produced on a crime by crime basis, with each individual crime that an offender commits leading to a prediction. This thesis will use a very recent real-world police dataset, comprising many different types of offences and offenders, who have not necessarily been subject to either arraignment or incarceration. The details of how these three methods will be employed and their comparative efficacy as predictors of reoffender behaviour will be further discussed in Chapter 2, Section 3.

Employing an extension of the Random Forests algorithm for survival data [43], an estimate of an offender's survival distribution will be produced with the aim of using this to predict the likely time to their next offence. While this model has been previously utilised in a predictive policing context, it has not been used in the context of a full police dataset before, nor has it been used to predict the likely outcome of crimes for which the offenders have not been subject to arraignment or incarceration. The details as to how this algorithm has been developed for use in a survival context and how it will be implemented in this case will be discussed in

1.2 Chapter 4 - Spatio-Temporal Crime Patterns

Moving our focus from general policing issues to those more specific to the Dyfed-Powys area, the second aim of this thesis will be to provide a solution to some of the daily challenges often experienced by the acting officers in that area. A particular challenge encountered by rural police forces like Dyfed-Powys is simply managing to effectively police a large, diverse geographical area. Dyfed-Powys police itself is responsible for the largest territory in England and Wales, comprising four different counties and over 350 miles of coastline. The infrastructure in the area is often relatively poor, with winding country roads and A-Roads of varying quality often being the only way to access many of the towns within their jurisdiction. With a sparse population, mainly clustered in small rural towns scattered across the four counties, it is nearly impossible for their officers to be "everywhere at once". As such, in a rural area like Dyfed-Powys, it is important that it is known where a crime is likely to occur. Committing too many of the force's resources to one location, when in fact crime is occurring in quite another, is a significant issue in an area where journey times between those locations are likely to be long. By producing recommendations as to where their resources are most needed, resources can be directed to the most appropriate locations within Dyfed-Powys, saving the force both time and money. Additionally, the presence of police officers in the correct location may well act as a deterrent to, or bring a quick halt to, criminal activity in that location.

When attempting to predict the timing of future crimes in a particular location, a good starting point would be to look at the distribution of past crimes within that location. However, this constraint omits potentially relevant information from other locations, which may influence the timing of crimes in a measurable way. Using the mathematical similarity between crimes committed in defined locations within the Dyfed-Powys area, a Collaborative Filtering-based [35] Recommender System [71] that clusters locations according to their similarity to all other locations within the area will be constructed. While Recommender Systems have not been utilised in this context before, there is significant evidence (to be discussed in Chapter 4, Section 1) that these systems may well be suitable for this kind of task. The practical uses

of this system, as well as the suitability of the technique and the reasons for its use in our dataset, will be further discussed in Chapter 4.

1.3 Chapter 5 - Further Research

Following the investigation into Recommender Systems and the relative similarity of locations, a further investigation was launched with the aim of investigating whether Recurrent Neural Networks (RNNs) can be designed to predict the time until next crime based solely on the distribution of past crimes within that location. These predictions will be based on a series of time differences between one crime and the next in an area and will attempt, without any basis in probability distributions or influence from outside factors, to predict the likely time to next offence in each of the individual towns in the Dyfed-Powys area. The application of this type of Neural Network has only rarely been tested in a crime setting [59, 2]. The details of this network, as well as how these towns and therefore sequences are to be defined, will be further discussed in Chapter 5, Section 1.

Returning to the reoffending problems first discussed in Chapters 2 and 3, the large number of highly-correlated features within the WIMD dataset will be replaced with a single risk-based score contingent on either the area in which the offender is resident or the area in which the offence was committed. This score, which will be based on the aggregate count of crimes by Lower Super Output Area (LSOA), a partitioning method based on equality of populations between each individual location, is intended to be a score that will represent the overall risk of offence per LSOA. In principle, this overall risk of offence should be representative of the risk of an individual crime leading to a reoffence. The details of this score, as well as its effectiveness as a predictive factor within the reoffence model, will be further discussed in Chapter 5, Section 2.

With the main aim of this research being the delivery of working algorithms to Dyfed-Powys police, a demonstration of their ability to run within in a live environment and their ability to perform acceptably within that context must be shown. As such, within Chapters 2-4, a series of results from a live test context will be included, representing the performance of the models implemented within Dyfed-

Powys police. Due to the more exploratory nature of Chapter 5, live test results will not be included, but suggestions pertaining to their potential future use within Dyfed-Powys will be included.

1.4 Publications and Achievements

Two main contributions have been made to the scientific and policing communities as a result of this research. The first formed the basis of a patent as well as a paper, and the second formed the basis of two deployable models to be utilised in a policing setting.

A paper, entitled Text Mining and Recommender Systems for Predictive Policing[66], was presented at and was published in the Proceedings of the ACM Symposium on Document Engineering 2018. The work presented in this paper formed the basis of Chapter 4 of this thesis and included an evaluation of several measures of similarity.

A patent has also been applied for, entitled Crime Similarity Areas for Crime Analysis and Prediction, based on the same work as the above paper. This is now pending with HP Labs.

Following the investigations within Chapters 2 and 3, two models have been supplied to Dyfed-Powys. These are a reoffence classification model and a reoffence survival model, of which both have been planned for deployment within the force. Both of these models make use of the Random Forests algorithm and are designed to underpin a new diversionary scheme, which is intended to reduce the incidence of reoffending within Dyfed-Powys. A deployable version of the Chapter 4 has also been delivered to Dyfed-Powys, but at the time of writing, its use has not yet been approved by the force.

Chapter 2

Machine Learning Algorithms for Reoffence Prediction

2.1 Risk of Recidivism

The rate of recidivism within a population, or the act of re-engaging in criminal activity despite having been punished, can be seen as a performance measure of the effectiveness of offender management strategies [60]. Although a lower recidivism rate is not necessarily caused by better offender management, as many other socioeconomic factors must be taken into account when evaluating the effectiveness of such a strategy, it is crucial that police forces learn to strike a balance between the monetary costs of various offender management strategies and their potential benefits to society in terms of reducing recidivism rates. One way in which mathematics can assist police forces to find this balance is to provide a way in which, from readily available data, an offender's risk of recidivism can be predicted. The models used to predict this risk of recidivism, as well as their comparative performance, will be the focus of this chapter. As previously mentioned in the introduction, this problem will be framed as one of binary classification, where the model's output will be a prediction as to whether or not an offence will lead to a reoffence from the same offender within three years of the offence being reported. The reasons for this decision, as well as a brief discussion of previous work completed within this area of research, will be outlined in this section.

To maximise the amount of data available for prediction, particularly data per-

taining to early offender behaviour, predictions of recidivism will be produced on a crime-by-crime basis. This means that although some details of an offender's criminal history will be carried over between crimes, each crime committed by an individual offender throughout their criminal career will be considered separately for prediction. This is due to the fact that we can consider the individual to be a sufficiently different individual at the point at which they committed each crime. It is therefore unnecessary to further complicate the model by introducing the concept of dependency on an individual offender.

Although technically the outcome of each crime could be considered to be dependent on the offender in question, it can be argued that when time-dependent information is included such as the number of previous offences committed by the offender, the time since their last offence and the details of any location changes that occurred between the offences, the offender can be considered to be a sufficiently different individual at the point at which they committed each crime that it is not necessary to further complicate the model by introducing the concept of dependency on the individual offender into the model.

2.1.1 Related Work

Since finding working solutions to the problem of recurrent recidivism is critical in today's economic climate, a great deal of research into the use of mathematics for the prediction of recidivism has already been undertaken in this field from a criminology perspective. These studies often focus on predicting the recidivism risk of a group of offenders considered to pose considerable risk to society, with many studies focusing on predicting the likelihood of recidivism for dangerously violent [58] or mentally ill individuals [54]. Some studies focus on investigating the effect of particular factors on recidivism, such as the level of poverty experienced by the offender [49], while others take a broader view, testing the predictive power of several factors on the likelihood of recidivism [9].

Many studies focus on predicting the likelihood of recidivism for offenders that have already been arrested and subjected to some kind of risk assessment tool [38, 28], often based on the results of a Logistic Regression [22] model. While this model is often seen as the "traditional" or "industry standard" method for predicting an of-

fender's risk of recidivism in the UK [42], in recent years, machine learning methods such as Random Forests [12, 67] have overtaken the traditional Logistic Regression approach in both popularity and predictive performance on criminal datasets. The history of using Neural Networks for this sort of predictive task, however, is a little more variable. In some earlier studies, it is suggested that Neural Networks do not offer any advantages over other traditional Machine Learning algorithms for these purposes [17], while other early studies suggest that there may be some benefit [62] to using Neural Networks for these purposes.

With this particular area of investigation comprising only half of the material to be covered in this thesis, it was essential that the number of algorithms tested was narrowed down to a limited number of appropriate, potentially deployable choices. In many recent studies, machine learning methods have formed the main basis of investigative work into Reoffence prediction. While in some comparisons, including one conducted from a statistical perspective [87], it is suggested that machine learning methods do not offer any comparative advantage in terms of model performance over classical methods. However, in most comparative studies, this is not seen to be the case. In fact, these methods often are more effective, simpler to build and deploy in a timely manner and also more accurate than traditional statistical methods.

In one study, the comparative efficacy of the Random Forests algorithm alongside other Machine Learning algorithms was tested on a dataset of violent offenders with a low rate of recidivism [11]. Here, the Random Forests algorithm was shown to provide a better standard of performance when compared to the conventional Logistic Regression model, as well as the less conventional Gradient Boosting approach. A second study, also conducted from a criminology perspective on a prison dataset [52], tested many machine learning algorithms, including Neural Networks. While Neural Networks offered the best performance of all models tested, they did not offer a large predictive advantage over Logistic Regression or CART-based models.

The most comprehensive recent study takes the form of a thesis conducted within the criminology department at the University of Texas [61]. In this study, which uses the Bureau of Justice Statistics' 1994 dataset on the recidivism of prisoners released from custody, the conventional Logistic Regression model was compared

against several machine learning methods, including Random Forests, Support Vector Machines, XGBoost, Neural Networks and the Search algorithm. In that case, XGBoost and Neural Networks were shown to be the best performing models, outranking Random Forests.

2.1.2 Applications of Current Research

Several conclusions can be drawn from the current research. Firstly, a lot of the research in this area, particularly the research that focuses on the use of machine learning algorithms, is designed to only predict the recidivism risk of offenders whose offences have caused great harm to society. In general, few academic investigations have recently been made into utilising machine learning algorithms for a more general purpose in typical police datasets comprising several types of crime, nor have they been made for a dataset including criminals who have been arrested but not necessarily subjected to arraignment or incarceration. While the results from the focused datasets that are often the subject of recidivism studies are useful, particularly in high population density areas with large amounts of serious crime, for the purpose of this thesis, focusing only on offences and the subsequent reoffences committed by certain sections of the population will cause the model to miss out on large amounts of potentially useful data. In many rural areas, including the area covered by Dyfed-Powys police, the relatively low number of serious offences means that the police will often prefer to focus on detecting and dealing with the large number of petty or "nuisance" crimes that will not necessarily lead to a prison sentence or even a court summons. As such, the decision has been taken to widen our focus, looking instead at predicting the outcome of a full spectrum of crimes recorded by Dyfed-Powys police. In this way, as soon as any new crime has been reported in the Dyfed-Powys area, the likelihood that each offender involved with the crime will continue to offend can be immediately predicted.

An issue frequently faced in these binary classification problems is knowing where to set a time limit for further criminal activity. As the dataset given by Dyfed-Powys police covers a limited time span, it is not possible to actually predict whether or not an offence will *ultimately* lead to a reoffence. It is possible, however, to predict whether or not an offence will lead to a reoffence within a reasonable time period. In

the context of this problem, this can be considered to be a further hyperparameter that should be adjusted in our testing period. In the context of this thesis, however, a single time limit will be set. This is so that work duplication with survival forests (which will attempt to predict the offender's time to next offence) to be discussed Chapter 3 can be avoided.

While the OGRS system, as well as many other systems, have set their limits to 1 and 2 years, this time period has been chosen to be 3 years, a time period by which approximately 93% of the reoffences to be committed within the relevant time period have already been committed. As such, this is very close to an "ultimate measure of reoffence". Moreover, from the perspective of trying to produce a balanced input dataset of reoffences/non-reoffences, which is required for many machine learning algorithms, setting this time period to 3 years (rather than 1 or 2 years) makes things computationally much simpler. While this does make 3 years of data unavailable for training, the subset considered for this purpose still contains over 30000 records, more than enough to generate a cohesive model.

2.2 The Dataset

The dataset that will be used to predict an offender's risk of recidivism is an amalgamation of information from two separate data sources. The first source of data is an anonymised dataset from Dyfed-Powys police, which gives a full picture of all crimes reported to the force between 1/1/2008 and 31/12/2014. The details of these offences, which are listed by victim according to UK guidelines, are joined by some demographic details pertaining to the offender and victim of the crime. A small example of some fields within this dataset prior to pre-processing is included in Table 2.1 below.

Table 2.1: Dyfed-Powys Dataset Example. See Appendix for descriptions and explanations of independent variables.

ID	AgeCommitted	BurgValue	Fine	MDAClass	MultipleOffences
1	31	160	2	U	0
2	26	0	0	B	0
3	33	1000	0	U	0
4	46	0	1	U	0
5	32	512	2	U	0
6	32	0	2	U	0
7	30	0	0	B	0
8	40	28000	2	U	0
9	35	0	2	U	0

The second data source, the demographic factors behind the Welsh Index of Multiple Deprivation [70], add location-based information to these demographic details based on the place in which the crime occurs and the place in which the offender is resident. Adding data from this source, which is is publicly available on the Welsh Government website and was last updated in 2014, allows us to look beyond what is currently readily available to UK police forces. It also allows us to gain insight into whether or not the demographics of the area in which a crime is committed, or in which an offender is resident, affect the risk of recidivism. The investigation of such information could potentially have many social implications, whether or not demographic factors used as indicators of deprivation are shown to be effective predictors of recidivism risk.

This dataset is keyed on Local Authority area, which can be matched to the Dyfed-Powys dataset by postcode. Examples of some fields within this dataset prior to pre-processing, but after joining to the Dyfed-Powys dataset, are included in Tables 2.2 and 2.3 below.

Table 2.2: WIMD Dataset Example (Joined to Crime Postcodes)

ID	Crime_EmpBenefits_Perc	Crime_EmpLTSick_per100000
1	25	31772.0
2	10	23494.6
3	6	16145.4
4	7	15631.6
5	10	23494.6
6	10	23494.6
7	21	27915.4
8	10	18897.3
9	20	21385.1

Table 2.3: WIMD Dataset Example (Joined to Offender Postcodes)

ID	Off_EducNotInHE_Perc	Off_EducNoQuals_Perc	Off_Burglary_per100
1	80	26.91	2.53
2	87	35.08	0.68
3	69	19.36	1.43
4	59	14.61	0.53
5	69	19.36	1.43
6	69	19.36	1.43
7	64	21.46	1.43
8	80	24.27	1.43
9	69	19.36	1.43

While making use of the first dataset could lead to valuable insights from an academic perspective, the results of this analysis (including the models produced) are unlikely to be helpful to Dyfed-Powys police if this historical data is not representative of their current dataset. Moreover, with an improvement in Dyfed-Powys data capture processes having been put into effect between 2014 and the current date, several of the issues with missing location fields are no longer present in this dataset, meaning that it is possible that many of the variables that rely on an accurate crime or offender location (including the WIMD variables and the distance between the offender's location and the location of the crime) will be more likely to take a higher level of importance in the model. As such, our third dataset will be a further anonymised dataset from Dyfed-Powys police, which gives a full picture of all crimes reported to the force between 1/1/2011 and 31/12/2017. This will be known as the "live test" data. As Dyfed-Powys have not made any additional data available for capture between the original and the "live test" dataset, the form of this dataset will be identical to the first.

2.2.1 Dependent Variable

The Reoffence Variable Keeping in common with most previous studies (discounting those that defined a reoffence as a return to prison), a reoffence will be defined to be to be a further offence committed by the same offender within three years of the most recent reported offence date. Since this is a binary classification problem, the dependent variable $Reoffend = 0$ when an offence does not lead to a reoffence and $Reoffend = 1$ when an offence does lead to a reoffence. For the following reasons, however, not all further offences reported within a three year period will be considered to be a reoffence.

What is Considered to be a Reoffence? The two variables within our dataset that will be used to define whether an offence is considered to be a reoffence are *TimeSincePrevious*, defined as the number of days since the offender's most recent offence, and*NoPreviousArrests*. The former variable is based on the latter, with an offence only contributing to the *TimeSincePrevious* variable if it adds to *NoPreviousArrests*. In this thesis, a "reoffence" will correspond to a record for which the variable *NoPreviousArrests* ¿ 0 and the variable *TimeSincePrevious* $\leq$ 1095. Examples of this are given in Table 2.4:

Table 2.4: No. Previous Arrests: Examples

Record	Offender ID	TimeSincePrevious	NoPreviousArrests	Reoffence?
1	A	1090	1	Yes
2	B	5	0	Yes
3	B	105	1	Yes
4	B	2015	2	No
5	C	0	0	No
6	C	1250	0	No

For offender B, their 1st and 2nd offences will be counted as reoffences, but their 3rd offence will not be - this is because the 3rd offence was committed 2015 days after the 2nd one and so will be considered to be an independent offence - the No. of Previous Arrests undergone by the offender will still be accounted for, however.

To clarify, *NoPreviousArrests*, though not a complete misnomer, *does not strictly stand for the actual number of times the offender has been arrested*; this variable is an estimate, but not an actual tally, of the offender's number of previous arrests.

Specifically, it is the number of previous crimes that the offender has committed, discounting any offences that were reported on the same date from the total. As such, an individual with two crimes reported on the same date but no previous offences will not be considered to have committed a reoffence with the second crime. From a policing perspective, this treatment of reoffences is much more intuitive; if an arrest is made for more than one offence, it can be very difficult or even impossible to determine which of the offences is the "original offence" and therefore, which of the offences can be considered to be "reoffences".

Which Offences Don't Count Towards the Reoffence Total? To avoid confusion and complications with the decision as to whether or not a particular type of offence really "counts" as a reoffence, offences that fall into the following categories will be excluded from the dataset:

1. Offences that are reported as TIC, or "Taken Into Consideration". These offences have been excluded due to the uncertain nature of their reporting and treatment by the justice system. The fact that these offences can be reported some time before the offender is brought to justice also complicates the calculation of how many offences the offender can actually be considered to have committed. While the records (dataset rows) containing TIC offences will be removed from the dataset, they will count towards the total number of arrests if they were reported at different times.

2. Offences that have not lead to any kind of prosecution by the police. This lack of prosecution can occur for many reasons, which include the decision by either the CPS or acting officer that prosecution would not be in the interest of the public, or the death or serious illness of the offender involved. Since the police do not consider these entries to be offences, they will not be considered to contribute to an offender's offence history. These offences will be removed for prediction and will not count towards an offender's offence history in any way.

The precise practical difference between 1 and 2 can be illustrated in the following way:

1. Suppose an offender is caught for an offence and brought in for questioning.

While under questioning, the offender is asked whether or not they have committed any other offences (perhaps because the police are suspicious that they may have been involved, or may have some evidence to that effect). If they do admit to further offences, the outcome of the offence may be TIC, or "Taken Into Consideration".

2. Suppose that the same offender is brought in for the same offence. While under questioning, the police decide that it is not worth prosecuting the offender - perhaps they decide that the reported offence is not actually an offence, or the individual who reported the offence decides not to prosecute. In this case, the outcome of the offence will be registered as "Not A Crime" or "Not Prosecuted"

Now that what is considered to be a reoffence has been defined, the nature of the independent variables that will be used to predict the reoffence outcome will be discussed.

2.2.2 Independent Variables

In this analysis, it was decided to investigate the maximum number of available variables for prediction. Although previous studies do give some idea of the likely most effective predictors of recidivism, including the age and sex of the offender [9] and the sanction imposed on their most recent offence [42], it is unknown if there may be other factors, of which many have not been considered by these studies, that will have a large effect an offender's risk of recidivism.

The independent variables that have been selected and their properties, which have been extracted both from information already routinely collected by Dyfed-Powys police and the Welsh Index of Multiple Deprivation, are fully detailed in Table 1 of the Appendix. Here, some of the less obviously definable variables within this dataset will briefly be outlined.

OffenceCode

Firstly, the types of offence (OffenceCat) that individual offences are categorised by will be outlined. In the Dyfed-Powys police dataset, the offence is classified by a variable known as OffenceCode, which details the specific type of offence that

has been committed under UK law. However, since OffenceCode contains over 50 different categories, these will be re-mapped into categories based on higher-level crime classifications utilised across the UK police force. These categories are outlined in Table 2.5 below.

Table 2.5: Offence Types as Used to Categorise Crime

OffenceCat	Description of Offence
BurgDW	Burglary Committed in a Dwelling
BurgNDW	Burglary Not Committed in a Dwelling
CrimDam	Criminal Damage
DrugPoss	Drug Offence (Possession)
DrugOth	Drug Offence (Other)
Misc	Miscellaneous Offence
Sex	Sex Offence
Theft	Theft
Violence	Violent Offence
Weapon	Weapon-Related Offence

Although the official guidelines sometimes separate violent offences into two different categories depending on the level of injury caused to the victim, it has been chosen to include any offences related to violence in the same category. This is due to the fact that considering the level of harm the offence presents to society (and therefore, in this case, the severity of the injury to the person) makes the distinction between violent offences with and without injury to the person largely redundant. Since many of the public order offences also involve violence or at the very least harassment, this category has been merged into the violent offence category, with (once again) the level of harm to society done by such an offence to be determined by the variables that are generated by the crime harm values as defined by the Cambridge Crime Harm Index.

Cambridge Crime Harm Index

Level of Harm to Society Following recent research in the field [79], the Cambridge Crime Harm Index will be considered as a possible method of measuring the severity of an offence from the perspective of the level of harm it likely brings to society. This method, which assigns a number to a type of crime based on the likely minimum sentence or "starting point" for that particular crime, will provide a different perspective on crime classification grounded in the likely level of harm that

the crime presents to society. While this index does not cover all crimes recorded in this dataset as it excludes crime that is recorded due to random detection by police or security officers (i.e. drug arrests, traffic arrests, some cases of shoplifting), it is worth investigating whether or not a slightly modified version of this index would be a useful predictor of an individual's tendency to reoffend.

The harm index with corresponding number of sentencing days attached, as it appears in the original paper, is detailed in Table 2.6 below:

Table 2.6: Cambridge Crime Harm Index [79]

Crime Type	Subtype	Starting Point Sentence Days
Homicide		5475
GBH	Intent	1460
ABH		20
Assault		1
Rape		1825
Sexual Assault		365
Robbery		365
Burglary	Dwelling	20
Burglary	Non-Dwelling	20
Vehicle	Theft of	20
Vehicle	Theft from	2
Theft	from Person	20
Theft	from Shop	2
Theft	from Other	2
Criminal Damage	Arson	33
Criminal Damage	Other	2
Fraud		20

These values, which are attached to each individual crime, are stored in the database as either Fine or Sentence variables depending on whether the value corresponds to paying off a minimum fine or serving a minimum sentence.

In order to provide as full a picture as possible of the offender's history, if a criminal history is present, the total past harm that the individual has inflicted upon society will also be looked into. This variable, PrevFine or PrevSentence, will be represented by the sum of the Cambridge Crime Harm Index values (stored in Fine and Sentence) of their previous crimes. This will provide a different, more specific picture to that presented by the simple number of arrests that the individual has undergone and the number of days that have passed since their previous offence.

Amendments to Cambridge Crime Harm Index To better suit the purposes of this thesis, as not all of the crimes within our dataset fit exactly into these categories, a few minor amendments to the index have been made based on CPS sentencing guidelines to cater to this particular dataset. Please note that a more detailed breakdown of the harm index for more variably charged offences such as drug possession, sexual assaults and repeat burglaries will not be in scope for this thesis, though will likely be in scope for the further development of this model as it is routinely refreshed by the police.

Table 2.7: Additions and Amendments to Cambridge Crime Harm Index

Crime Type	Subtype	Min.Sentence Days
Kidnap		548
Death by Dangerous Driving		365
GBH	Without Intent	365
Drug Offences		0
Firearms Offences		1825
Uncategorised Offences With Victim		1
Uncategorised Victimless Offences		0

Urban Rural Classification

The final variable to be introduced is a variable that quantifies the rural/urban status of a location. In this case, it has been chosen to use the classification as defined for each census output area, matching these to the postcode sectors within each area to attain a rural-urban classification for each postcode sector. This system breaks down the rural/urban classification of areas into 10 different categories, some of which are dependent on the general population density of the surrounding area. Details of exactly how these classifications were arrived at can be found on the UK Government website [25]. The 10 categories that have been used in this analysis are outlined in Table 2.8 below:

Table 2.8: Urban Rural Classification Categories, 2011

Classification Description	Code
Urban Major Conurbation	Not in Dataset
Urban Minor Conurbation	Not in Dataset
Urban City and Town	1
Urban City and Town in a Sparse Setting	2
Rural Town and Fringe	3
Rural Town and Fringe in a Sparse Setting	4
Rural Village	5
Rural Village in a Sparse Setting	6
Rural Hamlets and Isolated Dwellings	7
Rural Hamlets and Isolated Dwellings in a Sparse Setting	8

As there are no conurbations within the Dyfed-Powys policing area, it has been chosen to simply number each of the different classifications with the numbers 1-8. These will not, however, be used as an ordinal scale; these are simply categories that have been designated integer numbers for convenience.

Other Data Items

Alcohol and Drug Related Offences An interesting issue that arises for certain variables within the dataset, such whether or not an offence is Alcohol or Drug Related, is that these variables are entirely based on the immediate observations of individual officers. As such, the decision as to whether or not an offence is Alcohol or Drug Related, as well as whether or not this statistic is even recorded, may be somewhat subjective. This does not mean, however, that such information will not be useful in the analysis; it simply means that this subjectivity must be borne in mind when including this information as a predictor of reoffence.

Offence Categorisation A similar issue also arises with the categorisation of offence types and the level of harm these offences are deemed to cause on society; although the possible categories of offence types have been set as per the official UK police categorisations and the level of harm as per the Cambridge Crime Harm Index with some minor amendments, this categorisation is still somewhat subjective and there may be more useful ways of categorising these offences in a predictive situation. For the purposes of this analysis, however, the outlines provided here will be sufficient.

Dimensionality Reduction

The principle of dimensionality reduction aims to exploit the redundancy of variables within input data in order to find a smaller set of new variables, which are made up of a combination of the set of the input variables fed into the technique [81]. This technique is particularly useful when attempting to fit a model to a dataset with a number of highly correlated variables.

While there are not a large number of variables to investigate in this case and it is therefore unlikely to be useful to investigate a large number of dimensionality reduction techniques for this particular dataset, there are a small number of highly correlated variables within the WIMD dataset that may benefit from the use of a dimensionality reduction technique. Moreover, making use of a smaller number of variables reduces model size, complexity and processing time, which are all concerns for Dyfed-Powys police - with limited computing power and space, reducing the size of the model will allow the police to refresh the model on a regular basis, allowing it to be kept up to date.

To illustrate the correlation between various variables in this dataset, all WIMD variables that have a correlation of at least 0.7 with another WIMD variable in the original dataset are listed in Table 2.9 below.

Table 2.9: Highly-Correlated WIMD Variables (Correlation > 0.7). See Appendix for descriptions and explanations of independent variables.

Factor	Max. Correlation	Max. Correlation Variable
Off_EmpBenefits_Perc	0.9370	Off_Income_Perc
Crime_EmpBenefits_Perc	0.9395	Crime_Income_Perc
Off_Income_Perc	0.9370	Off_EmpBenefits_Perc
Crime_Income_Perc	0.9395	Crime_EmpBenefits_Perc
Off_EducNoQuals_Perc	0.8761	Off_Income_Perc
Crime_EmpLTSick_per100000	0.8853	Crime_EmpBenefits_Perc
Crime_EducNoQuals_Perc	0.8707	Crime_Income_Perc
Off_EmpLTSick_per100000	0.8961	Off_EmpBenefits_Perc
Off_EducAbs_Perc	0.7739	Off_Income_Perc
Off_EducNotInHE_Perc	0.8296	Off_Income_Perc
Crime_EducAbs_Perc	0.7735	Crime_EmpBenefits_Perc
Off_EducKS4L2_Perc	0.8295	Off_EducKS4_Pts
Crime_EducKS4L2_Perc	0.7955	Crime_EducKS4_Pts
Off_ASB_per100	0.8704	Off_Violence_per100
Crime_ASB_per100	0.9140	Crime_Violence_per100
Off_CrimDam_per100	0.8750	Off_Violence_per100
Crime_CrimDam_per100	0.9326	Crime_Violence_per100
Crime_Fire_per100	0.7185	Crime_ASB_per100

In this case, it has been chosen to make use of one of the simplest and most widely-used dimensionality reduction techniques, Principal Components Analysis [64]. This technique was chosen for its explainability and wide use in many mathematical, statistical and data science contexts - this technique should be familiar to many members of staff within Dyfed-Powys police and should be simple to explain to higher-up staff members, if an explanation is requested.

PCA converts a set of observations of possibly correlated variables (in this case, a set of numerical variables relating to two potentially separate geographic areas, of which some are highly correlated) into a set of linearly uncorrelated variables known as Principal Components. These components are defined such that the first of these components accounts for as much of the variability in the data as possible. Successive components will then have the highest variance possible under the constraint that they are orthogonal to all preceding components.

Defining this input set of WIMD variables to be w, the first step is to look for a linear function $\alpha_1^T w$ of the elements of w with maximum variance. Here, α_1^T is a transposed vector of p constants $\alpha_{11}, \alpha_{12}, ..., \alpha_{1p}$, such that:

$$\alpha_1^T w = \alpha_{11} w_1 + \alpha_{12} w_2 + ... + \alpha_{1p} w_p = \sum_{j=1}^{p} \alpha_{1j} w_j \tag{2.1}$$

In successive steps for all $k > 1$, a linear function $\alpha_k^T w$ uncorrelated with $\alpha_1^T w, ..., \alpha_{k-1}^T w$ that has maximum variance will be found. The kth derived variable using this method, $\alpha_k^T w$, is the kth Principal Component [44].

Applying Principal Components Analysis to our set of input variables w, each Principal Component generated can be plotted against the cumulative proportion of variance explained by the successive components in order to determine how many components are necessary in order to explain a sufficient proportion of the variance. This plot of the analysis conducted on the WIMD features is included in Figure 2.1 below.

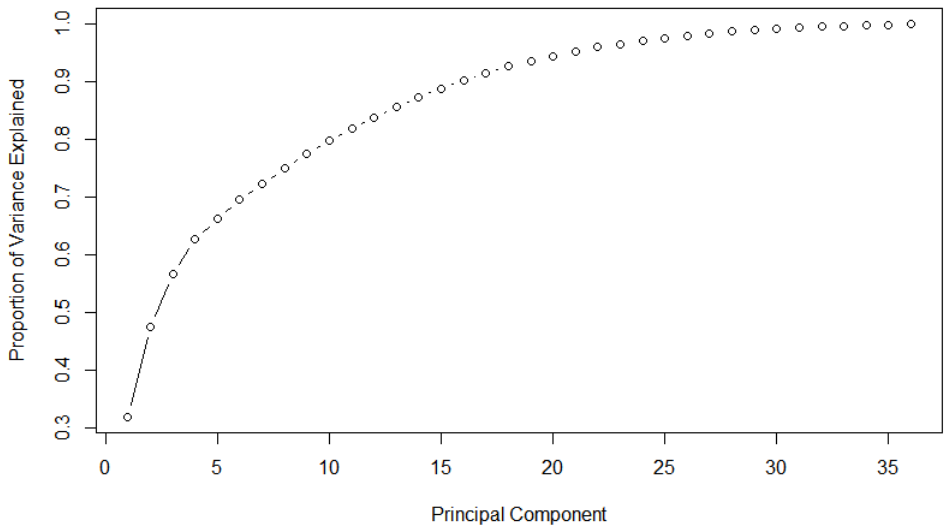

Figure 2.1: Principal Components Analysis, WIMD Features

As shown above, around 30% of the total variance can be explained by the first component and around 70% by the first 5. 95% of the variance, a proportion of the variance that would be considered appropriate for these Principal Components to be used in place of the raw features, is not explained until 21 Principal Components has been reached. At this level, unless a large improvement can be made in terms of model fit or predictive ability, the loss of information and model explainability resulting from the PCA process is unlikely to outweigh the modelling benefits from

this process. As such, if this process does not offer any major benefits on the first model tested, its testing will not be extended to further models.

2.3 Algorithms for Prediction

Now that it has been discussed how the relevant algorithms will be applied in the context of this dataset, each of the algorithms to be used to predict an offender's likelihood of recidivism will be outlined. As the focus of this thesis is to use the most recent machine learning methods to predict the occurrence of crime, it has been chosen to avoid traditional statistical methods such as Binomial Generalised Linear Models (GLMs). Binomial GLMs, in particular, have been avoided for the reason that these models (without the proper training and relevant statistical background) can be mathematically complicated and difficult to understand and maintain - this would therefore defeat the purpose of this research, which is to provide a simple, easy to deploy model to the police that can be maintained and refreshed over a period of time.

The three models that have been chosen for investigation, as previously stated in the introduction, are Random Forests, XGBoost and Neural Networks. Each of these models will now be outlined in turn, along with the advantages and disadvantages of each in a policing context, beginning with Random Forests.

2.3.1 Random Forests

The Random Forests method is a Decision Tree-based ensemble learning method not based solely on parameterised families of probability distributions[15]. It uses the technique of bootstrap aggregation to improve the unstable Decision Tree procedure [14] on many deep decision trees, producing a more stable set of classifications that have a much lower tendency to overfit to the training data than classifications produced by a single decision tree. The approach exhibited by Random Forests is very suited to automatically uncovering complex data structures. In particular, the Random Forests algorithm has the ability to handle a large number of variables of both categorical and numerical types, independent variables that have non-linear effects on the dependent variable, or interactions between independent variables. In the

implementation that has been chosen, R's Ranger [93] package, the Decision Trees will be split according to the rules of the CART algorithm and the "best splits" found through the Gini Impurity.

Here, the method by which Random Forests are used to assess the risk of recidivism for each crime in the dataset will be outlined. This risk assessment will firstly be completed by classifying each crime as one that will lead to a reoffence or not, then producing an estimate of the probability that the crime in question will lead to a reoffence.

Random Forests for Classification

For this dataset, the Random Forests algorithm classifies variables and calculates probabilities of class membership (i.e. will or will not reoffend) in the following way:

1. For each of a series of m crimes c_1,..., $c_m \in \mathbb{R}^n$, take the independent variables attached to this claim to be a set of values $x_1, ..., x_n$ and the corresponding reoffence indicator variable for each of these crimes to be a series $y_1, ..., y_m \in \mathbb{R}$, also of length m.

2. Select a number of Decision Trees, b, to be grown for the forest.

3. For each of the b Decision Trees, sample, with replacement, a subset of crimes $c_1, ..., c_i, i < m$ for each of the m crimes within the dataset. The $m - i$ samples not included in the bootstrapping process are known as the Out Of Bag (OOB) samples in the dataset. These samples, essentially, will be used as a validation set.

4. We then randomly sample, with replacement, a subset $x_1, ..., x_j, j < n$ of the n explanatory variables within the dataset. In this case, j will be initially set to its default value, $\sqrt{n}$ rounded to the next largest integer. This is the value that is most recommended for a classification task and the parameter value will be optimised using the Grid Search method.

5. By recursively splitting the dataset into subsets so that one parent node splits

into two separate child nodes, we consider all splits involving the j available features in the bootstrap sample for that tree. A new sample of j features is chosen at each node and from all splits considered using the j available features, the locally optimal split at each node is chosen based on their Gini Impurity.

6. Grow each of the b Decision Trees to their maximum depth, i.e. keep splitting the data until it is certain that a pathway will lead to a particular outcome, or to a depth determined by a minimum node size parameter o.

7. For each of the m crimes within the dataset, take the majority vote on the classification of each crime c_m as the decision as to whether or not the crime will lead to a reoffence within three years of the crime being reported.

Random Forests for Probability Estimation

If the aim is to output a probability of reoffence rather than simply classify whether or not a reoffence will occur, Step 7 can be replaced with these two steps:

7. As described in Malley et.al [55], in each of the terminal nodes of the tree, the percentage of crimes that lead to a reoffence in each terminal node is determined.

8. To estimate the probability that a given crime within the m crimes in the training set will lead to a reoffence, the crime is dropped down each of the b trees until it reaches the terminal node. The probability of reoffence (percentage of crimes that lead to a reoffence) at the crime's terminal node is then averaged across all of the b trees.

Use and Implications

As Random Forests can be described by taking the concept of decision trees and combining it with a concept that can essentially be described by "taking a majority vote", this algorithm is relatively simple to understand for those with a non-mathematical background. It is also simple to add and remove factors, though these factors will require some processing, particularly if they are categorical in nature.

In terms of parameter tuning and cross-validation, beyond finding the appropriate number of trees (for which there exists a large range of "good enough" numbers), very little tuning is necessary in order for this model to operate, as this tuning will rarely increase the predictive accuracy of the model by more than 1-2%. Nonetheless, in order to ensure good performance, grid search was used to find the optimum value of mtry, or the number of features (j in the above description) competing at each node. While it is also possible to tune other parameters, developing new features will generally be far more effective in increasing the accuracy of the model on unseen data than altering the parameters.

Maintaining and refreshing the model is a relatively simple process, though will require some expertise to refresh it if the factors that affect reoffending change over time - though it is not expected for this to happen in the immediate future, this is always an inherent danger in modelling using historical data.

2.3.2 XGBoost

XGBoost, or extreme gradient boosting, is a scalable machine learning system for tree boosting. Among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 of those solutions incorporated the XGBoost algorithm. In the 2015 KDDCup, an annual data mining competition, it was also used by every team in the top 10 [18]. This method, as such, is certainly worth investigating both as a viable alternative to the Random Forests algorithm and as a further investigation into the use of Gradient Boosting methods for the purposes of predicting recidivism. In general, Gradient Boosting shares some similarities with Random Forests as it can also be thought of as the process of combining multiple weak learners into a single strong learner using Gradient Descent and Boosting techniques. The weak learners in the XGBoost algorithm, like Random Forests, are CART trees. However, unlike Random Forests, the trees are grown to a fixed size and are not allowed to reach their maximum possible depth.

As before, the aim of this algorithm is to teach a model (in this case, a number of CART trees) to predict whether or not an offence will lead to a reoffence within three years, as indicated by reoffence indicator variables $y = y_1, ..., y_m \in \mathbb{R}$, for a

test set of m crimes $c_1, c_2, ..., c_m \in \mathbb{R}^n$ with corresponding independent variables $x = x_1, ..., x_n$. Rather than simply setting a number of trees, the learning in XGBoost is completed over a series of iterations $1 \leq b \leq B$, where B is the maximum number of iterations set by the user. For a number of CART trees B (keeping the notation consistent with Random Forests), the prediction generated by the model for the reoffence outcome of an individual crime $\hat{y}_i$ is generated as follows:

1. An initial model $F_1(x)$ is fitted to the set of reoffence indicators, y. In this case, this will be a Decision Tree that attempts to predict an offender's probability of reoffence.

2. The residuals of this model, $y - F_1(x)$ are then taken and a model, $f_1(x)$, is fit to the residuals. By fitting a further decision tree to just the cases where our model produced incorrect predictions, the aim is to find a pattern within this segment of the data that our initial model may have missed.

3. This model, $f_1(x)$, is then added to the first model to make a new model $F_2(x)$, as follows:

$$F_2(x) = F_1(x) + f_1(x) \tag{2.2}$$

This process of taking a previous model, finding the residuals, fitting a model (Decision Tree) to these residuals and adding this to the previous model then continues until either:

1. A Loss Function is minimised as follows:

$$\mathcal{L} = \sum_i l(\hat{y}_i, y_i) + \sum_b \Omega(F_b) \tag{2.3}$$

The first term, $l(\hat{y}_i, y_i)$, is a differentiable convex loss function measuring the difference between the predicted value $\hat{y}_i$ and the target value y_i. In the XGBoost package for R [19], several different evaluation metrics are available. The second term $\Omega(F)$, where $\Omega(F) = \gamma E + \frac{1}{2}\lambda||w||^2$ is a term that penalises the complexity of each of the CART trees (in this case, E represents the number of leaves in the tree and w the weights applied to those leaves). When this regularisation term is set to

0, traditional gradient tree boosting is used.

2. Reach a point where the the process has been told to stop. The most common way of doing this is by using an early stopping procedure, which will evaluate the progress of the objective function's minimisation and bring learning to a stop when the objective function is no longer significantly improving (the aforementioned "significance" being set by a given level of tolerance).

In either case, a model comprising a number of Decision Trees, B, will eventually be arrived at as follows:

$$\hat{y} = F_B(x) = F_1(x) + \eta(\sum_{b=1}^{B-1} f_b(x)) \tag{2.4}$$

In this equation, η is the learning rate of the function, which controls the weighting of new trees that are added to the model. This is an important parameter that must be tuned in order to avoid overfitting and as before, we made use of grid search to tune this parameter within reasonable bounds. To minimise an objective, traditional optimisation methods in Euclidean space cannot be used, due to the fact that functions are included as parameters. This is the reason why the model must be trained in an additive manner over a number of iterations $1 < b \leq B$ rather than in a single step.

Objectives are optimised at each iteration b, the first and second order gradient statistics on the loss function are introduced so that a second order approximation can be used to quickly optimise this objective. Similarly to the Gini Impurity score that is usually used to evaluate decision trees, XGBoost uses this function as a scoring function to measure the quality of an individual tree structure b.

As it is usually impossible to enumerate all possible tree structures b, however, a greedy algorithm (i.e. one that follows the heuristic of making locally optimal choices in order to find a global optimum) that begins from a single leaf E and iteratively adds branches to each tree b is used instead. To find this best split and to decide which values of which features should be considered for splitting, an approximate version of the exact greedy algorithm, i.e. an algorithm that enumerates

over all possible splits on all features in order to find the best split, is used. This algorithm, instead of looking at all possible values of features, proposes split points according to percentiles of feature distribution. The XGBoost implementation of gradient boosting is also sparsity-aware, making it suitable for use with sparse input data.

Use and Implications

Once again, the XGBoost algorithm is based on decision trees, which makes it relatively simple to explain to those of a non-mathematical background. However, as the process requires the iterative addition of further trees that are fit on to the residuals of the first tree, it can be more difficult to visualise how this algorithm works than it will be for Random Forests. In the same vein as Random Forests, however, it is also simple to add and remove factors, though these factors will require some processing, particularly if they are categorical in nature.

In terms of parameter tuning and cross-validation, due to the nature of the XGBoost algorithm, slightly more care must be taken in order to avoid overfitting the model to the training data, particularly in terms of finding the optimal number of steps to maximise the fit to unseen data. Moreover, the parameters that control the gradient descent (in particular, the learning rate of the algorithm), will require adjustment every time. This will require some understanding of the process of gradient descent on the part of the police, which may or may not already be within their base of specific knowledge.

Maintaining and refreshing the model is, therefore a slightly more complex process compared to the process of maintaining and refreshing Random Forests. Once an appropriate cross-validation process is set up and understood, however, the only real issue appears in (as previously discussed) dealing with the possibility that the factors affecting reoffending will change over time.

2.3.3 Neural Networks

Artificial Neural Networks are computing systems inspired by biological neural networks within animal brains. These networks consist of a set of artificial neurons

(nodes) that produce a series of real-valued activations, joined by a series of directed edges, representing the synapses connecting these neurons. They are designed to interpret sensory data as received by input neurons, which must be numerical and in vector form, in order to recognise and exploit patterns within a given dataset. In these networks, input neurons are activated by sensors that perceive the environment, while other neurons are activated through weighted connections from previously activated neurons. Networks learn or assign credit by iteratively updating the aforementioned weights until the network exhibits the desired behaviour; in our case, this will be producing an accurate prediction as to whether or not a crime will lead to a reoffence within 3 years of its reported date.

In general, Neural Networks can be used to solve many different types of problems. For standard supervised learning classification tasks like this one, which require human knowledge of a dataset in order for a neural network to learn the correlation between a dataset and the labels attached to each observation, many different types of Neural Networks can be used. A series of Feedforward (acyclic) Neural Networks will therefore be built to make use of the independent variables in our dataset to determine whether or not an individual crime will lead to a reoffence within 3 years of the reported date. PCA analysis will not be incorporated into this section.

Neural Network Architectures

The layers in a neural network are made of a finite number H of interconnected nodes, h, which are associated with an activation function $a_h(\cdot)$. Each edge in the finite set of edges D that connects a node h to another node $h^{'}$ is associated with a weight $w_{hh'}$, which assigns a level of importance to the value of the input from the previous node. The value v_h output by each node h is then calculated by applying the activation function a_h to a weighted sum of the values of its input nodes, according to the weights $w_{hh'}$.

$$v_h = a_h\Big(\sum_{h^{'}} w_{hh'} \cdot v_{h'}\Big) \tag{2.5}$$

The Neural Network architecture of a Feedforward Neural Network with L layers is constructed as follows:

1. Beginning with a row of nodes representing an input layer ($\Lambda = 1$, where the nodes are a subset of H) that feeds data into the network.

2. This input layer then passes along an edge to the hidden layers of the neural network, where the input data is processed between nodes via a system of edges.

3. In a feedforward neural network, the Λ^{th} hidden layer of a neural network consists of all nodes $h \in H$ for which an edge path of length $\Lambda - 1$ exists between some input node in layer $\Lambda = 1$ and the node h in the Λ^{th} layer.

4. Once the hidden layers' calculations have been completed, these are fed into an output layer, where the answer that the network has decided upon is output. Once the information has been processed and passed through the neural network once, and the errors have been propagated back in order to adjust the weights, an *epoch* of training is complete.

5. The output of the network is then taken from the output layer and compared to the true outputs of the training dataset. The difference between the predicted and actual outputs is then defined by a loss function $\mathcal{L}(\hat{y}, y)$, which must be minimised over several *epochs* of training in order to produce the most accurate set of weights.

An example diagram of a typical Feedforward Neural Network is given in Figure 2.2 below:

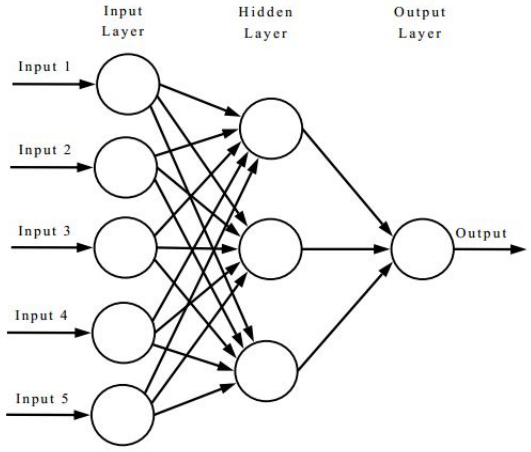

Figure 2.2: Example Neural Network Architecture [1]

To decide upon the values of the weights that produce the most correct output from the input data given to the network, neural networks contain some form of learning rule that modifies the weights along each edge according to the inputs each node is presented with. The degree by which a change in weight causes a change in the network's loss can be measured by the following derivative:

$$\frac{\delta \mathcal{L}(\hat{y}, y)}{\delta w_{hh'}} \tag{2.6}$$

Of course, since each of these weights is just one factor in a network that will pass through multiple activations and sum over several layers, calculating the value of this derivative for a single weight, the chain rule must be used to propagate back through the activations and outputs within the network until the weight in question is arrived at, and its relationship to the overall loss.

This concept is known as backpropagation [73]. It is so named due to it beginning with the final output error $\hat{y^{(h)}} - y^{(h)}$ for a neuron h, then propagating this error back through the network in order to update the weights $w_{hh'}$ so that a loss function $\mathcal{L}(\hat{y}, y)$ is minimised. Due to the non-convex nature of the loss surface, there is no guarantee that the backpropagation algorithm will arrive at a global minimum. In many cases, the network will minimise to a local minimum, which is not the best overall solution.

Cross-Entropy Cost The function $\mathcal{L}(\hat{y}, y)$ to be minimised is the cross-entropy cost. In the context of a neural network, the formula for this loss function for a single neuron h is as follows:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{m} \sum_{c} [y \ln \hat{y} + (1 - y) \ln(1 - \hat{y}) \tag{2.7}$$

where m is the total number of training examples, the sum is over all crimes c within the training dataset and as before, $\hat{y}$ is the model's output and y is the desired output. This function generalises across a network in the following way:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{m} \sum_{c} \sum_{j} [y_h \ln \hat{y}_h + (1 - y_h) \ln(1 - \hat{y}_h)] \tag{2.8}$$

where y_h is the desired output for a single neuron h, $\hat{y}_h$ is the model's output for

that neuron and as such, the function is summed over all output neurons h in the neural network's output layer. Its suitability for use as a cost function stems from two simple properties:

1. The cross-entropy cost is always non-negative.

2. If y gets closer to $\hat{y}$ for all training examples c, then the cross-entropy cost tends towards 0.

Neural Network Optimizers

The derivatives calculated by the backpropagation algorithm will then feed into an optimisation algorithm. Many algorithms are available for use, but here three particular optimisers will be utilised, Adam [48], Adamax and Nadam [26]. These three optimisers have been chosen for three reasons. Firstly, they are computationally efficient and have very low memory requirements, making them well-suited for problems on large datasets with large numbers of parameters. Secondly, the hyper-parameters within these algorithms generally require very little tuning. Thirdly, the algorithms themselves incorporate learning rate decay without the need to manually decrease the learning rate. This means that should the algorithm converge on a poor local minimum due to the learning rate being set too high, the automatic adjustments of the learning rate over time will allow the algorithm to possibly escape this poor local minimum and find a new, preferable, local minimum.

These algorithms will aim to minimise the expected value, $\mathbb{E}[\mathcal{S}(\theta)]$, w.r.t its parameters θ, of a noisy objective function $\mathcal{S}(\theta)$. $\mathcal{S}(\theta)$ is a stochastic scalar function that is differentiable w.r.t parameters θ, with its values at subsequent epochs $1, ..., S$ being denoted by $\mathcal{S}_1(\theta), ..., \mathcal{S}_S(\theta)$. The stochastic nature of the function will come from its evaluation across several random subsamples (minibatches) of training data during each epoch that the Neural Network is trained over.

Adam The Adaptive Moment Estimation (Adam) [48] algorithm is a first-order gradient-based optimisation method for stochastic objective functions. It is based on adaptive estimates of lower-order moments, which are calculated for individual parameters from estimates of first and second moments of their gradients.

This algorithm updates exponential moving averages of the gradient, or estimate of the 1st moment (p_s) and the squared gradient (q_s), or the 2nd raw moment (uncentered variance) of the gradient. Two hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these averages. The step size, or learning rate of the algorithm is denoted by α. All operations on vectors are element-wise. In the Keras version of this algorithm, as well as in the original paper, default settings for the adjustable hyper-parameters are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

Adamax AdaMax is a generalised variant of Adam based on the infinity norm. All operations on vectors are element-wise. In Keras, as well as in the original paper, default settings for the adjustable hyper-parameters are $\alpha = 0.002$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Nadam Nadam [26] is a variant of the Adam optimiser that incorporates Nesterov-accelerated momentum into the algorithm. Where the Adam algorithm combines RMSProp and momentum, Nadam takes it one step further and combines RMSProp with Nesterov-accelerated momentum.

As Nesterov-accelerated momentum often offers superior performance to classical momentum in optimisation in neural networks, there is a clear motivation to modify the Adam algorithm to include this kind of momentum. In the Keras implementation of this algorithm, as well as in the original documentation, the default parameters for this algorithm are $\alpha = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, Decay $= 0.004$. It is not currently recommended to alter these parameters.

Use and Implications

Neural Networks, while being relatively simple in the way they can be explained in basic terms, are quite mathematically complicated once you go beyond the basics. They are also relatively computationally expensive, especially once a large number of inputs or layers are integrated into the model. However, since their effectiveness relative to other Machine Learning methods increases as the number of rows in the input dataset increases, it may be worth the extra computation time if a sufficiently large dataset can be made use of.

In terms of parameter tuning and cross-validation, due to the nature of Neural Networks, slightly more care must be taken in order to avoid overfitting the model to the training data, particularly in terms of finding the optimal number of steps to maximise the fit to unseen data. Similarly to XGBoost, the learning rate and number of epochs (as well as the relationship between them) must be understood in order to appropriately fit the Neural Network to the training data so that it can produce appropriate predictions for unseen datasets.

Maintaining and refreshing the model is, therefore a slightly more complex process compared to the process of maintaining and refreshing Random Forests. Once an appropriate cross-validation process is set up and understood, however, the only real issue is (as previously discussed), the issue by which factors that affect reoffending will change over time.

2.4 Evaluation

The three algorithms detailed in the previous section will be evaluated in two basic ways. Firstly, the relative predictive power of each algorithm will be assessed in turn. Secondly, the importance of each independent variable as determined by each of the three algorithms will be evaluated and compared.

2.4.1 Predictive Power

Predictive Accuracy The two methods used to predict the probability of reoffending will be evaluated in two different ways. Firstly, the accuracy of the model against a holdout test set will be evaluated. Since the aim of this investigation is to select the best predictive model for a crime leading to a reoffence within the Dyfed-Powys police area, it is important to select a method that provides a good level of predictive accuracy. However, due to the possibility that simply choosing a model that maximises the classification accuracy on a test set may lead to fitting a model that is too specific to a particular test set, other metrics must also be considered when evaluating the performance of these models.

ROC Analysis

The secondary objective of this analysis will be to complete an ROC (Receiver Operating Characteristic) analysis on the holdout set predictions for this model. This analysis, which begins with a plot of the ROC curve itself, assesses the performance of a binary classifier (in this case, whether or not an individual will reoffend), as its discrimination threshold (the probability value that designates the boundary between an offence that will or will not lead to a reoffence) is altered. This curve is drawn by plotting the true positive rate, or sensitivity, against the false positive rate, or 1 - specificity, for several different cut-off thresholds. ROC analysis, therefore, allows a model to be selected based on this weighting of true against false positives, which in turn allows us to see the trade-off between maximising the sensitivity of the model to positive responses against the corresponding rate of false positives at a particular threshold. If false positive results are considered to be more damaging than false negatives, then a higher true positive rate can be sacrificed for a lower false positive rate, and vice-versa if false positive rates are considered to be less damaging and a high true positive rate more desirable.

From an ROC plot, several summary statistics can be generated. While generating these summary statistics causes a certain amount of information to be lost in the trade-off between true and false positives, it can be useful to generate these statistics in order to make comparisons between different models.

AUC The statistic that this thesis will be most concerned with is the AUC (Area Under the Curve) statistic. In our case, this is equivalent to the probability that a randomly chosen instance of crime leading to a reoffence will be ranked higher (i.e. the predicted probability will be higher) than a randomly chosen instance of a crime not leading to a reoffence. When TPR is the true positive rate and FPR is the false positive rate, the pointwise estimate of the AUC score is calculated as follows:

$$AUC = \frac{(TPR - FPR + 1)}{2} \tag{2.9}$$

The AUC score is the sum of the area under the ROC curve and is insensitive to imbalanced classes.

Informedness While the AUC statistic has often been used for model comparison across the machine learning community, more recent critical research has suggested that the AUC score may not be as reliable and valid a measure as previously thought [53]. As such, alternative measures must be considered. One such suggested measure is the Informedness (or, in dichotomous cases such as this, Youden's J-Statistic), which specifies the probability that a prediction is informed in relation to the condition (versus chance). In betting terms, this is the probability that an individual is making an informed bet, i.e. the proportion of the time that the individual's information improves the proportion of successes versus pure guesswork. This measure (I) is calculated from and related to the AUC score in the following way:

$$I = TPR - FPR = 2(AUC - 1) \tag{2.10}$$

As cost weighting will not be introduced into either of the models, maximising the AUC score is equivalent to maximising the Informedness. Maximising the Informedness, therefore, can be proxied by maximising the AUC score. This is useful, as many models allow the AUC score to be maximised in order to select the best model, but do not allow similar comparisons between Weighted Relative Accuracy or Informedness. In the case of the XGBoost algorithm, which requires some sort of loss function to be maximised in order to select the best algorithm in the the cross-validation step, the Informedness will, in fact, be maximised by maximising their AUC scores.

2.4.2 Variable Importance

While the test set accuracy and the ROC analysis of each model gives a great deal of information pertaining to the performance of the model, these measures cannot give insight into the effect each of the independent variables have on the predictions produced by the model. This, from a policing point of view, is not ideal; while a "black box" model that simply outputs predictions can be useful, it is also important for the police to know which factors are considered to be the major predictors of whether or not an individual crime will lead to a reoffence. In this way, if a simplification of the model that makes use of fewer features is required, it will be known exactly which features will contribute the most to an offender's risk of recidivism following an offence.

In the Ranger implementation of the Random Forests algorithm, two different options exist for measuring the importance of the variables within the model. While it is possible to use the Gini Importance (mean decrease in impurity) to estimate this for classification and probability trees, as originally suggested by the Random Forests algorithm's author[15], there has been research to suggest that these importance measures may be biased towards variables with a larger number of categories [84]. Since the number of categories in each of the categorical variables can vary quite considerably, this is somewhat of a concern. As such, here, it has been to use the bias-corrected permutation importance [3] as our measure of variable importance. This measure of importance is assessed for each independent variable by randomly permuting the values of the variable over all b trees in the forest and measuring the resulting increase in error. For this measure, the influence of correlated features is removed, which is likely to be extremely useful given that it is likely at least some of our features are correlated.

In R's implementation of XGBoost, these importances are measured in the gain contribution of each feature to the model. This, like the permutation importance, is an indication of the importance of a feature in making the branches of the decision tree more pure. The gain contribution of each feature of each tree is taken into account, then averaged per feature to give a good idea of the importance of each feature in the context of the entire model.

Currently, little conclusive research has been conducted into measuring variable importance for Neural Networks, though sensitivity analysis can be conducted on this topic. As such, in this thesis, the variable importances of each independent variable in the Neural Network that has been constructed here will not be assessed.

2.5 Results

In this section, the results of the three tested algorithms will be presented and their implications will be briefly discussed. The format of these results and their discussion will be slightly different for all algorithms tested, but will be similar enough for direct comparisons to be made between them. In each case, the reported

metrics correspond to the most accurate results from 10 different training runs, in which the training and test data was randomly reshuffled and split each time in a 75:25 training:test ratio. Due to the classes being relatively well-balanced within the dataset, stratification was unnecessary and so a random sample was taken. Validation within each algorithm is conducted on a 10-fold cross-validation basis within the training process and again, the accuracy of a validation set is averaged across the 10 folds. The optimum number of variables at each split and the optimium number of trees in the forest were determined via Grid Search.

2.5.1 Random Forests: Classification

Here, the results will be split into those generated for both classification and probability forests, beginning with the former. Each set of results is generated with the same number of trees and the same optimised number of variables in each bootstrap sample (as determined via Grid Search). The same split criterion (Gini) is used in each case. The number of variables to be included in each bootstrap sample was altered a number of times throughout the analysis, but it was found that the increased predictive accuracy brought by including a larger number of variables in each sample was insufficient to justify either an increase in runtime or an increase in tree complexity. While the Extremely Randomised Trees split criterion, an option within the Ranger package, was briefly considered for its faster runtime, the decrease in runtime was not sufficient to offset the decrease in predictive accuracy.

In this model, the results of model runs incorporating the feature reduction technique, Principal Components Analysis will also be included. This will be to determine whether or not the benefits of employing this feature reduction method on our dataset are likely to outweigh the potential loss in explainability.

Original Dataset

Predictive Accuracy To evaluate the effectiveness of the classification Random Forest, as previously outlined in Section 2.3, the accuracy and AUC scores from the training, test and validation sets must be calculated. These metrics, as calculated for the classification Random Forest are given below in Table 2.10.

Table 2.10: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 8. Splitrule: Gini

Metric	Value	PCA Included
Training Set Accuracy	0.9885	0.9850
Validation Set Accuracy	0.7341	0.7307
Test Set Accuracy	0.7283	0.7248
Training Set AUC Score	0.9882	0.9847
Training Set Informedness	0.9763	0.9694
Test Set AUC Score	0.7281	0.7239
Test Set Informedness	0.4563	0.4478

Firstly, no major differences are observed in any of the metrics are present between the model that uses Raw Features and the model that makes use of PCA. In this case, that the ease of understanding afforded by making use of the raw features is unlikely to be trumped by an increase in predictive accuracy from the use of a technique to reduce the number of features and therefore size of the model.

Secondly, it can clearly be seen that the Random Forests algorithm provides a good fit to the training data, with over 98% accuracy on the training set in each case, a corresponding AUC score of a similar value. The AUC score indicates that a classification forest model provides a good fit to the validation and test data in each case, but not an excellent one. The informedness scores further confirms this, showing that their predictions do provide an improvement above chance, but the models are not as effective at producing accurate predictions as they could be.

Thirdly, the classification accuracy is in line with the out of bag (validation set) error but is somewhat lower than the training set error, indicating that the model overfits to the training data. While the prediction accuracy offers a major improvement above chance (from 52% to 73%), this is no better than the relative accuracy of many comparable models. To have a better idea of where the errors in classification may be coming from, confusion matrices for predictions on both the training and test datasets can be produced, which are shown in Tables 2.11 - 2.14.

Table 2.11: Random Forest Confusion Matrix (Training Data, Without PCA). Parameters: No. Trees = 500, No. of Variables per Split = 8. Splitrule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	21884	447	22331
Predicted Reoffence	44	20168	20212
Total	21928	20615	42543

Table 2.12: Random Forest Confusion Matrix (Training Data, With PCA). Parameters: No. Trees = 500, No. of Variables per Split = 8. Splitrule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	21729	562	22291
Predicted Reoffence	77	20175	20252
Total	21806	20737	42543

In both Tables 2.11 and 2.12, it is more common to encounter a false negative error (a prediction of no reoffence when one actually occurred) than a false positive (a prediction of a reoffence when one did not occur). There appears to be very little difference between the two models in terms of their goodness of fit to the training data, though the model based on raw data does appear to fit it more closely.

Table 2.13: Random Forest Confusion Matrix (Test Data, Without PCA). Parameters: No. Trees = 500, No. of Variables per Split = 8. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5445	2177	7622
Predicted Reoffence	1676	4883	6559
Total	7121	7060	14221

Table 2.14: Random Forest Confusion Matrix (Test Data, With PCA). Parameters: No. Trees = 500, No. of Variables per Split = 8. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5535	2195	7730
Predicted Reoffence	1708	4743	6938
Total	7243	7060	14668

In both Tables 2.13 and 2.14, it is more common to encounter a false negative error (a prediction of no reoffence when one actually occurred) than a false positive (a prediction of a reoffence when one did not occur). Similarly to the training set, the model using raw data appears to fit the data better than the model that makes

use of PCA.

Regarding the test data in the raw data case, of those who were predicted not to reoffend, 72% did not reoffend. Of those who were, however, 74% continued to commit a reoffence within three years. This means that if the model predicts a reoffence from a given crime, it is slightly more certain that this prediction will be correct than if the model predicts that the crime will not lead to a reoffence. In general, this means that the model is slightly optimistic and leans a little too heavily towards predicting that an individual will not reoffend.

While the PCA model performed similarly in terms of those who were predicted not to reoffend, it performed much worse in terms of those who were - of those who were predicted to reoffend, only 68% actually did. This suggests that the PCA model is slightly more "pessimistic" and will more often predict false positives than false negatives.

Variable Importance Following the investigation into the predictive accuracy of Random Forests on our training, validation and test sets, the permutation importance will now be used to investigate which independent variables in the dataset have the greatest effect on the output of the dependent variable.

The 15 most important independent variables, as output by the model on its most predictive iteration, are listed in Table 2.15 in order of relative importance. This table outlines the variable importances for the case in which no dimensionality reduction techniques were used.

Table 2.15: Random Forest Permutation Importances, Raw Data Only. Parameters: No. Trees = 500, No. of Variables per Split = 8.

Variable	Permutation Importance
NoPreviousArrests	**0.0508**
PrevFine	**0.0342**
Outcome	**0.0210**
AgeCommitted	**0.0197**
Off_ASB_per100	**0.0146**
OffenceCat	0.0095
PrevViolence	0.0093
Crime_EmpLTSick_per100000	0.0088
Off_EducAbs_Perc	0.0088
HaversineDist	0.0086
Crime_EducNoQuals_Perc	0.0083
MultipleOffences	0.0082
Off_EmpBenefits_Perc	0.0082
Crime_EmpBenefits_Perc	0.0082
Off_Income_Perc	0.0082

Here, two out of the five most important features here centre around an offender's previous criminal activity - NoPreviousArrests (as previously described in Section 2.2.1) refers to the number of previous arrests committed by the offender, while PrevFine refers to the previous level of harm that they have inflicted on society, as described by the minimum level of fines that would be incurred for such an offence. Interestingly, however, the level of harm of the most recent offence appears to have little impact on whether or not the offence will lead to a reoffence. Without taking an offender's criminal history into account, the most important variables appear to be the police's actions following the offence, the age of the offender at the time the offence was committed and the level of Anti-Social behaviour within the offender's immediate area.

The case in which PCA features replace raw WIMD features will now be observed. In this case, the 35 WIMD features have been reduced to 21 features, which explain 95% of the total variance. A plot of each Principal Component and the % of explained variance was illustrated in Section 2.2. The relative importances of the Top 15 features, including those created by PCA, are outlined in Table 2.16 below.

Table 2.16: Random Forest Permutation Importances, Principal Components Included. Parameters: No. Trees = 500, No. of Variables per Split = 8.

Variable	Permutation Importance
NoPreviousArrests	**0.0537**
PrevFine	**0.0333**
AgeCommitted	**0.0198**
Outcome	**0.0191**
PC1	**0.0156**
HaversineDist	0.0110
OffenceCat	0.0102
PrevViolence	0.0096
MultipleOffences	0.0089
PC3	0.0087
Fine	0.0083
PC7	0.0070
Off_Sex	0.0069
PC4	0.0066
PC8	0.0065

Here, the two features considered to be most predictive of an offender's reoffence remain unchanged and for the most part, the top 5 features are also static - however, one of the WIMD features has been replaced by a Principal Component. In general, features from the Dyfed-Powys dataset are considered to be more important than WIMD features, with the exception of the first principal component (which explains 30% of the variance in the WIMD fetures on its own).

First-Time Offences: Predictive Accuracy As an individual's offence history clearly has a large effect on their risk of recidivism, it may be worthwhile to separate first-time offences from repeat offences to test the model's performance under such restrictions. Firstly, the test set used in Random Forest's best iteration (i.e. the iteration with the highest AUC score on the test set) will be made use of, then restricted so that only predictions made for first-time offences will be included. The accuracy of these predictions is shown in Table 2.17 below.

Please note that "PCA Included" here (and elsewhere) refers to the WIMD variables being replaced by a number of Principal Components and other non-WIMD derived features being left as-is.

Table 2.17: Random Forest Evaluation Metrics, First-Time Offenders. Parameters: No. Trees = 500, No. of Variables per Split = 8.

Metric	Raw Features Only	PCA Included
Test Set Accuracy	0.7111	0.7033
Test Set AUC Score	0.6209	0.6091
Test Set Informedness	0.2418	0.2181

The predictive accuracy on first-time offences is somewhat lower than the that for all offences in the dataset, with 71% of these being classified correctly versus 73%. This drop in the model's predictive power is also seen in the AUC and Importance scores, which have dropped somewhat. As such, the predictions of recidivism risk produced for first-time offences are not quite as accurate as those produced for other offences.

Table 2.18: Random Forest Confusion Matrix (Raw Features Only, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 8. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5246	1861	7107
Predicted Reoffence	693	1040	1733
Total	5939	2901	8840

74% of those who were predicted not to reoffend did not commit a reoffence in reality. When it came to predicting who would reoffend, however, only 60% of those who were predicted to reoffend actually did so.

Table 2.19: Random Forest Confusion Matrix (PCA Included, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 8. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5189	1991	7180
Predicted Reoffence	596	942	1538
Total	5785	2933	8718

Here, similarly to the previous model, 73% of those who were predicted not to reoffend did not commit a reoffence in reality and 61% of those who were predicted to reoffend actually did so. Considering that both models tend heavily towards predicting that an offence will not lead to a reoffence, this suggests that the models are much less efficient at detecting the variables that will make an offence more likely to lead to a reoffence than they are at detecting the variables that will make an offence less likely to lead to a reoffence.

Live Test Dataset

The "Live Test Dataset" is the dataset used by Dyfed-Powys police to independently test the efficacy of the algorithm on unseen data, as well as test its future deployment. The "Live Test Dataset", as defined here, differs from the "Original Dataset" in that the time period considered is different. While the data as given by Dyfed-Powys police encompasses the time period from January 1st 2008 to December 31st 2014, the Live Test data encompasses the time period from January 1st 2015 to December 31st 2017.

Predictive Accuracy To evaluate the effectiveness of the classification forest, as previously outlined in Section 2.4 the accuracy and AUC scores from the training, test and validation sets must be calculated. These metrics, as calculated for the classification Random Forest are given below in Table 2.20.

Table 2.20: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 9. Splitrule: Gini

Metric	Raw Features Only	PCA Included
Training Set Accuracy	0.9962	0.9959
Validation Set Accuracy	0.8026	0.7983
Test Set Accuracy	0.7980	0.8066
Training Set AUC Score	0.9926	0.9921
Training Set Informedness	0.9852	0.9843
Test Set AUC Score	0.6710	0.6749
Test Set Informedness	0.3419	0.3497

Here, it can clearly be seen that the Random Forests algorithm provides a good fit to the training data, with over 98% accuracy on the training set in each case, a corresponding AUC score of a similar value. While the model does appear to have overfit to the training data, given the difference between the training and test set accuracy, the accuracy on both the validation and test datasets has increased by 7% in the case of both models when compared to the original dataset - the AUC score, however, has decreased somewhat.

Again, the AUC score indicates that a classification forest model provides a good fit to the validation and test data in each case, but not an excellent one. The importances scores further confirm this, showing that their predictions do provide an improvement above chance, but the models are not as effective at producing accu-

rate predictions as they could be. In order to determine the likely reason behind this decrease in AUC, a confusion matrix for predictions on both the training and test datasets can be produced, which is shown below in Tables 2.21 - 2.24.

Table 2.21: Random Forest Confusion Matrix (Raw Features Only, Training Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	15433	78	15511
Predicted Reoffence	0	5193	5193
Total	15433	5271	20704

Table 2.22: Random Forest Confusion Matrix (PCA Included, Training Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	15354	84	15438
Predicted Reoffence	0	5266	5266
Total	15354	5350	20704

From these matrices, the model fits perfectly to the training data when the individual reoffends. As such, the model appears to have a tendency to overfit to the training data and inaccuracies on predicting unseen data may be an issue. This tendency is confirmed below.

Table 2.23: Random Forest Confusion Matrix (Raw Features Only, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	6140	1401	7541
Predicted Reoffence	391	941	1332
Total	6531	2341	8873

Table 2.24: Random Forest Confusion Matrix (PCA Included, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	6239	1339	7578
Predicted Reoffence	377	918	1295
Total	6616	2257	8873

Firstly, in both cases, 81% of "did not reoffend" predictions were accurate and 71% of "did reoffend" predictions were accurate. Predictions that the individual will

not reoffend are, therefore, more likely to be reliable than predictions that they will not. An issue, however, arises in the live test data where those that did reoffend are more likely to be predicted as non-reoffenders than reoffenders. This is obviously a worry for the police, as it means that false negatives are more likely to be output by the model than false positives. This could be due to many reasons, but is likely to be due to too small a sample of reoffenders compared to non-reoffenders within the live test data subset. Thankfully, this is an easy enough issue to correct and can be addressed in several ways, including the following:

1. Assuming the factors that affect reoffending have not altered in a major way over time, more historical reoffence data could be included from the Dyfed-Powys base.

2. The current sampling method could be altered so that the proportion of offenders who committed reoffences is adjusted, either by under-sampling from those who did not reoffend or over-sampling from those who did.

3. The "class.weights" parameter in the Ranger forest can be adjusted to take the relative proportions of the did/didn't reoffend classes within the dataset into account. This parameter allows individual weights to be assigned to each class - for example, the following formula might be used to apply weights to j classes in inverse proportion to the frequencies with which they appear in the data:

$$w_j = \frac{m}{km_j} \tag{2.11}$$

where w_j is the weight to be applied to class j, m is the number of observations in the dataset, m_j is the number of observations in class j and k is the number of classes.

The choice of method once these models go live will be up to Dyfed-Powys themselves and different methods may be appropriate for different refreshes, depending on whether the proportion of offences that lead to reoffences increase or decrease as time goes on.

Variable Importance Here, the variable importances generated by the model in order to see whether or not the differences between the live data and the original Dyfed-Powys dataset will be examined to see if these have resulted in any fundamental differences in the factors the model deems to most affect reoffending. These are detailed in Table 2.25 below.

Table 2.25: Random Forest Permutation Importances, Raw Data Only. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Variable	Permutation Importance
HaversineDist	**0.0431**
NoPreviousArrests	**0.0209**
PrevFine	**0.0187**
Outcome	**0.0131**
MultipleOffences	**0.0129**
AgeCommitted	0.0122
OffenceCat	0.0088
Crime_EmpLTSick_per100000	0.0087
Crime_EmpBenefits_Perc	0.0078
Off_ASB_per100	0.0077
Fine	0.0071
Off_EducNoQuals_Perc	0.0070
Off_EmpLTSick_per100000	0.0066
Off_EmpBenefits_Perc	0.0065
Crime_EducNoQuals_Perc	0.0065

Here, the distance between the offender's residence and the location of the crime has become much more important in predicting their probability of reoffence. The age at which the offender committed the crime has now dropped from the top 5 factors, though is still of great importance. The majority of the factors, however, have stayed roughly the same in terms of their importance - interestingly, however, the number of long-term sick individuals and the percentage of those on employment benefits within the area in which the crime was committed appears to be more important than before.

When considering the PCA features created from the WIMD features, once again only the first Principal Component is considered to be especially important. The top 6 features are the same in both models, regardless of whether or not PCA is used. This is shown in Table 2.26.

Table 2.26: Random Forest Permutation Importances, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Variable	Permutation Importance
HaversineDist	**0.0502**
NoPreviousArrests	**0.022**
PrevFine	**0.0186**
MultipleOffences	**0.0159**
Outcome	**0.0158**
AgeCommitted	0.0136
PC1	0.0090
Fine	0.0084
OffenceCat	0.0077
Off_UR01IND	0.0065
PC3	0.0059
PC6	0.0058
PC11	0.0056
PrevViolence	0.0054
PC2	0.0054

First-Time Offences: Predictive Accuracy In order to see the effectiveness of the model on "cold-start" data, the predictive accuracy of the model when it comes to first-time offenders within the live test data must be assessed. The accuracy, AUC and importance metrics for this dataset are included in Table 2.27 below.

Table 2.27: Random Forest Evaluation Metrics, First-Time Offenders. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Metric	Raw Features Only	PCA Included
Test Set Accuracy	0.8454	0.8541
Test Set AUC Score	0.6449	0.6091
Test Set Informedness	0.2899	0.2181

Firstly, the use of PCA features makes the predictive accuracy of the model on unseen data slightly better, but decreases the AUC and importance scores. Secondly,the test set accuracy for first-time offenders has increased compared to the test set accuracy for all offenders - however, the AUC has dropped. In order to see the reasons behind this, the confusion matrices in the model must be examined, shown in Tables 2.28 and 2.29 below.

Table 2.28: Random Forest Confusion Matrix (Raw Features Only, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5166	933	6099
Predicted Reoffence	86	405	941
Total	5252	1338	6590

Table 2.29: Random Forest Confusion Matrix (PCA Included, Test Data). Parameters: No. Trees = 500, No. of Variables per Split = 9. SplitRule: Gini

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	4384	725	5109
Predicted Reoffence	79	322	401
Total	4463	1047	5510

The difference between the accuracy and AUC scores can be easily observed from these matrices - again, it can be seen that imbalanced data has caused an issue in accurately predicting the reoffence outcome of those individuals who did reoffend. The same considerations can, therefore, be applied to producing predictions of reoffence (or otherwise) for first-time offenders.

2.5.2 Random Forests: Probability

While it is doubtless useful to deliver these results to Dyfed-Powys police in the form of a binary classification, it may also be useful for them to consider the raw probabilities of belonging to each class, which in this case is interpreted as the probability that the offence will lead to a reoffence. If, for example, the probability of an reoffence is deemed to be extremely low or extremely high, does this result reflect reality? If so, what sort of decisions are Dyfed-Powys police able to make once armed with this information? In order to properly answer these questions, the list of predicted probabilities and their behaviour at both extremes of the spectrum must be examined.

In order to do this, our classification forest to a probability forest, using the method as first detailed in Malley et al. [55]. These forests, which operate as detailed in Section 2.3, will now be grown in the same way as the classification forests. For consistency, the probability forest will also make use of the same independent vari-

ables for prediction and their predictions will be averaged over the same number of training sets.

Original Data - Predictive Accuracy As before, this analysis will begin with an evaluation of the algorithm's predictive accuracy. A summary table of the predictive accuracy metrics, as evaluated for the best Random Forest out of 10 model runs, is detailed in Table 2.30 below. A threshold of 0.5 is used to convert probabilities into classes.

Table 2.30: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Metric	Raw Features Only	PCA Included
Validation Accuracy	0.8204	0.8197
Training Set AUC Score	0.9924	0.9913
Training Set Informedness	0.9847	0.9826
Test Set AUC Score	0.8084	0.8006
Test Set Informedness	0.6169	0.6011

Here, in both cases, the accuracy of the algorithm on the validation set (OOB error) is much improved and has jumped to around 82%. While the training set AUC score and importance has not altered from the classification forests, the AUC score and importance values calculated from the ROC curve from the test and validation sets have shown improvement. This indicates that a probability forest provides a much better fit on an unseen dataset than a classification forest, and is also somewhat more robust to overfitting on the training set.

To provide a more detailed picture of how well the model performs at all levels of probability prediction, the test set predictions have been split into 10% intervals and compared with the actual probabilities within these intervals.

Table 2.31: Random Forest Predicted Probability Distribution, Test Dataset, Raw Features Only. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{Prob} \leq 0.1$	468	0.05402	0.0534
$0.1 < \text{Prob} \leq 0.2$	1300	0.1548	0.1531
$0.2 < \text{Prob} \leq 0.3$	2071	0.2519	0.2308
$0.3 < \text{Prob} \leq 0.4$	2172	0.3492	0.3066
$0.4 < \text{Prob} \leq 0.5$	1757	0.4491	0.4206
$0.5 < \text{Prob} \leq 0.6$	1535	0.5503	0.5394
$0.6 < \text{Prob} \leq 0.7$	1456	0.6487	0.6731
$0.7 < \text{Prob} \leq 0.8$	1298	0.7493	0.7827
$0.8 < \text{Prob} \leq 0.9$	1160	0.8491	0.8853
$0.9 < \text{Prob} \leq 1$	960	0.9499	0.9635

From this table, in general, the predicted probabilities of reoffence for these 10% intervals tend to be close to the actual probabilities of reoffence. This implies that the probabilities output by the Random Forest algorithm can be assumed to be reasonably reliable, and are therefore suitable for use as read. However, for predicted probabilities close to 1, the model errs on the side of caution and the predicted probability of reoffence is somewhat higher than expected. For predicted probabilities closer to 0, the actual probability of reoffence is somewhat lower than expected. This, from a policing perspective, is actually not an issue; for extreme predictions, the police can actually be more sure of their decision to either put a lesser (in the case of a low probability of reoffence) or greater amount of focus (in the case of a high probability of reoffence) on monitoring an offender following an offence.

Table 2.32: Random Forest Predicted Probability Distribution, Test Dataset, PCA Features Included. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{Prob} \leq 0.1$	487	0.0557	0.0554
$0.1 < \text{Prob} \leq 0.2$	1233	0.1563	0.1655
$0.2 < \text{Prob} \leq 0.3$	2148	0.2523	0.2249
$0.3 < \text{Prob} \leq 0.4$	2198	0.3499	0.3244
$0.4 < \text{Prob} \leq 0.5$	1775	0.4465	0.4282
$0.5 < \text{Prob} \leq 0.6$	1441	0.5481	0.5302
$0.6 < \text{Prob} \leq 0.7$	1474	0.6487	0.6635
$0.7 < \text{Prob} \leq 0.8$	1347	0.7506	0.7825
$0.8 < \text{Prob} \leq 0.9$	1166	0.8504	0.8697
$0.9 < \text{Prob} \leq 1$	895	0.9458	0.9587

In the case of features generated using PCA, the same conclusions can be made; the predicted probability distribution and its relationship to actual probabilities do not appear to differ between the model in which PCA features are included and the model that makes use of raw features.

Original Data - Variable Importances As before, a list of the 15 variables with the highest permutation importance is detailed in Table 2.33 below.

Table 2.33: Random Forest Unscaled Variable Importances, Raw Features Only. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Variables	Permutation Importance
NoPreviousArrests	**0.0450**
PrevFine	**0.0297**
AgeCommitted	**0.0167**
Outcome	**0.0159**
Off_ASB_per100	**0.0124**
HaversineDist	0.0089
Off_EducAbs_Perc	0.0866
PrevViolence	0.0086
OffenceCat	0.0079
Crime_EmpLTSick_per100000	0.0076
Crime_EmpBenefits_Perc	0.0075
Crime_EducAbs_Perc	0.0073
Off_EducNoQuals_Perc	0.0079
Off_Income_Perc	0.0070
MultipleOffences	0.0070

The most important variables here appear to be the total previous harm committed by the offender, their number of previous arrests and the age of the offender at the time of the offence. In the probability forest version of the Random Forest algorithm, the offender's previous fine and the outcome of their offence matters somewhat less than the age at which the offence was committed, though the variables considered to have the greatest impact on whether or not an individual will reoffend have largely been unaltered from the classification model.

Table 2.34: Random Forest Unscaled Variable Importances, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Variables	Permutation Importance
NoPreviousArrests	**0.0466**
PrevFine	**0.0300**
Outcome	**0.0166**
AgeCommitted	**0.0162**
PC1	**0.0155**
HaversineDist	0.0118
OffenceCat	0.0088
PrevViolence	0.0085
PC3	0.0081
MultipleOffences	0.0073
PC8	0.0068
PC6	0.0068
PC9	0.0066
Fine	0.0065
Off_UR01IND	0.0064

No large differences can be observed between the ranking of features generated using PCA and those generated from raw features. PC1, the first feature created from PCA that (as previously stated) explains approx. 30% of the variance in the WIMD features, has replaced the Anti-Social Behaviour rate in the offender's location as the 5^{th} most important feature.

Original Data, First-Time Offences: Predictive Accuracy Returning to the dataset, the predictive power of the Random Forest for cold-start problems, or offences for which the offender is considered to be a first-time offender, will be assessed. The AUC and importance scores for this subset of the test data are given in Table 2.35 below.

Table 2.35: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Metric	Raw Data Only	PCA Included
Test Set AUC Score	0.7062	0.7107
Test Set Informedness	0.4124	0.4215

Here, the AUC and importance scores have decreased in both cases, which is not surprising when compared to the same results on the classification forests. The drop in AUC score was around the same here, with the AUC score dropping from

around 80% in the all offence test set case to around 70% in the first time offences only test set. Once again, the predicted probabilities generated for this restricted dataset at 10% intervals will be shown in Table 2.36 below.

Table 2.36: Random Forest Predicted Probability Distribution, Test Dataset, First-Time Offenders, Raw Dataset Only. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	958	0.0628	0.0125
$0.1 < \text{PProb} \leq 0.2$	2275	0.1512	0.0273
$0.2 < \text{PProb} \leq 0.3$	1715	0.2456	0.0822
$0.3 < \text{PProb} \leq 0.4$	936	0.3470	0.2308
$0.4 < \text{PProb} \leq 0.5$	619	0.4473	0.5267
$0.5 < \text{PProb} \leq 0.6$	710	0.5523	0.8465
$0.6 < \text{PProb} \leq 0.7$	793	0.6515	0.9634
$0.7 < \text{PProb} \leq 0.8$	512	0.7425	0.9805
$0.8 < \text{PProb} \leq 0.9$	158	0.8383	1.0000
$0.9 < \text{PProb} \leq 1$	122	0.9723	0.9918

Here, there is much more of a mismatch between what Random Forest is predicting and the actual probabilities of reoffence that occur within those 10% ranges. In fact, the actual probabilities for offences predicted to be extremely unlikely to lead to a reoffence are somewhat lower than predicted. Similarly, for those offences predicted to have over a 50% chance of leading to a reoffence, our model is too optimistic; in fact, far more of those who are predicted to have over a 50% chance of reoffending actually end up reoffending than the model predicts. The assumption can therefore be made (with a reasonable amount of certainty) that offenders who have been predicted more than a 50% probability of reoffending are extremely likely to reoffend within 3 years of their last offence. While from an empirical point of view, this is not ideal, from a policing point of view such a result can be useful. So long as it is earmarked whether or not the individual is a first-time offender, their predicted probability of reoffence can be assessed accordingly.

Table 2.37: Random Forest Predicted Probability Distribution, Test Dataset, First-Time Offenders, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 9.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	469	0.0569	0.0576
$0.1 < \text{PProb} \leq 0.2$	1226	0.1562	0.1664
$0.2 < \text{PProb} \leq 0.3$	2104	0.2522	0.2277
$0.3 < \text{PProb} \leq 0.4$	2092	0.3494	0.3260
$0.4 < \text{PProb} \leq 0.5$	1484	0.4446	0.4292
$0.5 < \text{PProb} \leq 0.6$	792	0.5423	0.5189
$0.6 < \text{PProb} \leq 0.7$	387	0.6423	0.6641
$0.7 < \text{PProb} \leq 0.8$	151	0.7477	0.7881
$0.8 < \text{PProb} \leq 0.9$	68	0.8430	0.8382
$0.9 < \text{PProb} \leq 1$	72	0.9700	1.0000

The model generated using PCA features is slightly different when it comes to predictions - here, the predicted probabilities are much more in line with the actual probabilities and that the caveats that will need to applied in the previous (raw feature) case will not need to be applied here.

Live Test Data - Predictive Accuracy To evaluate the effectiveness of the probability forest, as previously outlined in Section 2.4, the AUC scores from the training, test and validation sets must be calculated. These metrics, as calculated for the probability forest are given below in Table 2.38.

Table 2.38: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Metric	Raw Data Only	PCA Included
Validation Accuracy	0.8566	0.8560
Training Set AUC Score	0.9981	0.9982
Training Set Informedness	0.9963	0.9963
Test Set AUC Score	0.8095	0.8005
Test Set Informedness	0.6191	0.6011

The AUC on the test dataset for both models has increased by 7% in the case of both models when compared to the original dataset - the informedness score has also increased accordingly.

To provide a more detailed picture of how well the model performs at all levels of probability prediction, the test set predictions have once again been split into 10% intervals and compared with the actual probabilities within these intervals.

Table 2.39: Random Forest Predicted Probability Distribution, First-Time Offenders, Raw Features Only. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	1663	0.0483	0.0421
$0.1 < \text{PProb} \leq 0.2$	1723	0.1486	0.1294
$0.2 < \text{PProb} \leq 0.3$	1370	0.2452	0.2109
$0.3 < \text{PProb} \leq 0.4$	948	0.3461	0.3196
$0.4 < \text{PProb} \leq 0.5$	681	0.4438	0.4376
$0.5 < \text{PProb} \leq 0.6$	411	0.5465	0.5523
$0.6 < \text{PProb} \leq 0.7$	284	0.6446	0.7324
$0.7 < \text{PProb} \leq 0.8$	174	0.7496	0.9138
$0.8 < \text{PProb} \leq 0.9$	76	0.8382	1.0000
$0.9 < \text{PProb} \leq 1$	64	0.9651	0.9531

From this table, in general, the predicted probabilities of reoffence for these 10% intervals tend to be close to the actual probabilities of reoffence at lower probabilities - in fact, until a predicted probability of 60%, the predicted and actual probabilities are very close to one another. This implies that the probabilities output by the Random Forest algorithm can be assumed to be reasonably reliable up until a 60% probability of reoffence, and are therefore suitable for use as read. However, for predicted probabilities close to 1, the model errs on the side of caution and it is actually possible to be very slightly more certain of their reoffence than the model says. This, from a policing perspective, is actually not an issue. For extremely high probabilities of reoffence, the police can actually be more sure of their decision to either put a greater amount of focus (in the case of a high probability of reoffence) on monitoring an offender following an offence.

Table 2.40: Random Forest Predicted Probability Distribution, First-Time Offenders, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	1590	0.0520	0.0434
$0.1 < \text{PProb} \leq 0.2$	1826	0.1495	0.1303
$0.2 < \text{PProb} \leq 0.3$	1341	0.2459	0.2297
$0.3 < \text{PProb} \leq 0.4$	931	0.3470	0.3083
$0.4 < \text{PProb} \leq 0.5$	681	0.4470	0.4332
$0.5 < \text{PProb} \leq 0.6$	472	0.5447	0.5720
$0.6 < \text{PProb} \leq 0.7$	255	0.6484	0.6902
$0.7 < \text{PProb} \leq 0.8$	152	0.7458	0.8618
$0.8 < \text{PProb} \leq 0.9$	78	0.8450	0.9872
$0.9 < \text{PProb} \leq 1$	68	0.9617	1.0000

The same considerations can be applied to the model based on PCA features, but those offenders who receive a probability of reoffence of greater than 70% from this model are those that can be assumed to have a higher actual probability of reoffending.

Live Test Data - Variable Importance As before, a list of the 15 variables with the highest permutation importance is detailed in Table 2.41 below.

Table 2.41: Random Forest Unscaled Variable Importances, Raw Features Only. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Variables	Permutation Importance
HaversineDist	**0.0430**
NoPreviousArrests	**0.0168**
PrevFine	**0.0158**
Outcome	**0.0120**
MultipleOffences	**0.0112**
AgeCommitted	0.0104
OffenceCat	0.0071
Fine	0.0068
Off_EmpBenefits_Perc	0.0067
Off_ASB_per100	0.0067
Off_EducNoQuals_Perc	0.0065
Crime_EmpLTSick_per100000	0.0063
Off_HealthDR_per100000	0.0061
Off_Income_Perc	0.0060
Crime_EmpBenefits_Perc	0.0059

Once again, the distance between the offender's residence and the location of the crime has become much more important in predicting their probability of reoffence. Again, the age at which the offender committed the crime has now dropped from the top 5 factors, though is still of great importance. The majority of the factors, however, have stayed roughly the same in terms of their importance - in this case, the fine incurred by the offender and the Offence Category are considered to be more important relative to any of the WIMD variables.

Table 2.42: Random Forest Unscaled Variable Importances, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Variables	Permutation Importance
HaversineDist	**0.0465**
NoPreviousArrests	**0.01800**
PrevFine	**0.0171**
MultipleOffences	**0.0126**
Outcome	**0.0122**
AgeCommitted	0.0102
PC1	0.0082
Fine	0.0067
OffenceCat	0.0063
PC5	0.0058
Off_UR01IND	0.0056
PC3	0.0056
PC6	0.0049
PrevViolence	0.0046
PC2	0.0045

To emphasise, please note that PCA is used on WIMD variables only, so these features are included in the place of raw WIMD features, they do not generally appear to have a major impact on the model - the only exception to this is the PC1 feature, which is the 7th most important feature in the model.

First-Time Offences: Predictive Accuracy In order to see the effectiveness of the model on "cold-start" data, the predictive accuracy of the model when it comes to first-time offenders within the live test data will now be examined. The AUC and importance metrics for this dataset are included in Table 2.43 below.

Table 2.43: Random Forest Evaluation Metrics. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Metric	Raw Data Only	PCA Included
Test Set AUC Score	0.7927	0.7906
Test Set Informedness	0.5853	0.5813

Firstly, the use of PCA features makes little difference to the AUC or importance scores generated by the model. Secondly, the test set AUC and importance scores have improved greatly compared to both the classification model on the live test data and the probability forest on the original data - the probabilities predicted by the model appear to be a good fit to the first-time offender data provided here.

Table 2.44: Random Forest Predicted Probability Distribution, First-Time Offenders, Raw Features Only. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	1648	0.0481	0.0425
$0.1 < \text{PProb} \leq 0.2$	1645	0.1480	0.1295
$0.2 < \text{PProb} \leq 0.3$	1116	0.2427	0.1927
$0.3 < \text{PProb} \leq 0.4$	526	0.3410	0.2928
$0.4 < \text{PProb} \leq 0.5$	224	0.4397	0.4241
$0.5 < \text{PProb} \leq 0.6$	85	0.5472	0.6588
$0.6 < \text{PProb} \leq 0.7$	89	0.6517	0.9213
$0.7 < \text{PProb} \leq 0.8$	80	0.7524	1.0000
$0.8 < \text{PProb} \leq 0.9$	42	0.8349	1.0000
$0.9 < \text{PProb} \leq 1$	30	0.9743	1.0000

Once again, there is much more of a mismatch between what Random Forest is predicting and the actual probabilities of reoffence that occur within those 10% ranges. For those offences predicted to have over a 50% chance of leading to a reoffence, our model is too optimistic; in fact, far more of those who are predicted to have over a 50% chance of reoffending actually end up reoffending than the model predicts. From this distribution, it can be assumed (with a reasonable amount of certainty) that offenders who have been predicted more than a 60% probability of reoffending are extremely likely to reoffend within 3 years of their last offence. As long as it is earmarked whether or not the individual is a first-time offender, their predicted probability of reoffence can be assessed accordingly.

Table 2.45: Random Forest Predicted Probability Distribution, First-Time Offenders, PCA Included. Parameters: No. Trees = 500, No. of Variables per Split = 7.

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	1582	0.0520	0.04362
$0.1 < \text{PProb} \leq 0.2$	1746	0.1488	0.1289
$0.2 < \text{PProb} \leq 0.3$	1092	0.2435	0.2289
$0.3 < \text{PProb} \leq 0.4$	521	0.3429	0.2956
$0.4 < \text{PProb} \leq 0.5$	211	0.4401	0.4645
$0.5 < \text{PProb} \leq 0.6$	106	0.5427	0.6509
$0.6 < \text{PProb} \leq 0.7$	69	0.6569	0.9130
$0.7 < \text{PProb} \leq 0.8$	68	0.7507	1.0000
$0.8 < \text{PProb} \leq 0.9$	48	0.8475	1.0000
$0.9 < \text{PProb} \leq 1$	33	0.9634	1.0000

Similar results can be observed for those probabilities predicted using the model that includes PCA features. Once again, any first-time offences that are predicted to have a greater than 50% chance of reoffending will need some care in their interpretation.

Now that the results from the Random Forest algorithm and the implications thereof have been discussed, our attention will be turned to the second of the three algorithms to be discussed in this chapter, XGBoost.

2.5.3 XGBoost

The XGBoost algorithm, while still relatively simple to execute in R, requires a little more care in its execution than Random Forests. In fact, it is quite easy to run XGBoost over too many iterations, which will cause it to overfit to the training set, resulting in a lower than expected test set accuracy. To prevent overfitting, two actions can be taken. The first and the simplest of these actions is to alter the value of the learning rate, or the step parameter, so that gradient descent is performed in slightly smaller steps. After testing, it has been chosen to set this parameter to a slightly lower than default value of 0.25 and increase the maximum number of iterations accordingly. The second action, which is to add a cross-validation step to the algorithm before running it for prediction on a holdout test set, is an effective way of reducing overfitting caused by too many iterations of the algorithm on the training set. This cross-validation step, which is used to find a reasonable stopping point

for the algorithm, is implemented in XGBoost via the *xgb.cv* module. The default number of folds for cross validation, 10, remained for convenience. For this cross-validation step, a reasonable maximum of 500 iterations was decided upon, with the algorithm being set to stop after 25 consecutive iterations of reduced model performance. Once the optimum number of iterations was found, XGBoost was trained on the full training set and a prediction was produced using a holdout test set.

Predictive Accuracy The evaluation metrics calculated for this prediction are detailed below. The validation set metrics are omitted here, as these were used to evaluate the quality of the predictions at each iteration.

Table 2.46: XGBoost Evaluation Metrics. Parameters: Max. Iterations = 500, No. Cross-Validation Folds = 10

Metric	Value
Test Set Accuracy	0.7244
Training Set AUC Score	0.9670
Training Set Informedness	0.9341
Test Set AUC Score	0.7973
Test Set Informedness	0.5946

The test set classification accuracy for this model (based on a default cut-off probability of 0.5) is slightly lower than for either of the Random Forest models. Moreover, the AUC and Importance scores of both the training and test sets are also slightly lower than those of the corresponding probability forest. In comparison to the Random Forests algorithm, XGBoost performs comparatively poorly, both in terms of predictive power and goodness of fit to the dataset.

Table 2.47: XGBoost Confusion Matrix. Parameters: Max. Iterations = 500, No. Cross-Validation Folds = 10

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5534	2144	7678
Predicted Reoffence	1764	4739	6503
Total	7298	6883	14181

Once again, there are slightly more false negative predictions than false positives as a percentage of the total positive and negative predictions, meaning that predictions indicating that the offender will reoffend are likely to be slightly more

accurate than predictions that they will not. In accordance with the proportions of the test set, slightly more crimes were predicted not to lead to a reoffence. To see exactly how accurate the model's predictions are at each of the predicted probability values, this can be broken down in the same way as was done for the Random Forest algorithm in Section 2.3, looking at the mean of the actual results versus the mean of the results predicted by the model for each 10% increment. These results are shown below in Table 2.48.

Table 2.48: XGBoost Predicted Probability Distribution. Parameters: Max. Iterations = 500, No. Cross-Validation Folds = 10

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{Prob} \leq 0.1$	946	0.0704	0.0951
$0.1 < \text{Prob} \leq 0.2$	1950	0.1503	0.1974
$0.2 < \text{Prob} \leq 0.3$	1934	0.2485	0.2844
$0.3 < \text{Prob} \leq 0.4$	1589	0.3487	0.3682
$0.4 < \text{Prob} \leq 0.5$	1259	0.4486	0.4241
$0.5 < \text{Prob} \leq 0.6$	1163	0.5488	0.5056
$0.6 < \text{Prob} \leq 0.7$	1174	0.6193	0.6503
$0.7 < \text{Prob} \leq 0.8$	1145	0.7500	0.6760
$0.8 < \text{Prob} \leq 0.9$	1307	0.8511	0.8095
$0.9 < \text{Prob} \leq 1$	1714	0.9499	0.9288

While the predicted probabilities are somewhat close to the actual probabilities of reoffence for the most part, there are a few discrepancies. The crimes for which the probability of reoffending is predicted to be between 0.7 and 0.8, as well as those for which the estimate is predicted to be between 0.8 and 0.9, appear to be somewhat less likely to lead to a reoffence than they are predicted to be. In addition, the crimes for which the probability of reoffending is predicted to be low (less than 0.2) appear to be somewhat more likely to lead to a reoffence than they are predicted to be. With decreased certainty that these extreme predictions can be relied upon, it is more difficult for the user to be able to have confidence in supposedly "clear-cut" cases. As such, from the perspective of the end user, the performance of the XGBoost model does not compare favourably to the performance of the Random Forests model.

Variable Importances As before, an equally important part of the predictions produced by this model is an estimate of how much each individual variable affects the output of the model. In the XGBoost algorithm, this variable importance is

measured in the gain contribution of each feature to the model. In a boosted tree model like this one, the gain contribution of each feature in each tree is taken into account, then averaged per feature to get the importances for the model as a whole. The variable importances, as generated by the best run of the XGBoost algorithm, are listed in Table 2.49 below.

Table 2.49: XGBoost Variable Importances. Parameters: No. Trees = 500, No. of Variables per Split = 10.

Variable	Permutation Importance
NoPreviousArrests	**0.0890**
TimeSincePrevious	**0.0766**
HaversineDist	**0.0532**
AgeCommitted	**0.0439**
Off_HealthCancer_per100000	**0.02843**
Crime_HealthCancer_per100000	0.0254
Off_HealthDR_per100000	0.0247
Off_EducAbs_Perc	0.0241
PrevFine	0.0240
Off_ASB_per100	0.0212
BurgValue	0.0202
Crime_HealthDR_per100000	0.0194
Crime_HealthLBW_Perc	0.0192
Off_EducKS4L2_Perc	0.0189
Off_Fire_per100	0.0164

Once again, the two most important variables here relate to the offender's previous activity. Interestingly, the model rates continuous variables more highly than categorical variables, with many demographic variables, as well as the distance between the crime's location and the offender's home being rated more highly than in the RandomForest model. As yet, the reason for this is unclear, but it is possible that altering the continuous variables so that they behave as categories may change the model for the better and stop this bias in variable selection.

First-Time Offences: Predictive Accuracy As before, it was decided that since the most important variables here relate to the offender's previous activity, it is well worth examining the predictive accuracy on first-time offences, or the cold-start component of this problem. The results of the best run of the model, as evaluated only on first-time offences are detailed in Table 2.50 below.

Table 2.50: XGBoost Evaluation Metrics, First-Time Offenders. Parameters: Max. Iterations = 500, No. Cross-Validation Folds = 10

Metric	Value
Test Set Accuracy	0.7010
Test Set AUC Score	0.7015
Test Set Informedness	0.4030

Here, the accuracy of the test set has decreased compared to the accuracy across the whole dataset. The AUC and importance scores have also shown a decrease on this sample compared to that of the whole dataset, which implies that in general, the model classifies offences that are not committed by first-time offenders better than those committed by first-time offenders. To see where exactly these errors lie (i.e. whether they are mostly comprised of false positives or false negatives), the confusion matrix generated as in the full dataset is given in Table 2.51 below.

Table 2.51: XGBoost Confusion Matrix. Parameters: Max. Iterations = 500, No. Cross-Validation Folds = 10

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	5116	1820	6936
Predicted Reoffence	823	1081	1904
Total	5939	2901	8840

As a proportion of the predictions made in each case, there are far more false positives than false negatives. While 73.76% of the predictions of no reoffence were correct, only 56.78% of the "positive predictions" were correctly made, assuming that a probability of reoffence of 0.5 or more can be taken to mean that the individual will reoffend. While from this information, an assumption can be made that a prediction that an offender will not reoffend will be reasonably accurate following a first time offence, the same assumption cannot be made about a prediction that an offender will reoffend.

To see exactly how accurate the model's predictions are at each of the predicted probability values, this can be broken down in the same way as was done for the Random Forest algorithm in Section 2.3, looking at the mean of the actual results versus the mean of the results predicted by the model for each 10% increment. These results are shown below in Table 2.52.

Table 2.52: XGBoost Predicted Probability Distribution, First-Time Offenders. Parameters: No. Trees = 500

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{Prob} \leq 0.1$	916	0.0710	0.0961
$0.1 < \text{Prob} \leq 0.2$	1893	0.1500	0.1954
$0.2 < \text{Prob} \leq 0.3$	1813	0.2480	0.2785
$0.3 < \text{Prob} \leq 0.4$	1364	0.3479	0.3482
$0.4 < \text{Prob} \leq 0.5$	950	0.4477	0.4021
$0.5 < \text{Prob} \leq 0.6$	740	0.5470	0.4743
$0.6 < \text{Prob} \leq 0.7$	578	0.6488	0.5623
$0.7 < \text{Prob} \leq 0.8$	354	0.7439	0.5989
$0.8 < \text{Prob} \leq 0.9$	148	0.8426	0.7500
$0.9 < \text{Prob} \leq 1$	84	0.9553	0.9762

Here, there is a clear issue with low predicted probabilities of reoffence; in fact, when an offender is given a low probability of reoffence, their chance of reoffending is actually slightly higher than predicted. This issue, with the exception of reoffence probabilities predicted to be between 0.9 and 1, is actually much worse for high predicted probabilities of reoffence. It seems that once the model starts to predict a probability of reoffence of above 40%, these predicted probabilities of reoffence are actually higher than they should be - i.e. at these levels, the model is actually too quick to condemn an offender and thinks that their first offence is more likely to lead to a reoffence than it actually is.

Now that the results of the XGBoost algorithm and the implications thereof have been discussed, the results of the third and final classification algorithm, a feedforward Neural Network, will now be discussed.

2.5.4 Neural Networks

As previously stated in Section 2.3, the type of Neural Network that has been built here is a fully connected Feedforward Neural Network. The maximum number of epochs that the model was trained over was 250 and the batch size was set to 25. This number of epochs is set as a maximum rather than an absolute number due to the use of an early stopping condition, which stops the neural network from learning after 20 runs if there is not at least an improvement of 0.001 in the loss on the validation set. For the purposes of this model, it has been chosen to use 10% of the data for validation so that the size of the validation set is in keeping with the those

used in the Random Forests and XGBoost models.

After running a number of tests, it has been decided that it is unnecessary to include more than 1 hidden layer in the neural network. In fact, multiple hidden layers were actually found to either decrease model performance, or keep performance stable but drastically increase computing time. Since the focus of this analysis is to produce a model that performs as closely as possible on unseen data as it does on the training dataset, the model to be chosen here must *generalise* well and not overfit to the training set. As stated in Neural Network Design (2nd Edition) [37], the general principle of finding a model that generalises well is to find the simplest model that explains the data, which in the case of Neural Networks is the one that contains the fewest free parameters. Since the free parameters in a Neural Network are the weights and biases of the network, this also translates to finding the network with the minimum number of neurons. Following the general rules outlined in this book, a possible general rule for finding suitable numbers of neurons for use in supervised learning problems can be constructed. In this thesis, the upper and lower bounds on the number of hidden neurons that is unlikely to result in overfitting can be found by:

$$\frac{M}{\nu(I+O)} \tag{2.12}$$

where M is the number of samples in the training set, I is the number of input neurons, O is the number of output neurons and $2 \leq \nu \leq 10$. In our case, with 116 input neurons, 1 output neuron and a training set of size 30122 (minus the 10% of data that is removed for validation), the minimum number of neurons that should be considered is 26, and the maximum is 128. Following this rule, a number of different layer sizes within this range will be tested.

To further avoid overfitting to the training set, a number of dropout layers were also tested on this network. These layers function by randomly setting a fraction of units input into the layer to 0 during training. In the tested networks, 1 dropout layer was included in the architecture between the input and hidden layers. Including a greater number of dropout layers was found to cause the model to underfit. This dropout layer was included after the input layer and set to a number of different

values, beginning with 0.25.

Predictive Accuracy For consistency, all networks were tested on and the outputs averaged over the same training and test sets for each set of epochs. The results are detailed in Tables 2.53 and 2.54 below.

Table 2.53: Neural Network Predictive Accuracy (Training Set), Dropout = 0.25

Neurons	30	45	60	75	90	105	120
Optimiser							
Adam	0.7278	0.7398	0.7393	0.7602	0.7514	**0.7705**	0.7579
Adamax	0.7263	0.7280	0.7507	0.7461	0.7620	0.7585	0.7574
Nadam	0.7204	0.7322	0.7347	0.7331	0.7433	0.7502	0.7396

Table 2.54: Neural Network Predictive Accuracy (Test Set), Dropout = 0.25

Neurons	30	45	60	75	90	105	120
Optimiser							
Adam	0.7039	0.7052	0.7033	0.6981	0.7069	0.7022	0.7055
Adamax	0.7098	0.7077	0.7045	0.7087	0.7053	0.7009	0.7018
Nadam	**0.7142**	0.7091	0.7114	0.7080	0.7057	0.7030	0.7077

The model providing the highest accuracy on the training set was trained with the Adam optimiser, with 105 neurons in the hidden layer. The model that produced the highest accuracy on the test set, however, was trained with the Nadam optimiser, with 30 neurons present in the hidden layer. While the test set accuracy was relatively stable across all models tested, with the lowest being 0.6981 and the highest being 0.7142, the best model in terms of test set accuracy actually outperformed the second-best performing model by 0.3%. Here, increasing the number of neurons in the hidden layer increases the predictive accuracy on the training data, but this often comes at the expense of the predictive accuracy on the test set, meaning that a greater number of neurons in each layer is likely to lead to a certain amount overfitting. While this overfitting is less evident for networks trained using the Nadam optimiser than those trained using the Adam or Adamax optimisers, in all cases, the level of overfitting does increase as the number of neurons in the network increases.

From these results, it can be concluded that it is likely that the optimiser that

gives the best predictive accuracy is the Nadam optimiser and that there is little point in including more than 30 neurons in the network, as increasing the number of neurons only increases the predictive accuracy on the training set. To fully understand how well these networks fit the training and test datasets, however, the relative AUC scores on each set must be observed. These scores, as produced for the same set of networks as above, are given in Tables 2.55 and 2.56 below.

Table 2.55: Neural Network AUC Scores (Training Set), Dropout = 0.25

Neurons	30	45	60	75	90	105	120
Optimiser							
Adam	0.7947	0.8138	0.8154	0.8412	0.8318	**0.8548**	0.8379
Adamax	0.7948	0.7995	0.8118	0.8243	0.8401	0.8365	0.8381
Nadam	0.7880	0.8054	0.8098	0.8070	0.8214	0.8293	0.8224

Table 2.56: Neural Network AUC Scores (Test Set), Dropout = 0.25

Neurons	30	45	60	75	90	105	120
Optimiser							
Adam	0.7595	0.7647	0.7634	0.7562	0.7645	0.7587	0.7664
Adamax	**0.7710**	0.7667	**0.7700**	0.7656	0.7641	0.7616	0.7590
Nadam	**0.7670**	0.7684	0.7668	0.7689	0.7633	0.7616	0.7682

Once again, the model providing the highest AUC score on the training set was trained with the Adam optimiser, with 105 neurons in the hidden layer. The model with the highest AUC score on the test set, however, was actually not the 30 neuron version trained with the Nadam optimiser, but the 30 neuron Adamax version. While the test set AUC score was relatively stable across all models tested, with the lowest being 0.7587185 and the highest being 0.7709553, there is very little difference (less than 0.1%) between the best model and the second and third-best models in terms of test set AUC. Once again, it is quite obvious that increasing the number of neurons in the hidden layer increases the training AUC score (i.e. the goodness of fit to the training set), but this often comes at the expense of the goodness of fit to the test set, meaning that a greater number of neurons in each layer is likely to lead to overfitting. While the improvement in test set AUC scores between the Adam and Adamax/Nadam optimisers can be seen, it appears to be somewhat difficult to differentiate between the efficacy of the Adamax and Nadam optimisers in terms of AUC.

From these two tables, it can be said that the two best networks in terms of generalisation are the Adamax and Nadam-optimised network with 30 neurons in the hidden layer. In order to get the best performance from these networks, as well as to get an idea of how they perform under increased regularisation, the level of dropout between the layers of the network must be optimised. The predictive accuracy scores at different levels of dropout are included in Tables 2.57 and 2.58 below.

Table 2.57: Neural Network Predictive Accuracy (Training Set). Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Dropout	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Optimiser							
Adamax	0.7297	0.7278	**0.7347**	0.7216	0.7127	0.7041	0.6689
Nadam	**0.7300**	0.7262	0.7238	0.7161	0.7041	0.6848	0.6901

Table 2.58: Neural Network Predictive Accuracy (Test Set). Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Dropout	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Optimiser							
Adamax	0.7082	0.7053	0.7045	**0.7091**	**0.7093**	**0.7092**	0.6687
Nadam	0.7039	**0.7105**	0.7023	0.7080	0.7067	0.6884	0.6958

Here, the networks as fitted by the Adamax and Nadam optimisers take the addition of dropout layers quite differently. When considering the predictive accuracy on both the training and test sets, networks optimised by Adamax are less affected by higher rates of dropout than those built by Nadam and in fact, better test set accuracy can be found at higher dropout rates (0.5-0.6) for networks optimised by Adamax. The predictive accuracy for the training set remains stable even with a dropout rate of 0.5, which is not the case for networks built under Nadam.

From these accuracy scores, the best generalisation is found either in an Nadam-optimised network with a dropout rate of approximately 0.2 or an Adamax-optimised network with a dropout rate of approximately 0.4-0.6. Once again, however, in order to see how well each of these networks fits the training and test datasets, the AUC scores produced for each level of dropout must be calculated. The AUC scores at different levels of dropout are included in Tables 2.59 and 2.60 below.

Table 2.59: Neural Network AUC Scores (Training Set). Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Dropout	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Optimiser							
Adamax	0.7990	0.7984	**0.8032**	0.7861	0.7765	0.7629	0.7592
Nadam	**0.8040**	0.7977	0.7904	0.7859	0.7706	0.7585	0.7501

Table 2.60: Neural Network AUC Scores (Test Set). Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Dropout	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Optimiser							
Adamax	0.7626	0.7637	0.7656	**0.7684**	0.7670	0.7592	0.7629
Nadam	0.7601	0.7665	0.7637	**0.7684**	0.7644	0.7632	0.7569

The AUC scores tell a slightly different story; in fact, the best AUC score on the test set is found at a dropout rate of 0.4 in both cases. Training set-wise, unsurprisingly, the highest AUC scores are found for networks with a low level of dropout (0.3 for Adamax and 0.1 for Nadam). Overall, better AUC scores on the test set are found at medium levels of dropout, and generally the Adamax optimiser is more reliable in its performance. As such, the recommendation for this dataset is to produce a neural network with 30 neurons in the hidden layer, to use a dropout rate of 0.4 between the input and hidden layers and to make use of the Adamax optimiser.

2.5.5 Comparison: All Algorithms

Now that our final network has been selected, the predictive accuracy of this Feed-forward Neural Network can be compared with the results of the Random Forests and XGBoost algorithms. In general, while it appears to be much easier to control overfitting with this method without reducing the performance on the test set accordingly, the predictive ability of these networks is somewhat reduced in comparison to the Random Forests and XGBoost algorithms. To see exactly how the actual and predicted probabilities of reoffence compare across the scale, as before, the probabilities output by this network will once again be separated into 10% intervals. These results are displayed in Table 2.61 below.

Table 2.61: Neural Network Predicted Probability Distribution. Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	103	0.0748	0.0583
$0.1 < \text{PProb} \leq 0.2$	1094	0.1637	0.1088
$0.2 < \text{PProb} \leq 0.3$	2866	0.2550	0.2142
$0.3 < \text{PProb} \leq 0.4$	3016	0.3449	0.3219
$0.4 < \text{PProb} \leq 0.5$	2200	0.4461	0.4345
$0.5 < \text{PProb} \leq 0.6$	1706	0.5498	0.5639
$0.6 < \text{PProb} \leq 0.7$	1548	0.6472	0.6699
$0.7 < \text{PProb} \leq 0.8$	980	0.7443	0.7908
$0.8 < \text{PProb} \leq 0.9$	622	0.8437	0.9148
$0.9 < \text{PProb} \leq 1$	209	0.9444	0.9713

Here, the performance of a Neural Network does not compare favourably to the current best-performing model, Random Forest. Like XGBoost, the model is reluctant to give any extreme probability predictions and as such, tends to underestimate the probability of reoffence at high probability predictions and overestimate it at low probability predictions. From a policing perspective, this is not ideal, as it introduces a greater deal of uncertainty into the treatment of offenders deemed very likely or unlikely to reoffend. It is, however, entirely possible that its performance on cold-start data may be preferable to either of those algorithms. As such, in order to get a full picture of the model's performance, this performance must also be tested.

First-Time Offences: Predictive Accuracy The accuracy, AUC and importance metrics for the Neural Network are given in Table 2.62 below.

Table 2.62: Neural Network Evaluation Metrics, First-Time Offenders. Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2, Dropout Rate = 0.4, Optimiser = Adamax

Metric	Value
Test Set Accuracy	0.7092
Test Set AUC Score	0.6897
Importance	0.3794

Once again, the predictive accuracy on the cold-start dataset as assessed by both the test set accuracy and AUC scores is somewhat lower than the accuracy as assessed for the whole dataset. This implies that in general, Neural Networks are better at predicting the offence outcome of those who have committed offences in the past than those who are considered to be new offenders. To see exactly where these classification issues lie, a confusion matrix will be produced, the results of which are detailed in Table 2.63 below.

Table 2.63: Neural Network Confusion Matrix, First Time Offenders. Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2, Dropout Rate = 0.4, Optimiser = Adamax

	Didn't Reoffend	Did Reoffend	Total
Predicted No Reoffence	6102	527	6629
Predicted Reoffence	2275	731	3006
Total	8377	1258	9635

Here, the issues inherent in the predictions from the neural network are far more present than previously thought. While the network's predictions of no reoffence are reliable, individuals who are predicted to reoffend do not do so in far too many cases. While this is preferable to the alternative, in which individuals who do go on to reoffend are rarely picked up on, monitoring offenders in the method as indicated by this neural network could be somewhat cost-intensive for Dyfed-Powys police. On the other hand, if the model does predict that an individual will not reoffend, this prediction can be relied upon, so it will be quite simple for Dyfed-Powys police to exclude certain first-time offenders from the dataset on the basis that their offence is unlikely to lead to a reoffence.

To further illustrate where the issues with these predictions lie, the probabilities can also be separated into 10% intervals, as previously tabulated in Section 2.3. These results are shown in Table 2.64 below.

Table 2.64: Neural Network Predicted Probability Distribution, First-Time Offenders. Parameters: Max. Epochs = 250, No. Neurons per Hidden Layer = 30, No. Hidden Layers = 2

Predicted Prob.	Count	Predicted Mean Prob.	Actual Mean Prob.
$0 \leq \text{PProb} \leq 0.1$	101	0.0747	0.0495
$0.1 < \text{PProb} \leq 0.2$	1064	0.1640	0.1090
$0.2 < \text{PProb} \leq 0.3$	2761	0.2549	0.2097
$0.3 < \text{PProb} \leq 0.4$	2772	0.3439	0.3135
$0.4 < \text{PProb} \leq 0.5$	1679	0.4436	0.4205
$0.5 < \text{PProb} \leq 0.6$	886	0.5440	0.5237
$0.6 < \text{PProb} \leq 0.7$	259	0.6346	0.6371
$0.7 < \text{PProb} \leq 0.8$	37	0.7349	0.7568
$0.8 < \text{PProb} \leq 0.9$	59	0.8566	0.9661
$0.9 < \text{PProb} \leq 1$	17	0.9271	1.0000

The probabilities of reoffence as output by the Neural Network do not correspond very well with the actual probabilities of reoffence as exhibited within these 10% prediction ranges. At low predicted probabilities, it seems that the probabilities of reoffence are predicted to be much higher than they actually are. At high predicted probabilities, the opposite situation occurs; the probabilities of reoffence are predicted to be much lower than they actually are. Moreover, it is comparatively very rare for a high probability of reoffence to be predicted, meaning that the model is rarely near-certain that an individual will reoffend, but it often outputs low probabilities of reoffence. While this pattern is evident from the confusion matrix, it is also evident here.

2.6 Conclusions, Limitations and Further Research

While none of the models offer a perfect solution to the reoffending problem by any means, they do offer an interesting perspective on how existing police data can be used to predict whether or not an individual crime will lead to a reoffence. To decide which model should be recommended most highly for use to Dyfed-Powys police, however, the relative merits of each method must be discussed, paying particular attention to both the quantitative and non-quantitative concerns that Dyfed-Powys may have, as well as the limitations of the dataset as they relate to these predictions.

2.6.1 Model Performance

In terms of both raw classification accuracy and the relative usability of the probabilities produced by either model, Random Forests outperforms both XGBoost and a feedforward Neural Network by a reasonable margin. Although XGBoost's performance may well be improved by reformatting some of the continuous variables within the dataset as categorical, and the Neural Network's by the inclusion of more crime data or an alternative architecture, this analysis still sways in favour of Random Forests as the preferred model at this current time. When regarding the confusion matrix in the classification case, each model has similar issues with prediction; when considering the percentage of false negatives as a percentage of the total negative predictions versus the percentage of false positives as a percentage of the total positive predictions, it is clear that a greater percentage of false positives are produced than false negatives. This, for our purposes, is preferable to the other way around; from a policing perspective, the social cost of a false negative is likely to be far higher than the social cost of a false positive, as it is generally better to be on the safe side and monitor more people than have a potentially dangerous offender roaming the streets unchecked. With all things being relatively equal in terms of the balance between false negatives and false positives, it is possible to say that for this purpose, it appears that Random Forests is currently the best model for predicting whether or not an individual will reoffend within 3 years of their last offence.

When it comes to "cold start" models (i.e. models trained or tested on first-time offenders only), more testing and possibly more features are required in order to separate those more likely to reoffend from those less likely to do so. Such investigations, however, would require the collection of alternative datasets that may or may not rest in the public domain, and are therefore beyond the scope of this particular analysis. Once again, it was found that the Random Forests algorithm offered the best performance on cold-start data.

With the exception of the feedforward Neural Network, for which research is currently somewhat inconsistent, these models also offer some indications as to which currently recorded variables are most likely to affect the probability of reoffence. As such, from this perspective, Random Forests and XGBoost offer an advantage over a

feedforward Neural Network at the current time. However, exactly how each of the "important" features affect the probability of reoffence is quite difficult to determine from both of these models without examining a large variety of test cases. This is due to the fact that the interactions between factors in all three algorithms can be quite complicated, with multiple factors often affecting the output of a single tree at once. However, through repeated prediction and examination of these results, as well as the use of policing intuition, the individual effects of each variable on the predicted probability of reoffence should become clear.

Through an examination of the Live Test results, it can be concluded that some care must be taken in the implementation of the Random Forests model on live test data. Firstly, it must be ensured that the data is appropriately balanced. This can be achieved in many ways and may need more than one approach, depending on the nature of the historical data - if it can be assumed that the factors affecting re-offending are unlikely to change over time, then more reoffender data can be added from further back in the dataset. If not, there is the possibility of extending the time window (in order to pick up more long-term reoffenders) or altering the sampling approach. The appropriateness of each of these techniques may change over time and therefore may need to be altered as the police make use of the model. Secondly, it must be considered how effective the model is likely to be if the factors affecting reoffending do change dramatically in the real world - appropriate monitoring of changes in reoffender behaviour must therefore accompany the use of this model in a policing context, such that if the model becomes outdated, it is known how it may be updated.

2.6.2 Limitations

A general limitation present in all of the models is the assumption of independence between each of the data records present in the dataset - while we have made this assumption on reasonable grounds (stated earlier within this chapter), it is possible that the non-independence of dataset records will lead to biased performance estimates. If, for example, different records from the same individual are split across training and test datasets, the model may overperform on these records and underperform on other records, which are not split across the two sets. Testing the

model's sensitivity to this is difficult, but should be completed if the police believe that this is likely to significantly impact the performance of the model in a live setting.

One unfortunate limitation that the Dyfed-Powys datasets possess is the complete unavailability of offender data prior to the start date, 2008 (and in the second case, 2011). As such, it is entirely possible that an offender considered to be a "first-time offender" in this dataset (i.e. an offender not considered to have any previous offences) is not actually a first-time offender at all and has committed a number of offences prior to 2008. While this information is unavailable due to changes in data collection methods, prediction issues with cold-start problems akin to those presently found in the problems with this dataset are largely unavoidable; in fact, a number of these problems may well not be cold start at all! With an offender's offence history being shown to be such a strong predictor of their probable reoffence, it is clear that missing out on this crucial data is also likely to cause issues with prediction in the long run.

Another limitation that this dataset possesses is a lack of prison information. It is unclear from the information available in this dataset whether or not an individual offender is incarcerated following an offence, or how long the offender was incarcerated for. As such, the predictions of an offender's likely recidivism within 3 years of a certain offence are unable to include or exclude any jail sentences that the offender may be subject to during this monitoring window. This introduces considerable difficulties in the prediction of reoffence, as without information as to whether or not an offender was incarcerated, it cannot be decided whether or not they should be considered to be "removed" from the dataset at this time. Furthermore, without this prison data, it is also unknown if the individual is subject to any sort of probation period, which may well alter their likelihood of reoffence within this time period. With this crucial information not present in the dataset, it can be said that it is likely difficult to attain higher accuracy scores in the prediction of an individual's likelihood of reoffence at this time.

Moreover, Dyfed-Powys also faces limitations in their location data; should an offender move location between offences, or in fact at any point following an offence,

this move is not recorded. With the level of anti-social behaviour in the offender's home location being a large factor in whether or not the individual will reoffend, it is clear that having incorrect information here could lead to an incorrect prediction down the line. In addition, should the offender move to a location that lies a large distance outside Dyfed Powys' jurisdiction, it could be difficult for Dyfed-Powys to track the offender's activity, meaning that it is entirely possible that an offender could be falsely recorded as a non-reoffender when in fact they have been committing further offences outside of the Dyfed-Powys area.

2.6.3 Further Research

While these three models do represent the best quality binary classification models tested on this dataset so far, there are still many possible avenues to explore from a classification or probability estimation point of view. While these will not be the focus of this thesis, it is still possible that a different binary classification model, or a differently-constructed type of feedforward Neural Network, may be able to better predict the occurrence of reoffences on this dataset. However, with the current limitations on pre-2008 offender data in place, it is entirely possible that it is difficult, if not impossible, to attain a much better level of predictive accuracy than the current level.

The outperformance of Random Forest over XGBoost is surprising, given the state of the art nature of the XGBoost algorithm. Further parameter optimisation, including optimisation of the tree depth parameter and regularisations, would likely push the XGBoost algorithm into first place. However, the more complex the build and run processes of the model become, the more pressing the issue of comprehensibility and deployability also becomes. Without a coherent and fully automated parameter tuning script that has been tested within a clearly defined space over several runs, making use of a properly optimised XGBoost model in this context could prove too taxing for staff without a data science or statistical background. As such, we have left this open to further research, should Dyfed-Powys be willing to invest in this solution at a future date.

Another small extension that could be made to this work is to make use of the

Youden statistic. Some of the models within this chapter have suffered from imbalances between sensitivity and specificity and may benefit from the addition of an analysis or training approach based on this statistic.

Chapter 3

Survival Analysis for Reoffence Prediction

3.1 Survival Modelling for Recidivism

While it is certainly useful to know whether or not an individual offence is likely to lead to a reoffence from the same offender within a three-year period, it can also be said that it is much more useful for a police force to know when in that three-year period (or beyond), and with what probability, such a reoffence is likely to occur. With a more granular set of temporal predictions at their disposal, Dyfed-Powys police can make more informed decisions about when it is prudent to start and stop monitoring an individual offender. From a cost point of view, this is incredibly beneficial; if an individual offender is very likely to offend within the first month following their offence and very unlikely to offend thereafter, it makes little sense for the police to continue monitoring such an offender beyond the first month. Moreover, if a crime is reported in or near an area in which several recorded criminals are resident, these probabilities of reoffence can be used to estimate which of these offenders is most likely to have committed the crime.

The aim of this chapter, therefore, will be to predict (from the series of factors outlined in Chapter 2) how long it is likely to take for a reoffence by the same offender to occur following their most recent offence. As such, the prediction that will be produced is an estimate of the *time to an event*, where the *event* is represented by a reoffence committed by the individual in question. Should a reoffence occur by

the end of the monitoring period, this will be considered to be the occurrence of an event, or a "failure". Should an reoffence not occur before the end of the monitoring period, this will be considered to be a "censored event", whereby nothing is known about the subject after the time of censoring. As such, by considering a reoffence to be a "failure" and a lack of reoffence to be a "censored event", this time to event prediction problem can be taken to be one of survival, where the "survival" of an offender refers to the time it takes for that individual to either reoffend or be censored.

In particular, the focus of this investigation will be to assess the effects of various variables on the time it takes for an individual to commit an offence (or otherwise). Therefore, in keeping with the previous chapter, since each crime will be considered separately, only a single event will be considered for each crime and after this event, the offender in question will be seen to have exited the monitoring period. As survival analysis is a well-researched topic that encompasses many types of problems in many different areas of research, this chapter will begin with an overview of related work in tackling survival problems for criminal datasets, then continue with a discussion of these methods and how they may be appropriately used to predict the survival of offenders in this dataset.

3.1.1 Related Work

A great deal of research has already been undertaken in this field from a criminology perspective, mainly focusing (as was the case with the classification of offenders) on the relative survival of specific populations of offenders. As in the binary classification case, the act of reoffending is often defined as simply the act of committing a further crime. In some other cases, however, the act of committing a reoffence is considered to be a return to prison [82]. This distinction seems to be at the discretion of the researchers involved, with this consideration largely depending on the focus of the individual study. As such, wildly varying success rates in these predictions of survival were reported, depending entirely on the factors, population and scope involved.

Several different models have been put forward to model the survival of offenders in this context, most often following their release from prison. One of the most

popular models used for this purpose is the Cox Proportional Hazards [23] model, a well-known class of proportional hazards model. Like other proportional hazards models, this model aims to relate the time to event in a multiplicative way with one or more factors that may be associated with this quantity of time. At the time of writing, the Cox Proportional Hazards model has been shown to be instrumental in investigating certain issues within the field of recidivism. Examples of this include the notion of gender bias [6] as it relates to recidivism, the effect of an individual's employment status [88] and the effect of racial disparity on recidivism rates [47]. However, this model seems to be best suited to problems for which the number of factors being investigated is reasonably small and the purpose of the investigation relatively specific.

For many typical police datasets, which are often comprised of a large number of diverse predictors, the Cox Proportional Hazards model may be overwhelmed due to the proportional hazards assumption not holding for these datasets. As such, it has been necessary to investigate methods for which these assumptions need not apply. Examples of research making use of this algorithm include an investigation of the factors affecting the survival of graduates from a bootcamp [5] and the survival of population of drug offenders [39]. In recent times, the extension of the Random Forests algorithm for survival modelling [43] has seen an upsurge in popularity, due to its lack of reliance on a probability distribution with a fixed set of parameters and ability to handle large, complex datasets. In many cases, the offenders in question have already been incarcerated [95]. Again, the research most often focuses on predicting the survival of a small group of individuals determined to pose a risk to society, such as mentally disordered offenders [68], or to investigate the effect of a limited number of variables on the survival of individual offenders.

3.1.2 Discussion and Conclusion

On the inspection of this dataset, it was decided that the Random Forests algorithm as adapted for survival analysis provides a good basis for a general model of offender survival. In fact, this algorithm's ability to handle several complex between potentially dependent variables is outlined in the original Random Survival Forest paper by Ishwaran et al. [43]. Another reason for this choice was to avoid issues

caused by the proportional hazards assumption, which assumes that the ratio of any two hazards are constant over time. This assumption can be difficult to deal with in datasets with a large number of factors where the effect of these variables on survival is unknown; under these circumstances, it is highly probable that the ratio of at least one pair of hazards will not be constant over time.

In this particular problem, due to not all of the reoffenders having committed a further offence by the last date in the dataset, a number of censored times to event will be present within the dataset. For these times, it is known that the individual "survived" (i.e. did not reoffend, or at least did not get caught reoffending) for at least a certain length of time. As it is known that the offender "survived" at least that length of time, but it is uncertain when outside that time window they "survived" until without committing a further offence, these censored observations are defined to be *Right-Censored*. Moreover, as the observation period ends at a predetermined time (either at 31/12/2014 at 1:00pm or at 1095 days since the previous offence, whichever is sooner), these observations are also subject to *Type I Censoring*. This type of censoring occurs when an experiment or observation period stops at a particular time, leaving observations as either having "failed" (reoffended) before that date or "survived" (not reoffended) as far as is known beyond that date. As the current implementation of Random Forests for survival was designed to accept right-censored survival data of either censoring type, this further confirms the algorithm's suitability for use in this dataset.

To be in keeping with the classification models used in Chapter 2, the factors to be used for prediction in this model will be kept the same as in the classification model. Once again, TIC offences and offences that have not led to any kind of prosecution (as defined in Chapter 2) will not be considered to be reoffences and will therefore be excluded from the dataset as a whole before entry into the model. A description of the method behind this algorithm and the parameter possibilities that have been chosen to test as part of this investigation will be outlined in the following section.

3.2 Algorithm for Prediction

In this dataset, the Random Forests algorithm will be used to produce predictions of offender survival in the following way:

1. For each of a series of m crimes $c_1, ..., c_m$, there is a corresponding set of survival times $y_1, ..., y_m$. The n potentially predictive factors attached to this crime are then denoted by a set of values $x_1, ..., x_n$. The form of the dataset is given in Table 3.1 below:

Table 3.1: Dyfed-Powys Dataset Example: Survival Model. See Appendix for descriptions and explanations of independent variables.

ID	x_1 AgeCommitted	x_2 BurgValue	...
c_1	54	0	...
c_2	17	290	...
c_3	29	0	...
...	...	...	...
c_{m-1}	33	5000	...
c_m	21	25	...

ID	x_{n-1} MDAClass	x_n MultipleOffences	y Survival Time (Days)
c_1	A	0	$y_1 = 200$
c_2	U	1	$y_2 = 550$
c_3	U	0	$y_3 = 10$
...	...	...	...
c_{m-1}	C	0	$y_{m-1} = 375$
c_m	U	0	$y_m = 100$

2. Since some of these survival times y_i correspond to "deaths" (i.e. the offender having committed a reoffence at time y_i), and others correspond to "survivals" (i.e the offender not having committed a reoffence by time y_i), it also needs to be indicated whether or not a variable is right-censored at y_i. As such, the corresponding indicators of censoring for each crime y_i are denoted by $\delta_1, ..., \delta_m$, where $\delta_i = 1$ if the offender has committed a reoffence at time y_i and $\delta_i = 0$ has "survived" until time y_i without committing a reoffence.

3. Select a number of Decision Trees, B, to be grown for the forest. For each Decision Tree $b \in B$, sample, with replacement, a subset of m crimes $c_1, ..., c_i$ $i < m$.

4. Select a subset of the explanatory variables $x_1, ..., x_j$ $j < n$ from the n ex-

planatory variables within the dataset. In this case, j will be set to its default, $\sqrt{n}$ rounded to the next largest integer, then later tuned using the same Grid Search method as detailed in Chapter 2.

5. By recursively splitting the dataset into subsets so that one parent node splits into two separate child nodes, considering all j available features in the bootstrap sample for that tree and all j_v split values for that feature, either the feature $j*$ and the value of that feature j_{v*} that maximises survival difference in the child nodes according to a selected statistic (Logrank, C, Maxstat), or a random value from those available to split the sample at the parent node (Extremely Randomised Trees) will be chosen.

6. Using this method, grow each Decision Tree $b \in B$ to its maximum depth, i.e. keep splitting the data under the constraint that a terminal node should have no less than a predefined number $d^0 > 0$ of unique deaths, determined by the *min.node.size* parameter under ranger.

7. Once each Decision Tree has been grown, the individual survival times for the offender following each crime c_i within the dataset can be predicted. These predictions are produced from the set of terminal nodes H of each decision tree, i.e. the most extreme set of nodes in each of the trees grown by the algorithm, in the following way:

7.1. For a terminal node $h \in H$ containing T survival times, let the subset of the survival times in the original dataset that exists at node h be denoted by $y_{1,h}, ..., y_{T,h}$ and the corresponding censoring times by $\delta_{1,h}, ..., \delta_{T,h}$.

7.2. The unique reoffence times, i.e. the list of all possible survival times in that node for which $\delta_i = 1$, $t_{l,h}$, can then be defined to be a sequence $t_{1,h} < t_{2,h} < ... < t_{T,h}$.

7.3. Furthermore, defining $d_{l,h}$ to be the number of deaths at time $t_{l,h}$ and $s_{l,h}$ to be the number of individuals at risk, an estimate of the cumulative hazard function at node h, as defined by the Nelson-Aalen estimator, will be calculated:

$$\hat{A}_h(t) = \sum_{t_{l,h} \leq t} \frac{d_{l,h}}{s_{l,h}} \tag{3.1}$$

An illustration of this concept is detailed in Figure 3.1 below.

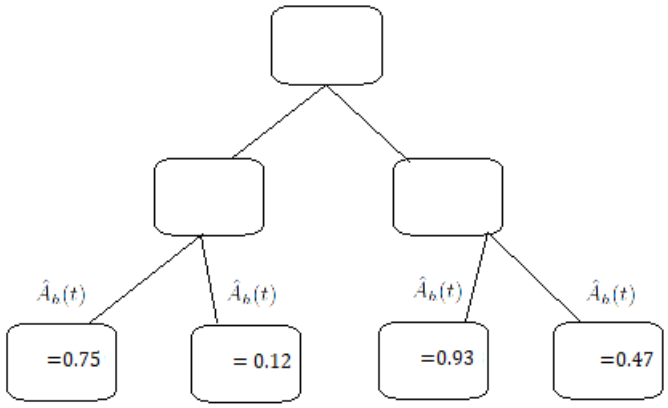

Figure 3.1: Tree Example 1: Nelson-Aalen Estimator, $\hat{A}_h(t)$, in Terminal Nodes

Every crime c_i that occupies the same terminal node h will have the same cumulative hazard function estimate. For example, if crimes c_{23}, c_{30} and c_{45} all occupy the leftmost node in the above tree, then they will all have the same cumulative hazard estimate $\hat{A}_h(t) = 0.75$. To determine, therefore, the estimate of the cumulative hazard function for any crime c_i, the crime is then simply dropped down the tree until it falls into one of the terminal nodes $h \in H$.

8. The cumulative hazard function $A(t|c_i)$ for this crime c_i will then be the Nelson-Aalen estimator for c_i's terminal node, defined as follows:

$$A(t|c_i) = \hat{A}_h(t) \quad if \quad c_i \in h \tag{3.2}$$

9. Once cumulative hazard function estimates $A_b^*(t|c_i)$ have been produced for each tree $b \in B$, the cumulative hazard function is averaged across each of the B decision trees to obtain the ensemble cumulative hazard function.

For a crime c_i, the ensemble cumulative hazard function $A_e^*(t|c_i)$ is as follows:

$$A_h^*(t|c_i) = \frac{1}{B} \sum_{b=1}^{B} A_b^*(t|c_i) \tag{3.3}$$

Assume that the following two trees exist in addition to the tree shown in Figure 3.1:

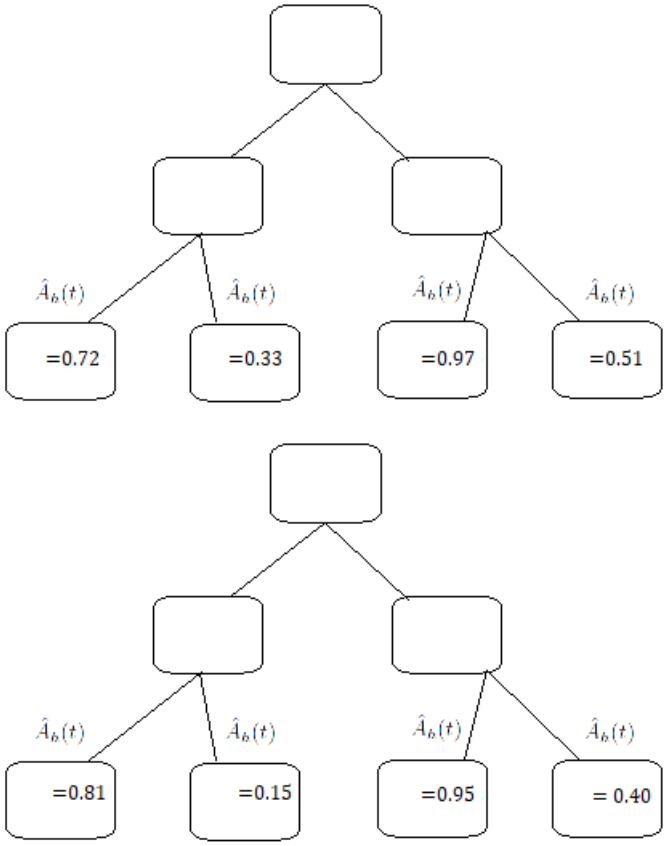

Figure 3.2: Tree Examples 2 and 3: Nelson-Aalen Estimator, $\hat{A}_h(t)$, in Terminal Nodes

In this case, cumulative hazard estimates and ensemble cumulative hazard are calculated as follows:

Table 3.2: Nelson-Aalen Estimators of Cumulative Hazard $\hat{A}_h(t)$ in each Terminal Node and Ensemble Cumulative Hazard $A_e^*(t|c_i)$

Tree	Node 1	Node 2	Node 3	Node 4
Tree $b = 1$	0.75	0.12	0.93	0.47
Tree $b = 2$	0.72	0.33	0.97	0.51
Tree $b = 3$	0.81	0.15	0.95	0.40
Ensemble (Average)	0.76	0.2	0.95	0.46

For each tree $b \in B$, $m - i$ crimes from the training set are not included in the bootstrapping process. These crimes are known as the out of bag (OOB) samples and can be used to validate the effectiveness of the training data's predictions on an unseen dataset. As such, it is also of interest to generate the cumulative hazard

function and corresponding survival probability estimates for the out of bag samples. These estimates are calculated as follows:

If a crime c_i is an out of bag sample for tree b, set $I_{i,b} = 1$ and if otherwise, set $I_{i,b} = 0$. The out of bag ensemble cumulative hazard function is then calculated as follows:

$$A_h^{**}(t|c_i) = \frac{\sum_{b=1}^{B} I_{i,b} A_b^*(t|c_i)}{\sum_{b=1}^{B} I_{i,b}} \tag{3.4}$$

Now that it has been determined how trees will be grown and how the survival function of an individual crime will be determined, an outline of how these estimates of survival will be evaluated against the actual survival times for the training, test and validation datasets will be provided below.

3.2.1 Split Rules

In order to provide a complete overview of the possibilities inherent within this method, four different variants of the Random Forests algorithm will be tested, which provide four different perspectives on how an individual Decision Tree within the forest should be grown. These variants, which all alter the rule that decides the values j_v of a feature j at which a node should be split, are all available within ranger's [93] R implementation, which has been chosen for both its relative speed and flexibility at the time of writing.

Split Rule 1: Log-rank Statistic (logrank) The first variant of this algorithm that will be tested uses the log-rank statistic, a special case of the Gehan statistic [31], to decide at which variables, and at which value of these variables, a Decision Tree's node would be optimally split. In this implementation, the nodes are split by maximising the log-rank statistic, which compares the relative hazard of two nodes of a tree for each feature value j_v in order to maximise the difference in survival times between these nodes. As the split rule originally recommended by Ishwaran et al. [43], this can be considered to be the 'default' option. With the use of this split rule being based on the proportional hazards assumption, however, concerns have been expressed with regards to its use in this context.

Split Rule 2: Extremely Randomised Trees (extratrees) The second variant that will be tested is known as Extremely Randomised Trees [33], which (true to its name) chooses the values j_v for the split at each Decision Tree node randomly, producing a series of completely randomised trees. Compared to the log-rank statistic method, the Extremely Randomised Trees method can address some issues with bias (i.e. overfitting to training data) and is usually much faster to compute, but may underfit to the training set.

Split Rule 3: Harrell's C Statistic (C) The third variant, as proposed by the authors of the ranger package [76], uses Harrell's C statistic [46], another special case of the Gehan statistic, to determine the optimal split at each node. The use of this statistic for the purpose of building optimally split trees is very intuitive, as this statistic is commonly used to evaluate the predictive accuracy of the Random Forest. Compared to the log-rank method, this method has often been shown to perform favourably if the dataset contains a lot of censored data or comparatively more continuous independent variables, but for large or noisy datasets (i.e. datasets that contain a lot of uninformative independent variables) there can be some performance issues.

Split Rule 4: Maximally Selected Rank Statistics (maxstat) The fourth variant, as similarly proposed by the authors of the ranger package [92], uses maximally selected rank statistics [50], an alternative to the linear rank statistics employed in conditional inference forests, to determine the optimal splits for each node. This method, which uses p-value approximations to determine splits instead of the usual conditional Monte-Carlo methods, avoids the log-rank statistic's tendency to be biased towards selecting variables with a greater number of possible split points [85] and also offers some improvement in algorithm runtime - this is due to the p-value approximation used in the Ranger package, which is detailed in the original paper.

3.3 Evaluation

The evaluation of this method will begin by plotting the survival, cumulative hazard and hazard functions (to be defined in Section 3.4) for this dataset. Examining

these functions, which will likely be most useful in the everyday use of this survival algorithm, allow the police to compare the relative survivals and hazards of different offenders as they enter the database, then make judgements on how these offenders should be treated. From a policing perspective, therefore, it is incredibly important that these functions are compared and examined, and suggestions are offered as to how they might be used by Dyfed-Powys police to assess the relative reoffence risk of future offenders.

From both our perspective and the perspective of Dyfed-Powys police, however, it is also important that the overall fit and predictive accuracy of the model are assessed appropriately. The method by which this will be completed, as well as the metric used to assess the level of error on the dataset, will be discussed in the following section.

3.3.1 Predictive Power

In the ranger package, as suggested in the original paper by Ishwaran et al., the prediction error is estimated by 1 - Harrell's C-index [46], so (as would be intuitively expected) the lower the prediction error, the better the survival distribution prediction is considered to be. This index, essentially, tests whether predictions of comparatively longer survival times actually result in comparatively longer survival times and conversely, whether predictions of comparatively shorter survival times actually result in comparatively shorter survival times. This statistic is related to the area under the ROC curve and can be considered to be the equivalent of calculating the AUC score for a survival problem.

The steps by which the C-index for this dataset can be calculated are defined below:

1. Form all possible pairs of crimes $c_1, c_2, ..., c_m$ alongside their corresponding indicators $\delta_1, \delta_2, ..., \delta_m$.

2. For a pair of crimes c_i, c_j with corresponding actual survival times T_i, T_j, omit the pair from the calculation if $T_i = T_j$ and $\delta_i = \delta_j = 0$, or $Ti < Tj$ and $\delta_i = 0$. All pairs c_i, c_j that are not omitted under these criteria are then denoted as permissi-

ble pairs. Let the total number of these permissible pairs be denoted by Permissible.

3. For each permissible pair where $T_i \neq T_j$ and their predicted survival times P_i, P_j, count 1 if $T_i < T_j$ and $P_i < P_j$, or conversely if $T_i > T_j$ and $P_i > P_j$. Count 0.5 if $P_i = P_j$.

4. For each permissible pair where $T_i = T_j$ and $\delta_i = \delta j = 1$, count 1 if $P_i = P_j$. Otherwise, count 0.5.

5. For each permissible pair where $T_i = T_j$ and $\delta_i \neq \delta_j$, count 1 if $P_i < P_j$ and $\delta_i = 1$, or $P_i > P_j$ and $\delta_j = 1$; otherwise, count 0.5.

6. Let the Concordance denote the sum over all permissible pairs.

7. The C-Index, C, is then defined by $C = \text{Concordance/Permissible}$.

Similarly to the AUC score, a C-statistic value of 0.5 indicates that the model is no better than random chance. A value of below 0.5, therefore, indicates that the model performs poorly and should not be used. A value of 0.6-0.7 would be generally considered to be a common result for survival data, while a C-Index value of above 0.7 would indicate that the model's performance is good.

3.3.2 Variable Importance

Once again, while computing the concordance index of the model gives information pertaining to the performance of the model, this measure cannot give insight into the effect each of the independent variables have on the predictions produced by the model.

Like the previously described classification and probability forests, it is possible to calculate a measure of importance for each independent variable input into the model. In R's ranger package, the only option for calculating this measure is the aforementioned bias-corrected permutation importance. This measure was previously used in Chapter 2 to calculate the variable importance for the independent

variables input into a classification forest and is calculated in the same way, by randomly permuting the values of the variable over all b trees in the forest and measuring the resulting increase in error. Again, the influence of correlated features on the outcome of this model is removed.

3.4 Results

3.4.1 Original Dataset

Hazard, Cumulative Hazard and Survival Functions

In order to visualise the output of the algorithm, plots of the estimated survival function, hazard and cumulative hazard functions will be produced for each individual crime as they alter over a series of four-week periods. From a policing perspective, it is important to be able to visualise the relative survival of offenders in a clear and simple way, so that judgements can easily be made on when an offender is most likely to reoffend. These plots, of which four examples will be shown below, can be evaluated individually, as well as compared against one another in order to decide which offenders should be prioritised in terms of monitoring.

Survival Plots Firstly, the survival function plots for four different crime examples will be displayed. The survival function $S(t)$, is defined as:

$$S(t) = Pr(T > t) \tag{3.5}$$

where t is a time (in this case, representing a 4-week interval of time), T is a variable representing the time at which the offender commits a reoffence and Pr denotes a probability. Therefore, the survival function represents the predicted probability that the time to reoffence T is greater than a certain time t. If no reoffence occurs, then T will be infinite.

Evaluations will begin with a side-by-side comparison of the survival plots for two crimes that led to a reoffence, as follows:

1. The left-hand plot shows the survival function (as predicted by the Random Survival Forests algorithm) for a crime that led to a reoffence during the time pe-

riod $t = 0$, or within 4 weeks of the previous offence.

2. The right-hand plot shows the survival function for a crime that led to a reoffence during the time period $t = 20$, or between 76 and 80 weeks after the previous offence. The corresponding $S(t)$ plots are shown in Figure 3.3 below.

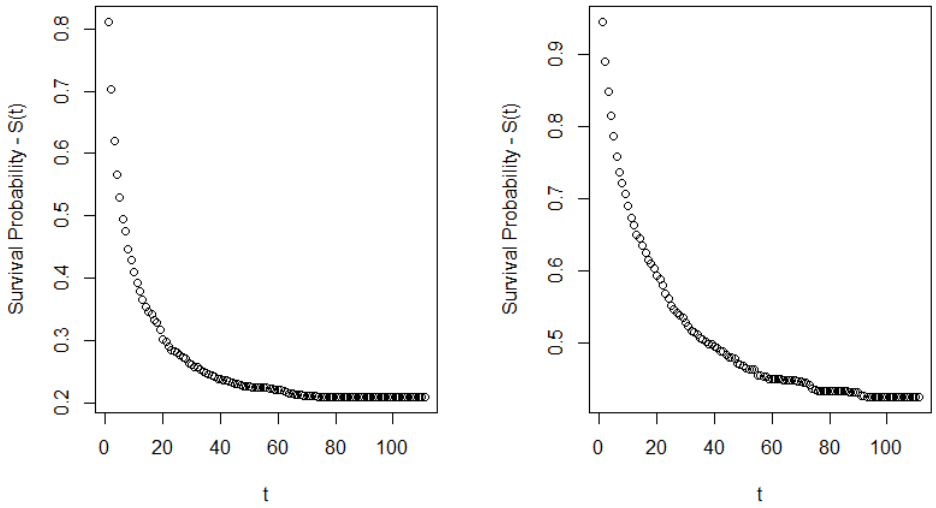

Figure 3.3: $S(t)$ Plots, Failures at t = 0 (left) and t = 20 (right). Please note that the scale of each graph is different.

The predicted $S(t)$ function on the left-hand survival plot decreases much more quickly and drops to a much lower value than the predicted $S(t)$ function for the right-hand survival plot (approx. 0.2 vs 0.4). Intuitively, therefore, the left-hand crime would be expected to lead to a reoffence much more quickly than the right-hand crime, which was exactly the case in these examples. While in both cases the eventual probability of reoffence is greater than 0.5, so it would be expected that both of these crimes will eventually lead to a reoffence, the offence that generates the left-hand $S(t)$ function corresponds to an offence that is likely to lead to a reoffence far more quickly than the offence that generates the right-hand $S(t)$ function. It is also more likely to lead to a reoffence in the long term.

While it is evident that these survival plots provide an intuitive way to predict the likely survival of an offender over time in the case that the offender does actually reoffend, it remains to be seen whether this is the case for censored offences. To

investigate this, two offences have been chosen for plotting in Figure 3.4 as follows:

1. The left-hand offence is censored at $t = 34$ with no further offences having been committed by the same offender at that time.

2. The right-hand offence is censored at $t = 77$, again with no further offences having been committed by the same offender at that time.

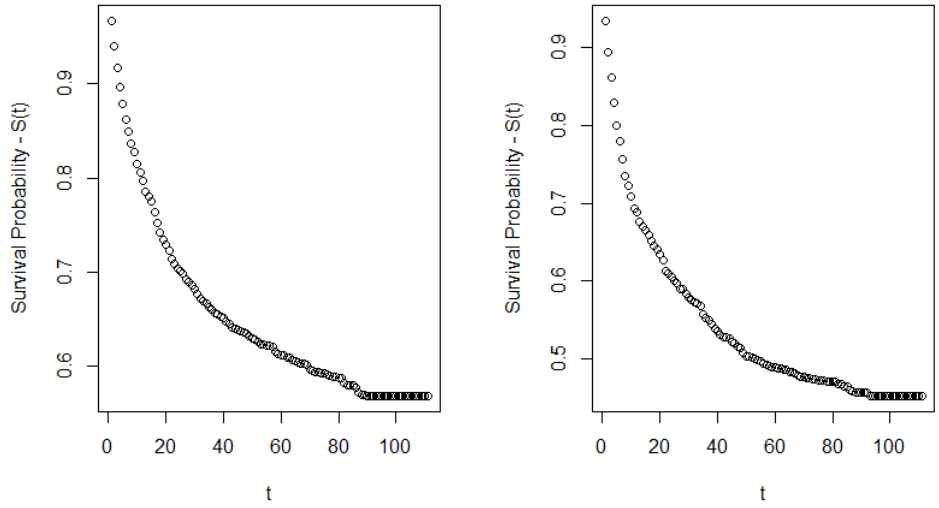

Figure 3.4: $S(t)$ Plots, Censoring at t = 34 (left) and t = 77 (right). Please note that the scale of each graph is different.

The likely survival of the offender decreases much less quickly than in the previous two examples, meaning that these offences are predicted to be much less likely to lead to a reoffence than the previously examined offences. When compared to the functions that led to a reoffence, a difference in the eventual predicted survival of the individuals whose survival conforms to this curve can be observed, although the right-hand plot in Figure 3.4 does eventually converge on a similar survival probability to that of the right-hand plot in Figure 3.3.

There are many possibilities as to why the survival function of a censored and an uncensored offence may be similar. Firstly, the censored offence may not have led to a reoffence just yet, although given that the offence has already reached $t = 77$ without leading to a reoffence, this unlikely to be the case - by that time, the cu-

mulative probability of survival has decreased sufficiently that the offender is likely to keep surviving (i.e. not reoffending) into the future. Secondly, it may be that the censored offender is in prison or otherwise unavailable for quite some time after committing the offence, with this data being missing but somehow inferrable from other variables in the dataset. Lastly, it may be that both of these cases are simply reasonably borderline cases, where the eventual survival of the offender could go either way. Whether this is the case or not remains to be seen.

Cumulative Hazard Plots Now that the survival functions of these four examples has been plotted, the cumulative hazard function $\Lambda(t)$ can be plotted for each of the four examples in turn. The cumulative hazard plot shows how the hazard of reoffending increases cumulatively over time, with the slope of the line between two points representing the increase in hazard, which can be thought of a measure of reoffence risk where the higher the hazard the higher the risk, between those two times. It is related to the survival function in the following way:

$$\Lambda(t) = -\log S(t) \tag{3.6}$$

This evaluation will begin with an examination of the $\Lambda(t)$ plots for those offences that led to a reoffence, shown in Figure 3.5 below.

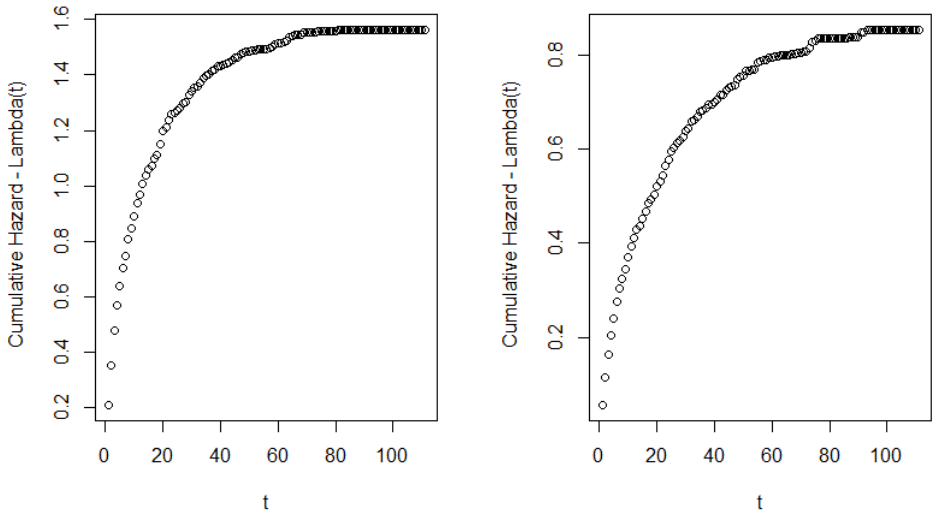

Figure 3.5: $\Lambda(t)$ Plots, Failures at t = 0 (left) and t = 20 (right). Please note that the scale of each graph is different.

Once again, the cumulative hazard increases much more sharply in the case of the offence that led to a reoffence at $t = 0$, which is as expected in this case. As before, the $\Lambda(t)$ function plateaus more quickly and converges on a comparatively higher cumulative hazard value.

To see whether similar conclusions can be drawn on censored data, refer to the plots in Figure 3.6 below.

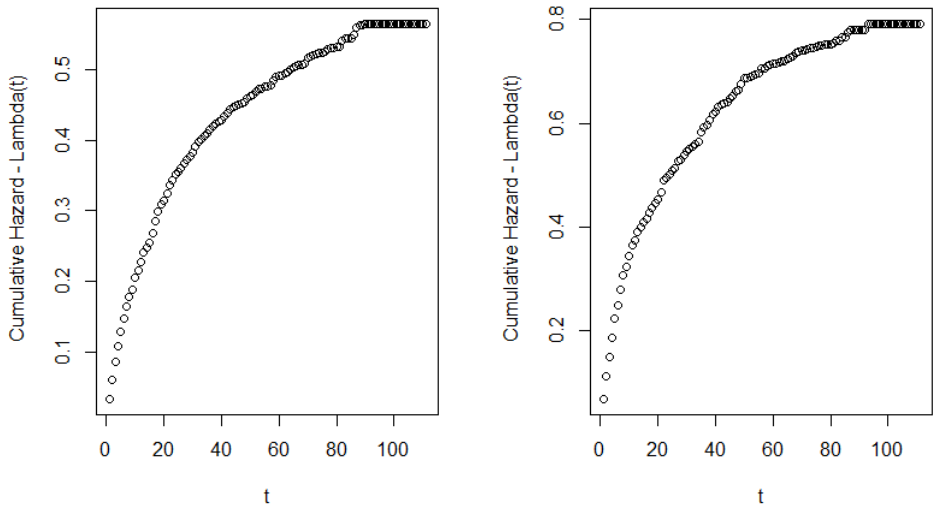

Figure 3.6: $\Lambda(t)$ Plots, Censoring at t = 34 (left) and t = 77 (right). Please note that the scale of each graph is different.

Again, the hazard rises more slowly compared to those offences that did lead to a reoffence. However, once again, the final value of $\Lambda(t)$ when regarding the $t = 77$ censored function is comparable to the $t = 20$ failure.

Instantaneous Hazard Plots While it is useful to see how the hazard rate increases over time, it can also be useful to look directly at the instantaneous hazard function. This function, as generated by the survival forest, is described as the event (reoffence) rate conditional on survival until at least a time t and is defined by the following function:

$$\lambda(t) = \frac{\delta}{\delta t}\Lambda(t) = -\frac{S'(t)}{S(t)} \tag{3.7}$$

Here, this instantaneous hazard $\lambda(t)$ can be plotted for all available values of t. These plots, beginning with the offences that led to a reoffence, are shown in Figure 3.7 below.

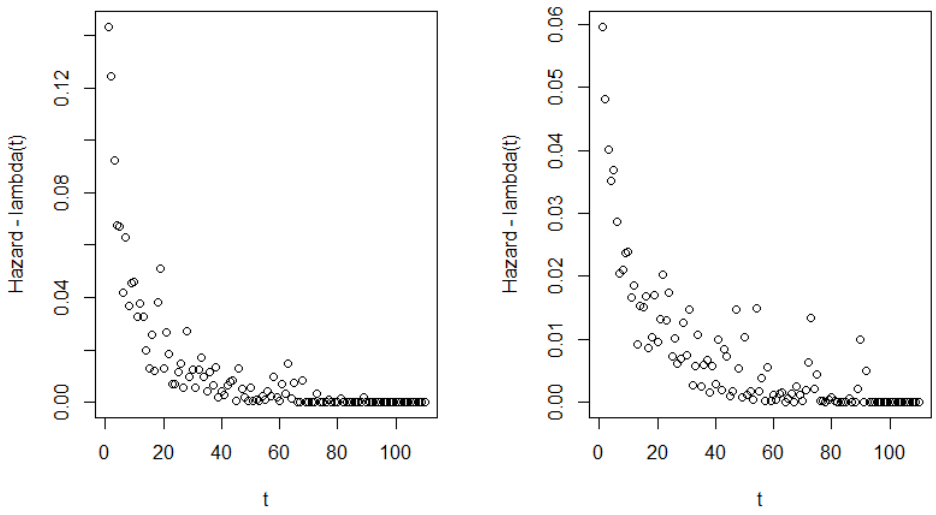

Figure 3.7: $\lambda(t)$ Plots, Failures at t = 0 (left) and t = 20 (right). Please note that the scale of each graph is different.

In both cases, the instantaneous hazard (i.e. the risk of reoffence at time t) is initially quite high and begins to drop off quite quickly. After $t = 4$ in the left-hand case and $t = 5$ in the right-hand case, it begins to decrease while fluctuating somewhat until a value of around 0.02 is reached. From this point, the hazard rate appears to fluctuate in both cases until around $t = 80$, where it remains very close to 0. From this, the hazard of a crime leading to a further offence is likely to be highest in the first few time periods following a crime and by the time it has been around 2 years since an offence was last committed by that offender, it is unlikely that they will commit any further offence. From this measure, it is unlikely that an offender will commit further offences many years after the date of their first offence.

In order to see what sort of hazard plots can be expected from a censored offence, the plots for the two censored offences detailed above can be constructed. These are shown in Figure 3.8 below.

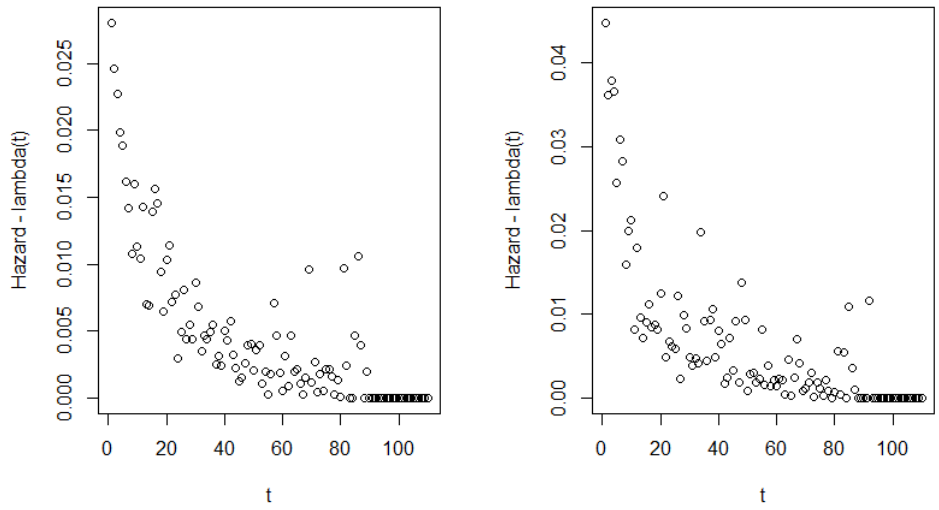

Figure 3.8: $\lambda(t)$ Plots, Censoring at t = 34 (left) and t = 77 (right). Please note that the scale of each graph is different.

Compared to the uncensored plots, the instantaneous hazard appears to fluctuate a great deal more than before - however, given that these plots are operating on a much smaller scale (the instantaneous hazard never reaches more than 0.03 on the first and 0.05 on the second plot), these fluctuations are likely no bigger or smaller than those in other plots. However, with the instantaneous hazard being much lower within each time window, the offence in question is predicted to be much less likely to lead to a reoffence.

In order to continue to assess the overall predictive performance of the model on the whole dataset, an assessment of the metrics chosen for evaluation on the training, test and validation sets has been provided below.

Metrics and Evaluation

Predictive Peformance and Runtime - All Offenders The metrics (Concordance Index) used to assess the predictive accuracy of the model on the training, test and validation sets are given below. As the runtime of the algorithms is likely to be a significant factor in choosing the split rule that best fits this dataset, a measure of the runtime in minutes has also been included alongside the error for the training, validation and test sets in each case.

Table 3.3: Random Survival Forest Evaluation Metrics. Split Rule = Logrank. Parameters: No. Trees = 100, No. Variables per Split = 13

Metric	Value
Runtime (Minutes)	19.6490
Training Set Error	0.1701
Validation Set (OOB) Error	0.2629
Test Set Error	0.2800

Here, the runtime required to grow the logrank trees is quite extensive - with a 19 minute runtime for a dataset of this size, bearing in mind that the size of the dataset is likely to increase at future dates, it is not possible to produce sufficiently fast predictions of survival time. With a training set error of 0.17 and comparable validation and test set errors of 0.26 and 0.28 respectively, this method produces a forest that overfits somewhat to the training data.

Table 3.4: Random Survival Forest Evaluation Metrics. Split Rule = Extratrees. Parameters: No. Trees = 100, No. Variables per Split = 13

Metric	Value
Runtime (Minutes)	1.9686
Training Set Error	0.1988
Validation Set (OOB) Error	0.2608
Test Set Error	0.2807

The runtime for Extremely Randomised Trees has significantly decreased compared to the log-rank split rule for a forest built under the same conditions. However, this decrease in runtime has come at the expense of the forest's fit to the training set, with the error on the training set increasing from 0.17 to almost 0.2. However, there is little (if any) change in the test set error. As such, it can be concluded that compared to the logrank split rule, the extratrees split rule overfits much less to the training set while producing similarly accurate predictions on the test set in a much shorter period of time. Therefore, for our purposes, it is preferable to the logrank split rule.

While comparable results were produced for the C-statistic split rule, they have been omitted here. This is due to the fact that splitting the nodes on the C-statistic slowed down the growth of trees so heavily that the entire process took significantly more than 24 hours to run. Due to this fact, given the increased computing power

constraints that will be present within a policing setting and the eventual increase in the number of training examples over time, using this split rule will not be conducive to producing a good working model.

The maxstat splitrule, however, did not have such issues with runtime. In fact, in this case, its runtime was found to be comparable to the extratrees splitrule. The errors produced by this method on the training, validation and test sets are shown in the table below.

Table 3.5: Random Survival Forest Evaluation Metrics. Split Rule = Maxstat. Parameters: No. Trees = 500, No. Variables per Split = 13, Alpha = 0.1, Minprop = 0.156

Metric	Value
Runtime (Minutes)	1.9904 mins
Training Set Error	0.2335
Validation Set (OOB) Error	0.2683
Test Set Error	0.2706

Using the maxstat splitrule to grow a random forest results in a greater training error, but comparable test and validation set errors. Given that the error on the training set is still relatively small and comparable to the error on the test and validation sets, this particular splitrule simply results in a much lower likelihood of overfitting to the training set than either of the logrank or extratrees splitrules.

Purely from the perspective of the concordance index and the comparative runtime of the algorithms, the best choice of splitrule for this dataset would either be the extratrees or maxstat splitrules, but it is entirely possible that there may be some differences between the performance of these splitrules when solely considering offences committed by first-time offenders. As such, it is worth investigating the relative performance of the Random Survival Forests algorithm as defined by each of the three appropriate splitrules. These results are contained in the following section.

Predictive Performance and Runtime - First-Time Offenders Only Briefly, the test set scores can be examined for trees grown via each of the three suitable splitrules on a test set comprised of only offences committed by first-time offenders. These are detailed below.

Table 3.6: Random Survival Forest Evaluation Metrics. Parameters: No. Trees = 100, No. Variables per Split = 13, Alpha = 0.1, Minprop = 0.156

Splitrule	Test Set Error
Logrank	0.3514
Extratrees	0.3571
Maxstat	0.3395

In general, once again, it is somewhat more difficult for the model to make accurate survival predictions on the test set when the test set is solely made up of offences committed by first-time offenders. While there does not appear to be a great difference in test set errors between the three methods, a Random Survival Forest grown using the Maxstat splitrule produces the lowest error on the test set and as such, this is the recommended splitrule for use in the live dataset. While the Extratrees splitrule may offer comparable performance on the entire dataset, the trees grown using the Maxstat splitrule fit the test data better when only first-time offences have been included within the test set.

Variable Importances Now that it has been decided which of the splitrules is most appropriate for this task, permutation importances can be generated for each of the variables to be considered for prediction. For further details as to how these permutation importances are calculated, refer back to Chapter 2 of this thesis.

Table 3.7: Variable Importances. Parameters: No. Trees = 100, No. Variables per Split = 13.

Variables	Permutation Importances
NoPreviousArrests	**0.0358**
PrevFine	**0.0325**
Outcome	**0.0149**
OffenceCat	**0.0100**
AgeCommitted	**0.0090**
Fine	0.0088
PrevViolence	0.0054
Off_Sex	0.0044
MultipleOffences	0.0040
Off_EducKS2_Pts	0.0038
Off_UR01IND	0.0032
Crime_UR01IND	0.0032
Crime_HealthLBW_Perc	0.0026
MDAClass	0.0026
Off_EmpBenefits_Perc	0.0025

As before, many of the variables that are deemed to be the most important in determining the survival of an offender following an individual crime are related to an offender's previous offence history. The most important variable, NoPreviousArrests, relates to the number of previous offences committed by the offender, while PrevFine relates to the level of harm (in terms of fines incurred) previously committed by the offender on society.

However, in comparison to the categorical model, factors not relating to an offender's offence history are considered to be a great deal more important. The outcome of the offence (as in the treatment of said offence after it was committed), the age at which the offence was committed and the actual high-level classification of the offence all contribute greatly to the survival of the model, as do many location-related factors. Surprisingly, however, the survival of offenders appears to be much less affected by the level of anti-social behaviour in the immediate area and more affected by the Urban-Rural classification of both the area in which the offender is resident and the area in which the crime is committed.

3.4.2 Live Test Data

Hazard, Cumulative Hazard and Survival Functions

Survival Plots Firstly, the survival function plots for four different crime examples will be displayed. A side-by-side comparison of the survival plots for two crimes that led to a reoffence is displayed in Figure 3.9.

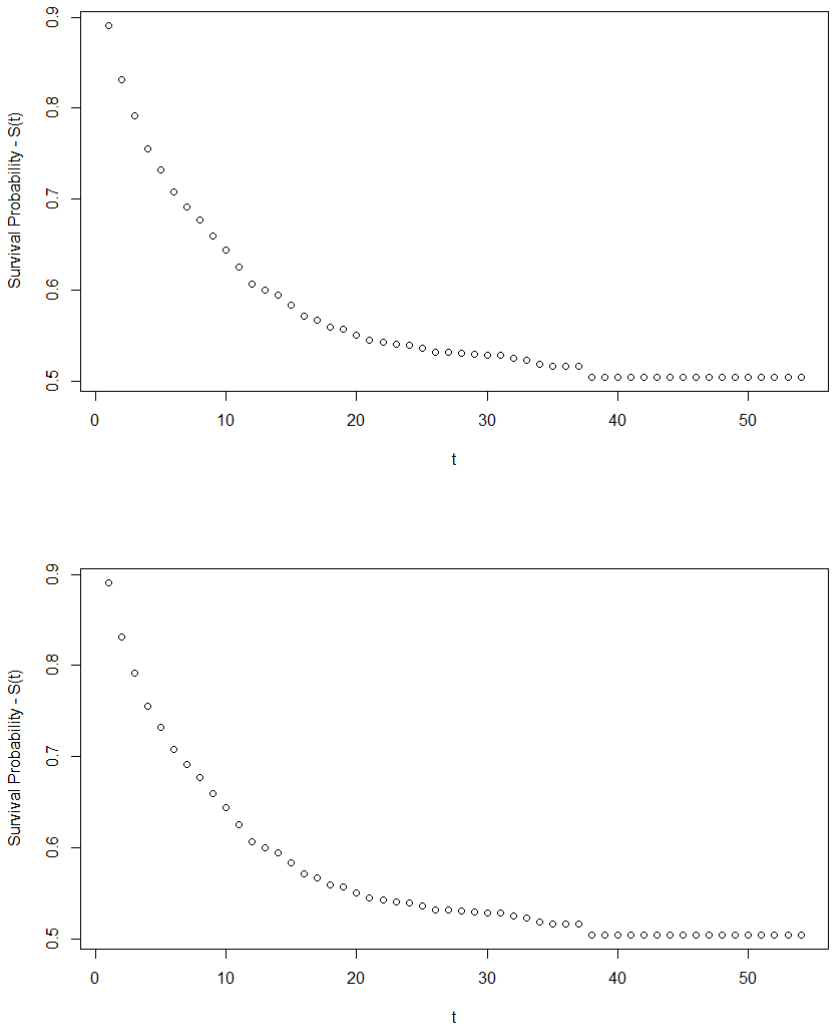

Figure 3.9: $S(t)$ Plots, Failures at t = 2 (top) and t = 30 (bottom). Please note that the scale of each graph is different.

Here, the two survival functions appear to be very similar - they start off with a 90% chance of "surviving" (i.e. not reoffending within) the first 4 weeks, which quickly drops to around 65% by the 40 week mark. Once again, the offender's probability of survival dropping sharply in the first few weeks and flattening out to around 50% by the time the offender has reached about 2 years without an offence.

The only difference between the two plots is that the first has a slightly steeper line than the second - it would therefore be reasonable to say that the first offender (who offended between 4 and 8 weeks post the original offence) would be slightly more likely to reoffend at an earlier time than the second.

The survival functions of two censored offences will now be examined. One has been committed 4 weeks prior to the end of the dataset (in this case, 31st December 2017), while the other has been committed over 2 years before the final censor date. The two plots are shown in Figure 3.10.

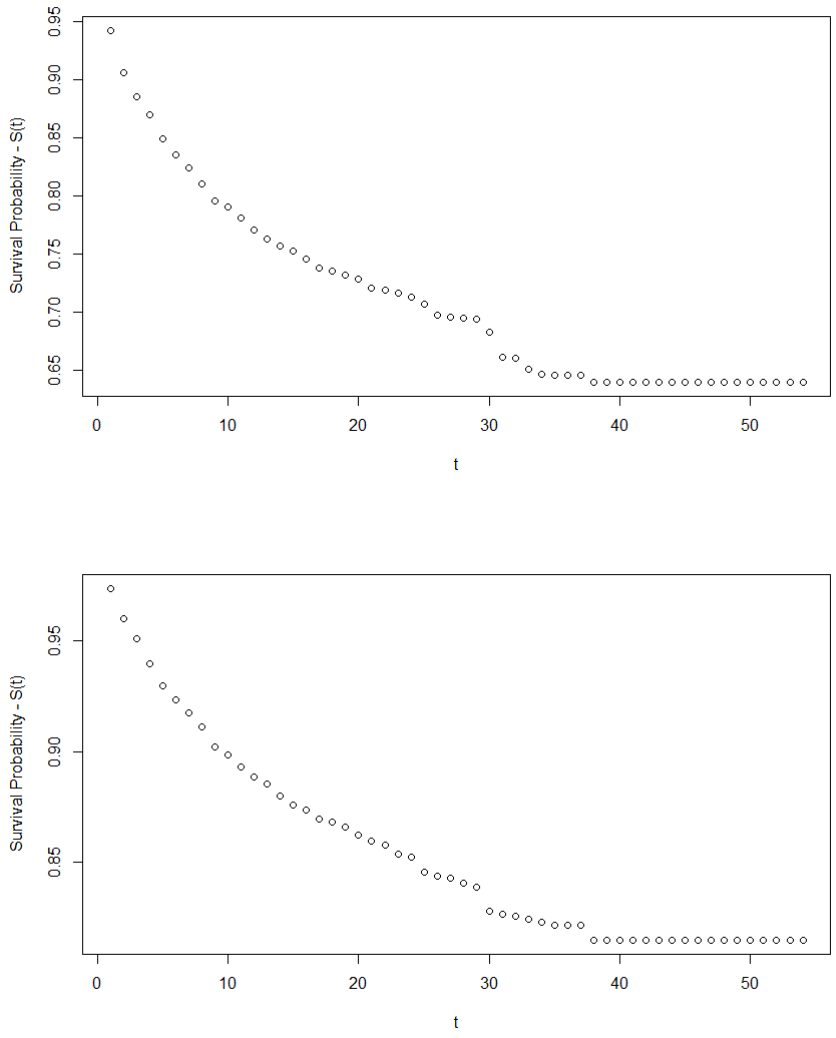

Figure 3.10: $S(t)$ Plots, Censored at t = 1 (top) and t = 26 (bottom). Please note that the scale of each graph is different.

A large difference can be observed between the two plots - the first censored individual has an eventual 65% chance of survival, whereas the second has an ap-

proximately 80% chance of survival. From these plots, it is unlikely the second offence (which has already reached approximately 2 years without resulting in any further offence activity) is going to result in a reoffence - the first offence is far more likely to lead to a reoffence both eventually and within the next 4 week period.

Cumulative Hazard Plots The cumulative hazard function $\Lambda(t)$ will now be plotted for each of the four examples in turn. The $\Lambda(t)$ plots for two selected offences that led to a reoffence are shown in Figure 3.11 below.

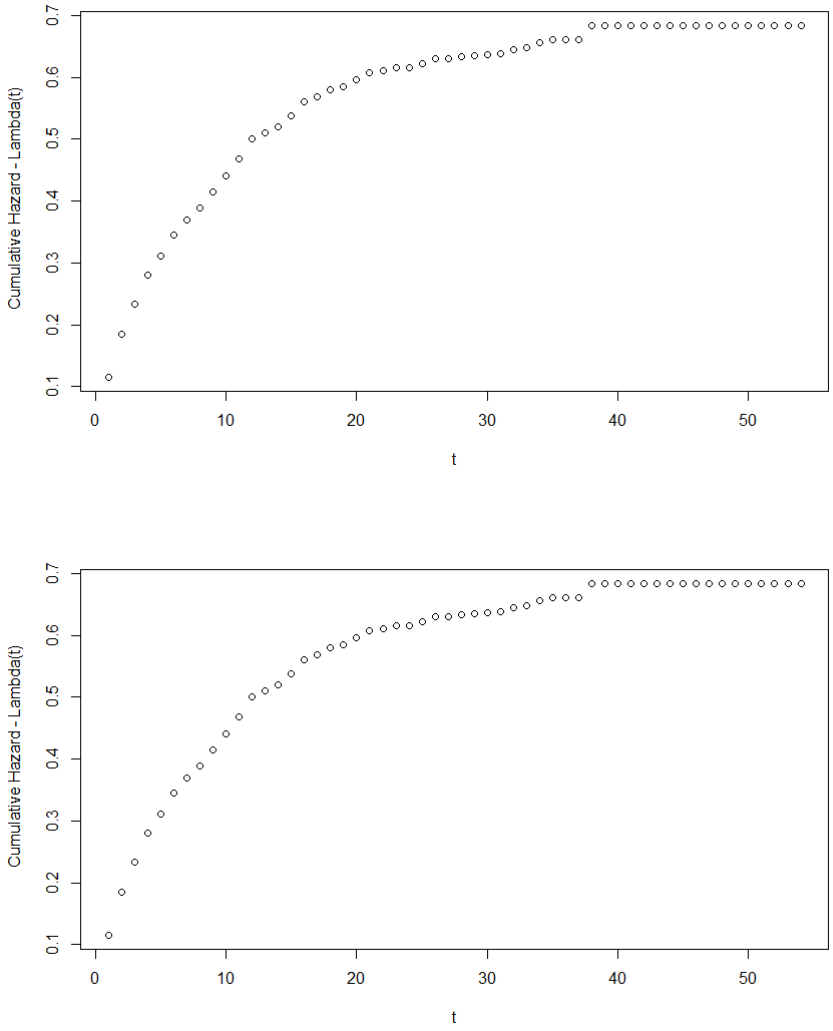

Figure 3.11: $\Lambda(t)$ Plots, Failures at t = 2 (top) and t = 30 (bottom). Please note that the scale of each graph is different.

Once again, very little difference is observed between the cumulative hazard for the offender who committed a reoffence between 4 and 8 weeks after the original offence and that for the offender who committed a reoffence over 2 years after the

original offence. A potential reason for this could be the differential sentencing treatment between the two offenders - the first offender could have been released with a fine (enabling them to be "in the community" to offend much more quickly) while the second could have been put into prison for the offence for a number of months or even years, rendering them unable to offend for that time. It could also be that these were simply very similar risks that have simply resulted in a reoffence at different times due to chance.

To see whether different conclusions should be drawn on censored data, refer to the plots in Figure 3.12 below.

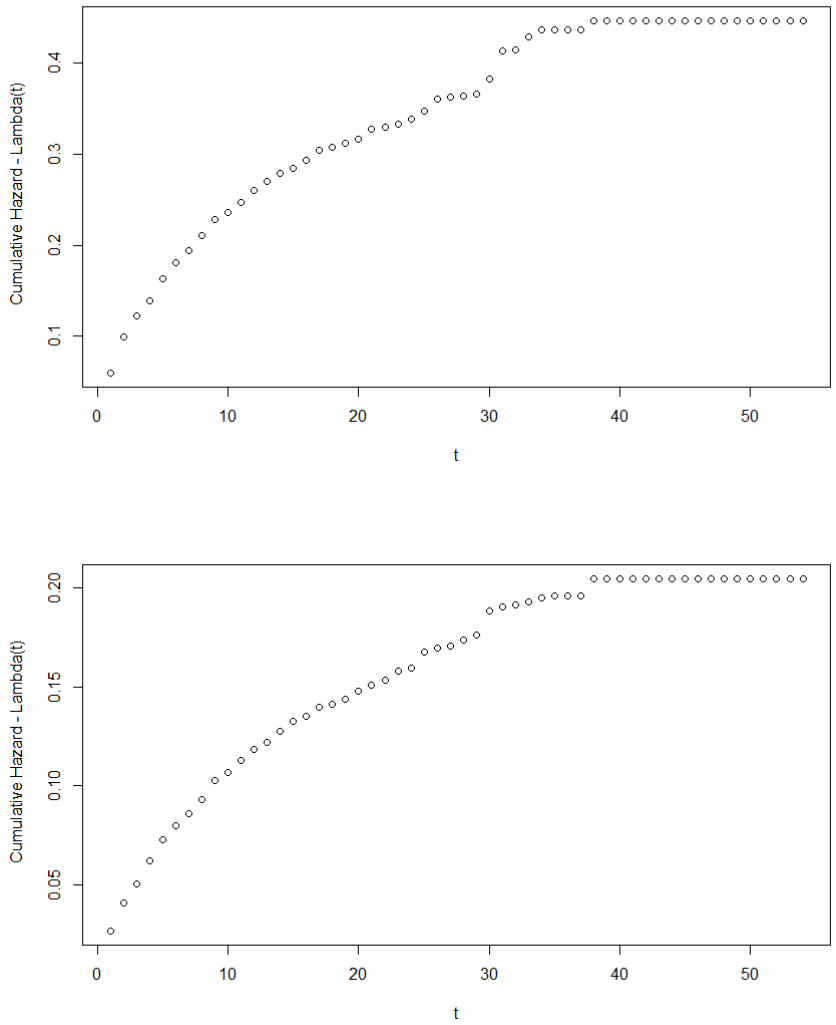

Figure 3.12: $\Lambda(t)$ Plots, Censoring at t = 1 (top) and t = 26 (bottom). Please note that the scale of each graph is different.

The two cumulative hazard plots, as expected, appear to be very different. Unlike

those offences that led to a reoffence, the cumulative hazards here are very different - the first offence appears to be much more likely to lead to a reoffence at every point in time

Instantaneous Hazard Plots Here, this instantaneous hazard $\lambda(t)$ is plotted for each of the available t values. These plots, beginning with the offences that led to a reoffence, are shown in Figure 3.13 below.

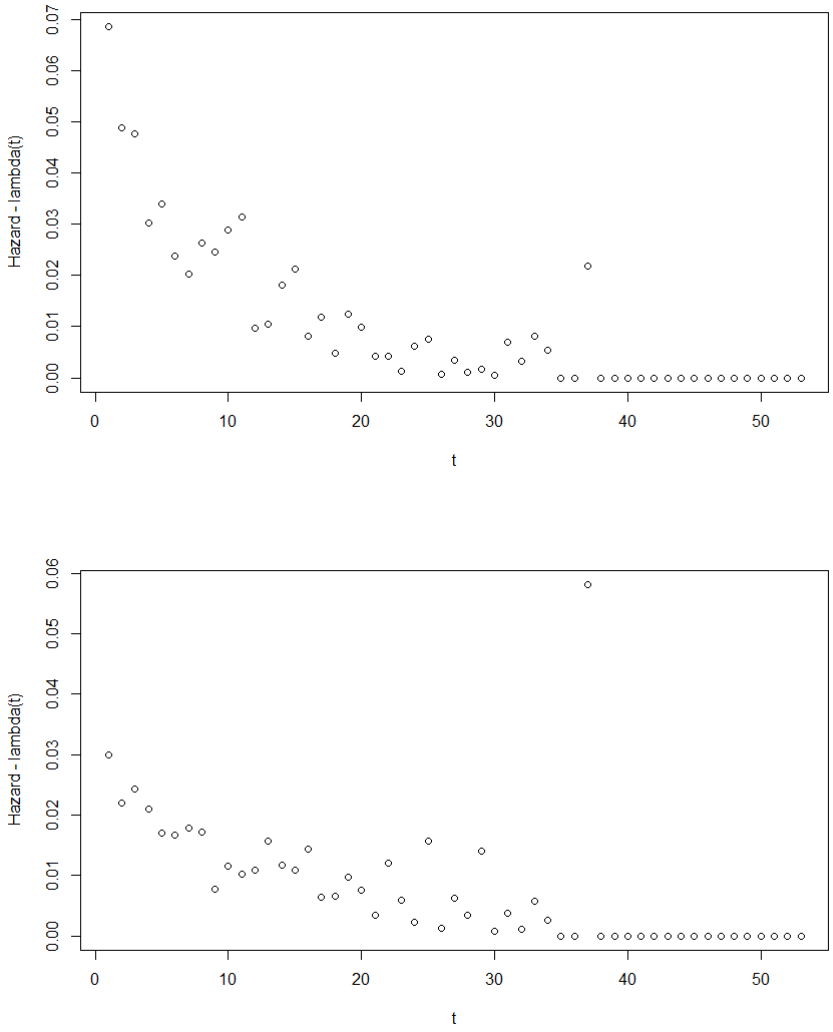

Figure 3.13: $\lambda(t)$ Plots, Failures at t = 2 (top) and t = 30 (bottom). Please note that the scale of each graph is different.

Here, large fluctuations in the instantaneous hazard are not present until around t = 20 (between 76 and 80 weeks since the original offence) and follow a relatively continuous decreasing pattern. The differences between the two crimes is more evident here and it is clear why the first offence may have led to a reoffence at t = 2

- the predicted instantaneous hazard of the first offence is much higher (0.06) at t = 2 than the instantaneous hazard of the second (0.02). The instantaneous hazard decreases much more sharply in the first case, but also starts much higher - this indicates that the first offence would be more likely to lead to an early reoffence than the second.

In order to see what sort of hazard plots can be expected from a censored offence, the plots for the two censored offences are shown in Figure 3.14 below.

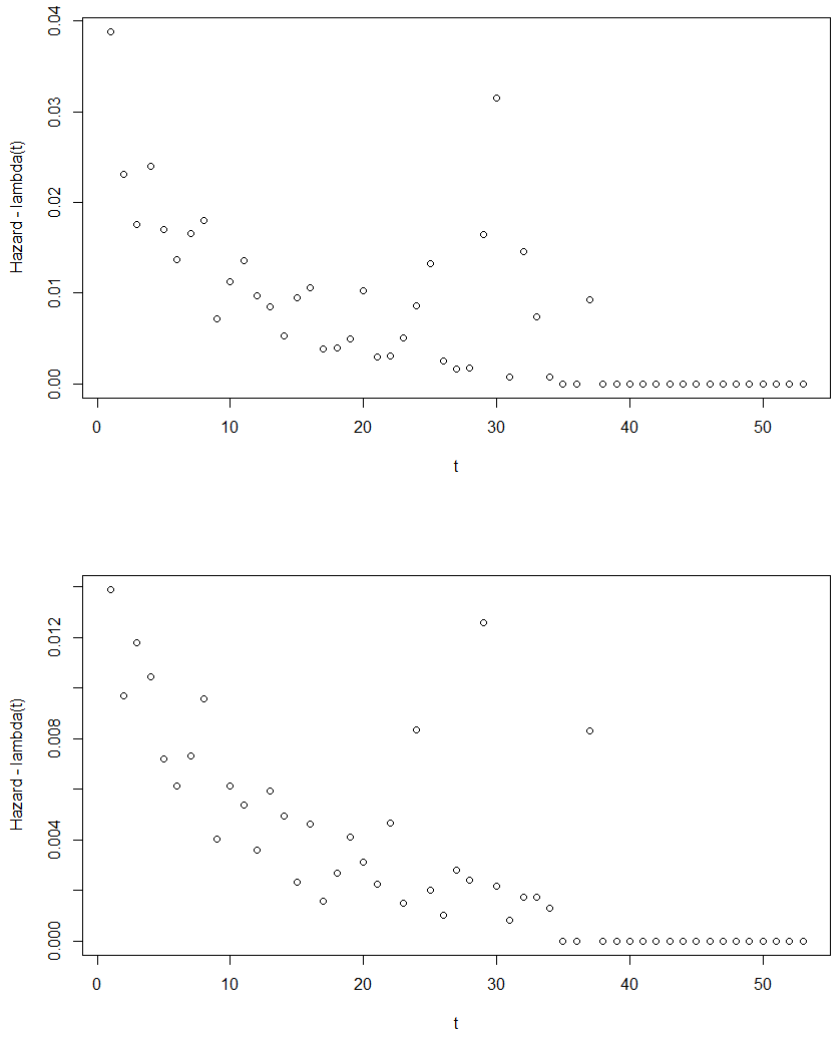

Figure 3.14: $\lambda(t)$ Plots, Censoring at t = 1 (top) and t = 26 (bottom). Please note that the scale of each graph is different.

Like the difference between the cumulative hazard and survival functions, there is a large difference between the instantaneous hazards here. While the t = 26 example is unlikely to ever lead to a reoffence, the t = 1 case may well still lead to one.

Metrics and Evaluation

Predictive Performance As in the "best case" scenario outlined in the previous section, trees are grown using the Maxstat splitrule.

Table 3.8: Random Survival Forest Evaluation Metrics. Parameters: No. Trees = 100, No. Variables per Split = 13, Alpha = 0.1, Minprop = 0.156

Metric	Value
Training Set Error	0.2091
Validation Set (OOB) Error	0.2458
Test Set Error	0.2507

The training and test set errors have, similarly to those produced by the classification and probability forests, decreased slightly relative to those present in the original dataset. As before, this likely reflects the relative improvements in data quality (especially relating to the area variables) between the original and live datasets. From this perspective, it appears that the algorithm is suitable for use on live data and performs relatively well on the live test set as a whole, once again performing better than many comparable survival models.

Variable Importances Now that it has been decided which of the splitrules is most appropriate for this task, permutation importances can be generated for each of the variables to be considered for prediction. For further details as to how these permutation importances are calculated, refer back to Chapter 2 of this thesis.

Table 3.9: Variable Importances. Parameters: No. Trees = 100, No. Variables per Split = 13,

Variables	Permutation Importances
Outcome	**0.0337**
NoPreviousArrests	**0.0326**
PrevFine	**0.0217**
HaversineDist	**0.0190**
OffenceCat	**0.0155**
Fine	0.01277
AgeCommitted	0.0111
MDAClass	0.0082
MultipleOffences	0.0072
Off_EmpBenefits_Perc	0.0069
Off_Sex	0.0069
Off_ASB_per100	0.0042
Off_Income_Perc	0.0044
PrevViolence	0.0041
Crime_Income_Perc	0.0038

Here, the variable importances from the original dataset are somewhat similar to those shown here - once again, Outcome, the NoPreviousArrests, PrevFine and OffenceCat are within the top 5. The Outcome of the offence does, however, appear to be much more important in the live dataset than it was in the original dataset and is now considered to be slightly more important than the number of previous arrests committed by the offender.

Similarly to the Live and Original results from Chapter 2, the Haversine distance becomes more important in the Live dataset compared to the Original, reflecting the improvements in location recording within the database. Other area-related variables, however, are still not considered to be particularly important within the model.

3.5 Conclusions, Limitations and Further Research

3.5.1 Conclusions

In general, the Random Survival Forests algorithm makes reasonably effective predictions as to how long it is likely to take for an offender to reoffend on both the original dataset provided by Dyfed-Powys police and the live data. The concor-

dance scores suggest that these results are either in line or better than many typical survival models fit to typical time to event datasets, meaning that the model is a suitable solution to this problem. Of the Random Survival Forest variants tested on this dataset, the Maxstat split rule (as invented by the creators of the ranger package) is likely to be the best option for calculating predictions on this dataset. Aside from offering large improvements in runtime and training set overfitting over Random Forests generated using the logrank and C split rules, this method also produces comparatively more accurate predictions for first-time offence data. As the generation of accurate predictions for first-time offence data is likely to be important as the results of this project become apparent to the police, given that new offenders are always entering the dataset, it is well worth making the final decision as to which of the many available splitrules is preferable on this basis.

In terms of variable importances, when it comes to predicting offender survival, there are some differences between the variables that the model considers to be important in the classification case and the survival case. While NoPreviousArrests (the estimated number of previous arrests committed by the offender) and PrevFine (the level of previous harm that the offender has inflicted on society) are still the most important variables, alongside AgeCommitted (the age at which the offender committed that particular offence) and Outcome (the treatment of the offence by Dyfed-Powys police), Off_ASB_per100 (the level of anti-social behaviour in the offender's local area) is not as important a factor as it was when the model's aim was simply to classify individuals as likely to or unlikely to reoffend. This implies that the level of anti-social behaviour in the offender's local area affects the offence's overall probability of leading to a reoffence, but not how long it takes for the reoffence to be committed.

3.5.2 Limitations

Once again, a general limitation present in all of the models is the assumption of independence between each of the data records present in the dataset. Again, testing the model's sensitivity to this is difficult, but should be completed if the police believe that this is likely to significantly impact the performance of the model in a live setting.

As before, one unfortunate limitation that this particular dataset possesses is the complete unavailability of consistent offender data prior to the start date, 1st January 2008. As such, it is entirely possible that an offender considered to be a "first-time offender" in this dataset (i.e. an offender not considered to have any previous offences) is not actually a first-time offender at all and has committed a number of offences prior to 2008, which means that a number of the cases that can be considered to be cold-start cases may well not be cold-start at all. In addition, these offenders may well have been present within the dataset for a number of days (or perhaps years) prior to a "first-time" offence. While our dataset isn't technically a left-censored dataset, each new individual on entry to the dataset can be considered to be an individual with no previous criminal record, this assumption may not actually be correct. Nevertheless, given the restriction on the availability of data prior to the start of the study, a criminal history cannot reasonably be assumed for new entries into the dataset.

Once again, within the original dataset, there are many inconsistencies regarding variables relating to the crime's or offender's location. These are mostly in the form of missing postcodes, which mean that an accurate latitude and longitude cannot be processed for some of the crimes within the original dataset as given by Dyfed-Powys police. This impacts on all area-related variables, especially the Haversine distance (as this requires both a relevant crime and offence location to be present in the dataset). As such, the relative importance of these location-related variables may well be understated in the original dataset.

Perhaps the greatest limitation with regards to this problem is that it is unknown (and from the dataset, there is no particular way of knowing) whether or not an offender was removed from observation by Dyfed-Powys police prior to the end date of the observation period. Removal from Dyfed-Powys' dataset could occur for a number of reasons, including (but not limited to) the offender having moved outside of Dyfed-Powys' jurisdiction, or the offender's incarceration. Since Dyfed-Powys police do not possess ongoing address data for each individual offender in their database, it is largely impossible for us to track their location as they move and become subject to different location factors. Moreover, it is also unknown (aside from what seems

to be a few inconsistently reported entries in the unprocessed outcome variable) whether or not an individual was incarcerated following an offence, where that individual was incarcerated, how long they were incarcerated for and whether or not they were subject to any probationary conditions following incarceration. These issues, while also effective in the classification model, become increasingly pressing in the context of a survival model, as incorrect censoring could cause information to be input into the survival model that should not be so input. As such, it is entirely possible that such a model could predict that an incarcerated offender is most likely to reoffend while they are still incarcerated.

3.5.3 Further Research

While the survival and hazard function plots will suffice to make judgements on offenders from a mathematical standpoint, it is possible that these results need to be better translated for use in a policing context. The introduction of risk levels, as they relate to various levels of instantaneous hazard, could help the police to make swifter judgements on offenders without a need to familiarise themselves with the statistical concepts. Without sufficient testing on real-time data, however, it can be quite difficult to ascertain at what hazard values these risk levels should be set and consequently, whether they should be set using instantaneous hazard or based on the eventual survival probability, as predicted by this model. Therefore, for the moment, these risk levels will be left as an open suggestion for a possible future improvement to the model, once sufficient testing has been undertaken on the dataset.

Moreover, extending the work completed in Chapter 2, it is entirely possible that the XGBoost algorithm or a Neural Network could be extended for survival prediction. However, as yet, there is no documented evidence that either of these algorithms are suitable for use in survival context. Therefore, in this case, it has been chosen not to investigate either of these methods for this particular use.

Chapter 4

Recommender Systems for Spatio-Temporal Crime Prediction

4.1 Recommender Systems

Recommender systems, tools designed to support and enhance the natural process of decision making through recommendation, assist individual users in identifying items of interest. These items are often selected from a large set of items that the individual may find it somewhat difficult or overwhelming to choose between [71]. These systems have many applications, mostly in commercial and retail situations, but have yet to be applied the field of predictive policing. The particular type of system to be applied to this problem will be based on Collaborative Filtering, a method first defined in 1992 to describe the e-mail filtering system Tapestry [35]. While the concept of Collaborative Filtering can be defined in many different ways, in this context, Collaborative Filtering will be defined to be a method by which information on the interests of several different *users* can be collaborated together to form automatic predictions about the interests of another *user*. In this case, the *users* in this dataset will be the various locations (in this case, Postcode sectors) within the Dyfed-Powys police dataset. The distributions of crimes (*items*) within different locations (*users*) in Dyfed-Powys, therefore, will be collaborated together to form automatic predictions about the crime distributions of another area. While alternative approaches to building a recommender system do exist, Collaborative Filtering is often considered to be the more popular approach [90] and in this case, is the simplest way to utilise the available data to produce effective recommenda-

tions.

On reviewing the various types of pre-existing Collaborative Filtering recommender systems, it has been decided that a memory-based or user-based system [13, 77, 74] would be most appropriate for our purposes. As *new user* problems within this dataset are unlikely to occur, given that the Dyfed-Powys police force does not significantly alter its coverage area and that there are only a very limited number of times at which each crime can occur, the simpler and more easily explainable memory-based methods are likely to be the most appropriate for this investigation. Since memory-based collaborative filtering systems operate on the assumption that if two *users* A and B *rate* a series of *items* similarly, or have similar behaviours (e.g. buying items on an e-Commerce site, listening to songs on a digital music listening site), they will continue to *rate* those *items* similarly in the future [13]. As the validity of this assumption for the Dyfed-Powys police dataset will require further discussion, this will be further discussed in Section 4.2, following a description of the particular measure of similarity that has been chosen to calculate the similarity between two locations in the Dyfed-Powys area.

Overall, the aim of this recommender system will be to determine, given that a certain crime is occurring or likely to occur in a certain location, whether or not a crime is also likely to occur in another location within Dyfed-Powys within that time window. Under this system, the similarity between two locations (*users*) A and B will be determined by the time windows within which a crime occurs (*items*) and the number of crimes that occur within that crime window (*ratings*). Therefore, under such a system, two locations A and B will be considered to be similar if a large overlap is present in the temporal distribution of their crimes. Once these similarities have been calculated, the matrix containing all pairwise similarities between locations in Dyfed-Powys will be used to assign each of the locations within the dataset to clusters, which will be dependent on the times at which crimes occur within the dataset. These clusters, as well as the similarity matrix used to generate them, can then be used by Dyfed-Powys police to immediately respond to crimes as they occur in real time, as well as plan for the consequences of events that are considered likely to occur within future time windows.

Now that the chosen type and aims of the recommender system have been outlined, it can be discussed how memory-based collaborative filtering systems have previously been applied to police data, as well as how similarly applicable systems have been utilised in other fields.

4.1.1 Related Work

While little or no current research has been undertaken into the use of recommender systems in this context, much can still be learnt from investigating the use of recommender systems in other fields. Although systems designed to provide movie recommendations are perhaps the most well-known in the field [8, 57], recommender systems have been shown to have many other uses and applications, including the personalisation of news homepages [24] and the design of eGovernment systems designed to improve the interaction between governments and citizens [86]. An overview of the many commercial applications and benefits of these systems from such a perspective can be found in Schafer, Konstan and Reidl's review of e-Commerce systems [75]. Similarly, an overview of the many varied design goals of Collaborative Filtering systems, as well as the metrics commonly used for their evaluation can be found in Herlocker, Konstan, Terveen and Reidl's review [40].

In many recommender systems, a variety of measures of similarity are used to calculate the similarity between users across a variety of datasets. Some of the most common measures used here include the Cosine Similarity, Jaccard Similarity, Pearson Correlation Similarity, Euclidean Distance and Manhattan Distance. The measure chosen will depend on both the input data and the design goal of the system. Some systems make use of both item data and the ratings on these items to determine user similarity, while others only use item data. Several memory-based recommender systems also use TF-IDF (Term Frequency-Inverse Document Frequency) as a weighting method to offset the impact of commonly chosen items on the measures of similarity within a matrix. Examples of the use of this weighting include recommender systems designed for citation systems in research papers [4], systems designed to recommend news or other personalised information on social media [78, 69], as well as more general content-based Recommender Systems [63]. Although TF-IDF vectorisation has seen some application in a criminal context from

both a text-mining [30] and crime prediction [91] perspective, its use in a Recommender System outside of a simple text-mining context has not yet been explored in this field.

4.1.2 Discussion and Conclusion

It is evident from previous studies that the measure of similarity between users must be carefully chosen for the recommender system to provide the best prediction of their preferences. Following an investigation of the previous research, the first measure that has been chosen is the Jaccard Similarity. Specifically, as the number of crimes that occur within each time window will be taken into account (and therefore considering the *rating* given to *items* within the dataset), the Jaccard Similarity measure as applied across multisets (also known as the Sorensen-Dice coefficient) will be used. Without producing a weighted combination of measures, it has been shown to be the best similarity measure for the MovieLens dataset in terms of predictive accuracy for a small neighbourhood [16]. Based on this study, since only the pairwise similarity for roughly 200 location (*user*) pairs will be calculated, the Jaccard Similarity will be an appropriate measure for this dataset. The second measure of similarity that will be tested is to be applied following a TF-IDF vectorisation. There are many methods of calculating similarity between location pairs in a dataset, but generally, the most common measure to follow a TF-IDF vectorisation is the cosine similarity [56] and as such, this is the measure that has been chosen. These two methods of calculating measures of similarity will be tested on the same dataset, with their comparative effectiveness in this case being evaluated by a metric designed to evaluate the distribution of their clusters.

From these previous studies, much of the research into recommender systems is conducted from a commercial perspective, with the end goal being the development of a user-friendly, effective system that fulfils the needs of the consumer. While some systems are designed to produce accurate predictions of what items the user may enjoy, others are simply designed to provide an enjoyable browsing experience, with others still being designed to influence the future interests of the user. The appropriate method of evaluating of these systems, therefore, must depend entirely on the design goal of the system. Unlike most studies into Recommender systems,

in which an evaluation in terms of their predictive accuracy [34] is required, the effectiveness of this system will not be judged on its predictive accuracy. The reasons for this choice are two-fold. Firstly, since a group of “”correct clusters”” cannot be given (this being an unsupervised learning problem), there is no absolute standard to measure our predictions against. Secondly, due to the fact that there is no restriction on how many crimes happen in each location, it is uncertain how the predictive accuracy of the system's output should be measured. Predicting that 25 crimes will occur in a location at a given time when 20 crimes actually occurred, for example, is likely to be a far less costly mistake from a policing perspective than predicting that 5 crimes will occur when none occurred. With these practical uncertainties in place, it is not entirely appropriate to judge the system's performance on the basis of various predictive accuracy metrics. As such, an alternative way of evaluating the efficacy of the system must be found. This method of evaluation, the silhouette score, will be detailed further in Section 3.1.

4.2 Measure of Similarity

The first step in designing a user-based Collaborative Filtering system is to select an appropriate measure of similarity. Before an appropriate measure of similarity can be chosen, however, it must be defined what exactly is meant by a "similar distribution of crimes" in each location. In this case, a "similar distribution of crimes" will be defined to have occurred between two locations A and B if the times at which these offences occur are similar. Although it was briefly considered that the type of offence (Burglary, Violent Offence etc.) should be included in the measure of similarity calculations, doing so introduced several issues, mostly due to the high level of correlation between the various offence categories and the time at which they were committed. As such, it was decided that it would likely be sufficient to simply take the time at which the offence occurred into account.

In order to extract the maximum amount of information from the similarity matrix and produce the most accurate measure of similarity between these locations, however, it must be certain that crimes are sorted into the correct number of time intervals. If too great a number of time intervals are chosen, the similarities will all be too close to zero and the chosen measure will therefore become somewhat

meaningless. Taking too few time intervals into consideration, however, could lead significant patterns in the distribution of these crimes to be omitted. Upon testing various similarity matrices, investigating the patterns of several different time variable combinations and consulting Dyfed-Powys police on the appropriateness of these intervals, it was determined that the minimum number of time variables needed to properly capture the similarities between each location was two. These variables were the day of the week (Monday-Sunday) and the time of day at which the crime occurred. Since little to no evidence could be found of seasonal or monthly trends in the occurrence of crime at a either a general or more category specific level, except in the case of one particular type of theft, it is unlikely that including a seasonal variable in our similarity measure calculations would aid us in capturing the trends within the data.

While the days of the week easily translate into a small number of categories, the time of day as given in the dataset was not so simple to separate into a number of appropriate intervals. Using hourly intervals was found to be too granular for this dataset, so the time of day was broken down into four intervals, giving a total of 28 possible time intervals for a crime to fall into. The final partitions used to categorise the time of day at which a crime was committed are described below:

Table 4.1: Time Intervals as Used to Categorise Crime Occurrence

5:00am - 11:59am	Morning
12:00pm - 4:59pm	Afternoon
5:00pm - 8:59pm	Evening
9:00pm - 4:59am	Night

Following a discussion with Dyfed-Powys police, it was also decided that any crime occurring before 5:00am on any one day would be treated as if it occurred on the previous day. Treating overnight crime in this way puts the categorisation of crime in line with actual criminal movements and allows the day on which overnight crime occurs to be more easily determined. This is particularly useful for offences that often tend to happen at indeterminate times in the early hours of the morning; for example, an offence said to have been committed between Saturday at 11:00pm and Sunday at 1:00am would, under this adjustment criterion, be considered to have occurred on the Saturday between 11:00pm and 1:00am. Now, instead of only being

certain of the fact that the crime occurred at night,it is also possible to be certain of the day it occurred on. Since there are many crimes like this one within the dataset, constructing the time periods in this way assists greatly in the removal of unnecessary uncertainty.

4.2.1 Jaccard Similarity

Jaccard Similarity of Sets

For two sets of items A and B, which in this case will be a list of the crimes occurring in two locations A and B, the Jaccard Similarity of these two sets is defined as the size of the intersection of the two sets (i.e. number of unique crime times that appear in both A and B) divided by the size of their union (i.e. number of unique crime times that appear in either A or B). The formula for the Jaccard Similarity of sets is as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4.1}$$

Defining the similarity between two locations to simply be the Jaccard Similarity for sets between these two locations does, however, have a large disadvantage in this context. This is due to the fact that the Jaccard Similarity of sets does not take into account the number of crimes in each category that occur at each location, only whether or not the crime of that specific type/time combination occurs within that location. While in many situations this is not an issue, in this situation not taking duplicate elements into consideration could mean that some vital information about the distribution of crimes is lost. An observation of 30 offences in one area on a Saturday night is, after all, very different to an observation of 5 offences in another area, given that the same total number of crimes occur in both areas. Therefore, rather than thinking of each location as a set of crimes with each element in that set being a unique crime-time combination that occurs in that area, it is preferable to think of each location as a multiset of crimes in which each individual crime of each combination is counted separately.

As such, in this case, the Jaccard Similarity across multisets is likely a far more appropriate measure than the Jaccard Similarity of sets. If we consider each set A

and B to instead be a multiset, the definition of the measure of similarity remains the same.

In the context of this thesis, each multiset A and B represents a location within the Dyfed-Powys area and the items within this set represent instances of criminal activity within the area.

It is worth noting that unlike the distance matrix (i.e. $1 - J(A, B)$), due to its violation of the triangle inequality, the Jaccard Similarity as applied to multisets is not a metric. However, since this similarity measure is symmetric and necessarily non-negative, it does satisfy two out of four conditions, meaning that it can be considered to be a semi-metric measure of similarity. This measure of similarity is bounded between 0 and 0.5, with 0 representing the lowest level of similarity between two bags and 0.5 the highest. For the purposes of the clustering techniques employed later on, it is sufficient that these three conditions are satisfied.

Before clustering algorithms are applied to this similarity matrix, it is helpful to visually analyse the similarity structures present within the dataset. With these visualisations in place, the results and implications of these similarities, including those pertaining to the appropriateness of Collaborative Filtering methods on this dataset, can be presented and discussed. In Section 4.4 below, heatmap visualisations of similarity matrices produced from our dataset will be analysed.

The Location Similarity Matrix

To visualise these matrices, it must be decided which years and months of data will be taken into account in the similarity calculations. At first, to avoid any possible issues with the aggregation of data across subsequent years, it was decided to visualise a similarity matrix for one particular year. Here, it has been chosen to use only crime data from 2010, with the locations i, j within the Dyfed Powys police area being divided by UK postcode sector. A heatmap visualisation of the similarity matrix s_{ij} for this dataset is provided in Figure 4.1 below. Darker areas indicate greater similarity between the pair of locations i, j, while brighter areas indicate less similarity between the two locations.

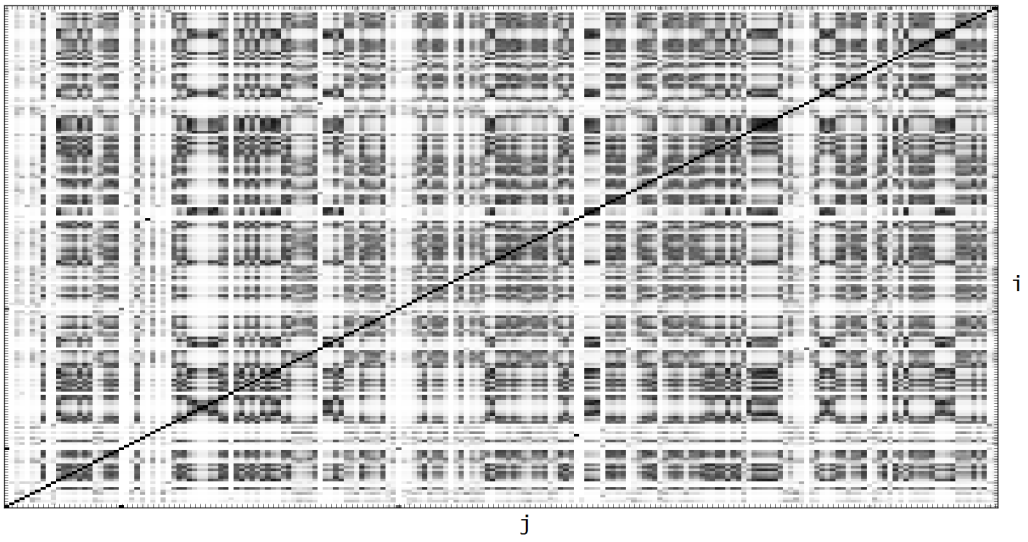

Figure 4.1: Similarity Matrix, 2010, Postcode Sectors

There is a clear pattern of similarities within the matrix, meaning that there is certainly a case that clustering algorithms should be looked into as a method of appropriately clustering these locations. Whether this pattern holds over time, however, must be established in order to determine whether or not Collaborative Filtering techniques are appropriate in this case. To illustrate the validity of the assumptions that firstly, individuals who held similar opinions in the past will continue to do so in the future and secondly, that individuals will like similar kinds of items in the future as they did in the past, further similarity matrix visualisations have been produced for various years and months within the dataset. These are shown in Figures 4.2 and 4.4.

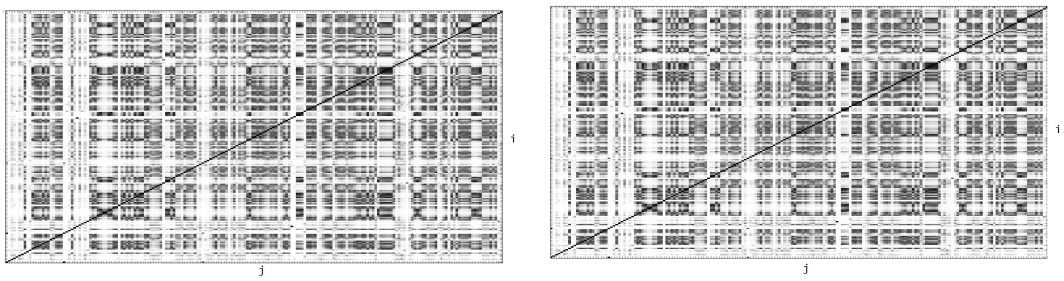

Figure 4.2: Similarity Matrices, 2012 and 2014, Postcode Sectors

The pairwise similarities between the locations in this dataset do not appear to change from year to year. In order to see whether there are any real significant differences between 2012 and 2014, a matrix of the differences between the similarities calculated for 2012 and 2014 datasets is shown below.

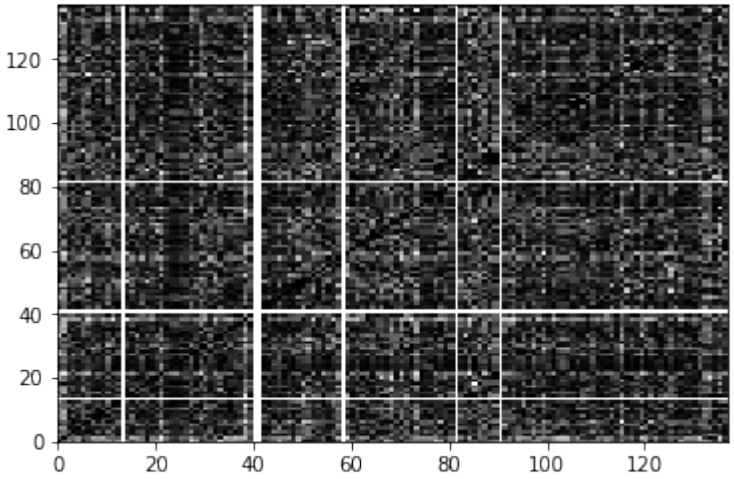

Figure 4.3: Differences Between Similarity Matrices, 2012 and 2014, Postcode Sectors

The maximum difference in similarities here is 0.20125 and the minimum is 0 - the majority of similarities are much closer to minimum of 0 than the maximum of 0.5. The median difference is 0.02824 and the interquartile range is (0.01295, 0.05064), meaning that the difference between similarities is low in most cases. As such, it would be reasonable to say that the differences between similarities are constant over time. Monthly similarity matrices for this dataset are included below.

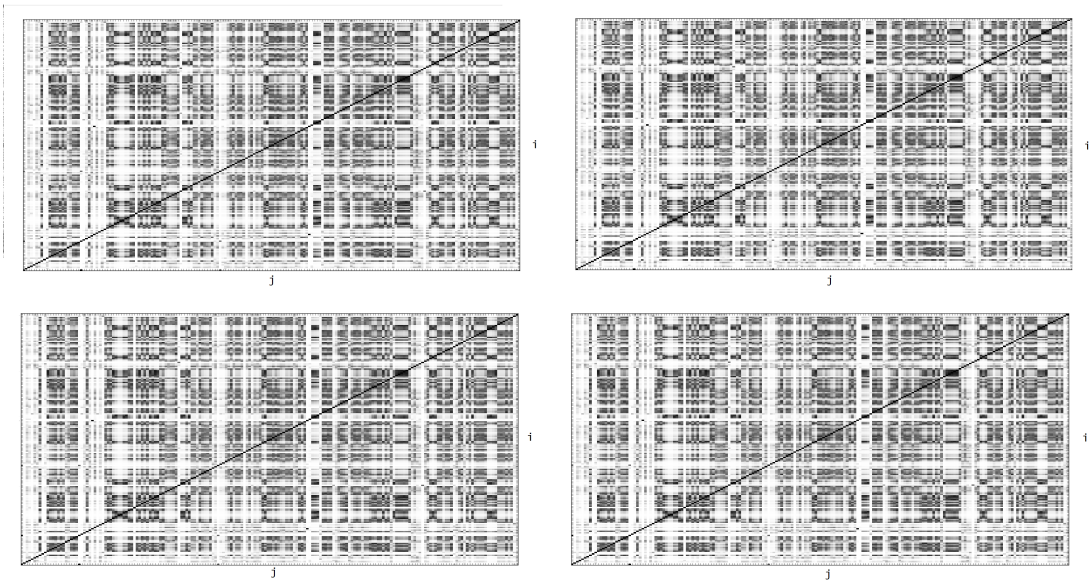

Figure 4.4: Similarity Matrices, February, May, August and October 2010, Postcode Sectors

They also do not appear to alter from month to month, indicating that there is no significant seasonal or yearly trend in the pattern of similarities. As such, the assumption that a "customer's preferences" (a location's crime patterns) do not change over time appears to hold. In order to check whether or not this is the case, the monthly differences in similarity must be examined.

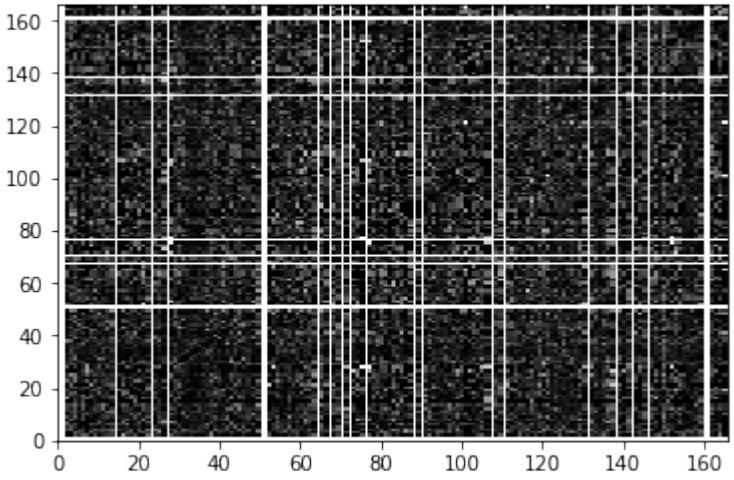

Figure 4.5: Differences Between Similarity Matrices, April 2010 and September 2010, Postcode Sectors

The maximum difference in similarities here is 0.5 and the minimum is 0 - the majority of similarities are much closer to minimum of 0 than the maximum of 0.5. The median difference is 0.04808 and the interquartile range is (0.01242, 0.09091),

meaning that the difference between similarities is low in most cases. As such, it would be reasonable to say that the differences between similarities are constant over time.

Now that the Jaccard Similarity of Bags has been defined and its appropriateness as a measure of similarity has been discussed, the second method by which such a matrix can be constructed must be defined and discussed.

4.2.2 TF-IDF and Cosine Similarity

TF-IDF

TF-IDF vectorisation, a technique often implemented in a text mining or document engineering context, transforms a table of term counts into a vector showing the relative importance of each term as it appears within a series of documents. While our use of TF-IDF will have nothing to do with terms or documents as such, as this research falls squarely outside of either a text mining or document engineering context, it is certainly possible to apply the technique to this dataset.

In order to understand why it is appropriate to use TF-IDF vectorisation in this context, each location within the dataset must first be thought of as a separate document within the overall corpus that is the Dyfed-Powys area. In each document, there are several terms, or crimes, that can be categorised by the time at which they were reported to the police. Therefore, the term counts in this instance are the raw crime counts in each of the locations, separated by the day of week on which and the interval of time in which they occur. These term counts for each weekday and time interval will, however, not be recorded for each of these variables separately. The choice to register the terms defining the crimes within the count matrix as a combination of the day of week and the time of day at which it was reported instead of each of these variables separately has been made due to significant evidence that the distribution of crimes within this dataset varies more markedly when both of these variables are taken in tandem, rather than when each of these variables are considered separately. Therefore, if a crime occurs on a Monday night, it will be registered within the 'document' (location) as the 'term' MondayNight.

From this definition, a count matrix can be produced detailing the number of each of these crimes in each location. An example of this is shown in Table 4.2.

Table 4.2: Crime Counts per Location, grouped by Time Interval

Time	A	B	C
FridayAfternoon	22	20	5
FridayEvening	10	7	14
FridayNight	15	12	25
SaturdayMorning	32	14	1
SaturdayAfternoon	20	28	2
SaturdayEvening	14	10	2
SaturdayNight	15	16	1
SundayMorning	15	10	3
SundayAfternoon	20	15	1
SundayEvening	29	13	1

As previously described, TF-IDF vectorisation can be applied to generate a weight for each of the terms (crime times) within this count matrix. This weighting is the product of two separate statistics, term frequency and inverse document frequency, and is defined as follows:

$$TF - IDF(t, d) = TF(t, d) \cdot IDF(t) \tag{4.2}$$

Firstly, the term frequency represents how frequently a term (crime time) appears in a single document (location) within the corpus (Dyfed-Powys area). The more frequently a crime occurs at a certain time within a given location, the higher the term frequency will be. The inverse document frequency, on the other hand, represents how commonly a crime happens at a given time across all locations. The more frequently crimes occur at that time within the Dyfed-Powys, the lower the inverse document frequency will be. As such, a high TF-IDF weighting corresponds to a high level of crime occurring at a certain time within a given location, combined with a low number of occurrences across Dyfed-Powys as a whole.

To efficiently transform our precomputed crime count matrix to a TF-IDF matrix, it has been chosen to use scikit-learn's *TfidfTransformer* function. Under the function's default settings, the term frequency is defined as follows:

$$TF(t, d) = f_{t,d} \tag{4.3}$$

and the inverse document frequency is defined as follows:

$$IDF(t) = \log(\frac{1 + n_d}{1 + df(d,t)} + 1) \tag{4.4}$$

Therefore, the TF-IDF statistic is defined as follows:

$$TF - IDF(t,d) = f_{t,d} \cdot \log(\frac{1 + n_d}{1 + df(d,t)} + 1) \tag{4.5}$$

Although this method of calculating the inverse document frequency differs slightly from the methodology found in most of the literature, it has been decided to use the defaults as given by the *TfidfTransformer* function. This is partly due to the fact that the one added to the numerator and denominator of the log term in the IDF fraction, representing a document in which each term (in this case, each type of crime) appears exactly once, prevents errors caused by zero divisions.

Cosine Similarity

With the transformation of the count matrix to a TF-IDF vectorisation, the TF-IDF weights can be combined with a measure of similarity between each pair of locations in the dataset. As previously stated, the measure of similarity that has been chosen is the cosine similarity, a measure that defines similarity in terms of the angle between two non-zero vectors. This measure is defined for two vectors **A** and **B** in the equation below:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}||||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_j}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_j^2}} \tag{4.6}$$

In this context, A and B are the vectors of TF-IDF weights given for any two separate locations within the dataset. A_i and B_j are, therefore, the components of these vectors A and B respectively. Due to the fact that TF-IDF weights cannot be negative, the cosine similarity between two locations will be bounded between 0 and 1. As such, it is possible to state that here, the angle between two vectors cannot be greater than 90°. These similarities, calculated for each pair of locations i, j within the Dyfed-Powys police area, are then calculated to form a similarity matrix s_{ij}.

As before, once these similarity matrices have been produced, they can be visualised below, firstly to investigate the patterns within the dataset and secondly to

illustrate the suitability of Collaborative Filtering methods in this context.

The Location Similarity Matrix

Once again, in order to illustrate the validity of the assumptions that firstly, individuals who held similar opinions in the past will continue to do so in the future and secondly, that individuals will 'like' similar kinds of items in the future as they did in the past (i.e. that crimes will be committed at similar times), visualisations can once again be produced of the cosine similarity matrices for various intervals of time.

In this instance, this analysis will begin by looking at monthly similarity patterns. If there is a difference between the patterns in each of the matrices, then our assumptions must be considered. Heatmap visualisations of these similarity matrices i, j, separated by month, are shown below. Brighter areas indicate a greater level of similarity between the pair of locations i, j, while darker areas indicate less similarity between the two locations. Once again, it has been chosen to separate and define the locations i, j within Dyfed-Powys by postcode sector.

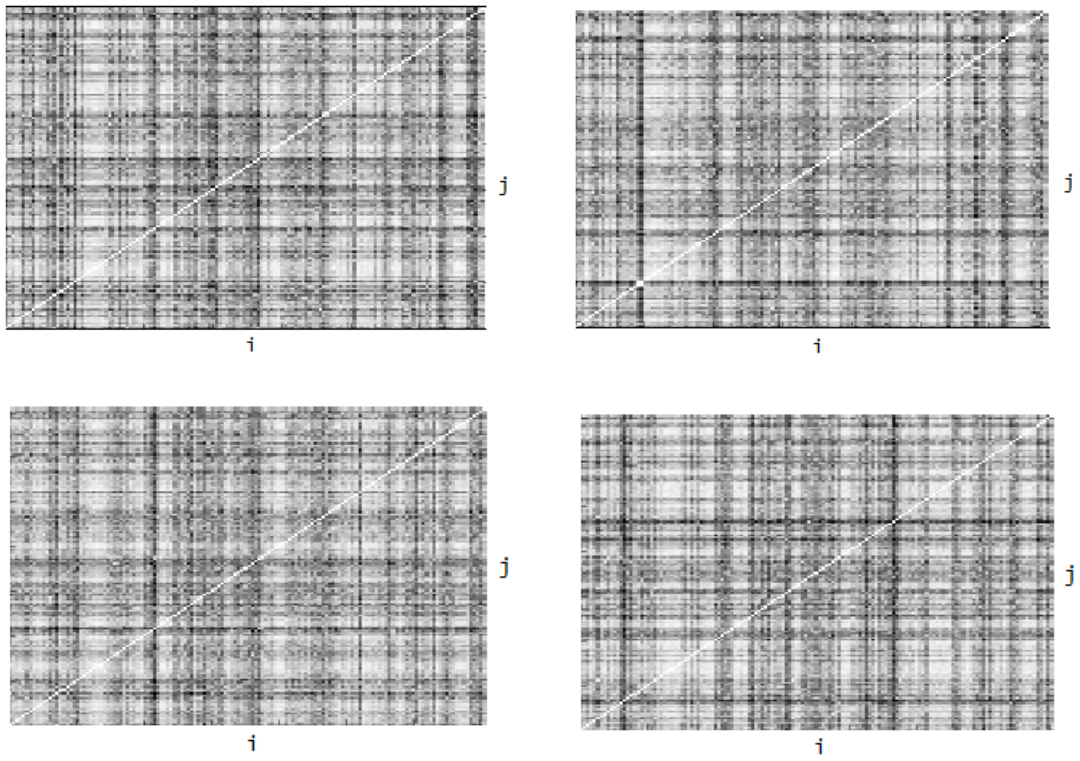

Figure 4.6: Similarity Matrices, January, March, June and September, Postcode Sectors

Here, the patterns of similarity between different locations do vary significantly from month to month. As such, given the current conditions, it is not possible to say that a constant pattern of similarities between locations holds over time. However, if grouping each crime by the month in which it is reported as well as the day of week and time of year, it is possible that this assumption may still hold for year-on-year crime.

Taking the month of year into account in the count matrix as well as the week-day and time of day, another similarity matrix s_{ij} can be produced. A heatmap visualisation of this similarity matrix for a single year is provided in Figure 4.7. This particular similarity matrix is for crime data from 2010 and as before, the locations i, j within the Dyfed Powys police area are divided by postcode sector.

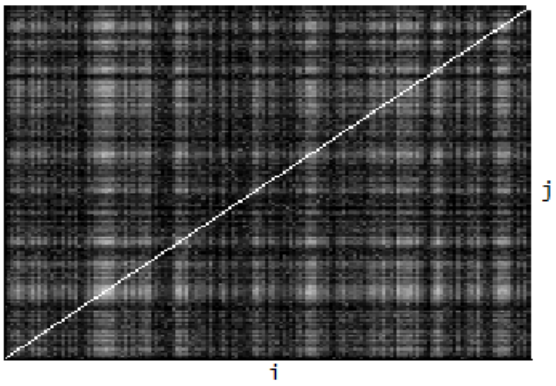

Figure 4.7: Similarity Matrix, 2010, Postcode Sectors

Once again, it can be seen that there is a clear similarity pattern within the matrix, meaning that there is certainly a case that clustering algorithms should be looked into. Whether this pattern holds over time, however, must be established in order to determine whether or not Collaborative Filtering techniques are appropriate. These matrices are shown in Figure 4.8.

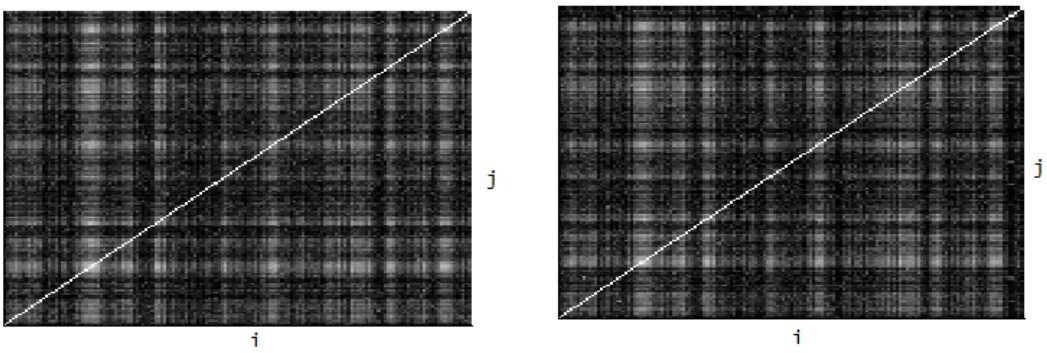

Figure 4.8: Similarity Matrices, 2012 and 2014, Postcode Sectors

While it appears that there are very few significant differences between the 2012 and 2014 similarity matrices, a heatmap of the differences between them must be produced in order to see whether this is the case. This is shown below in Figure 4.9.

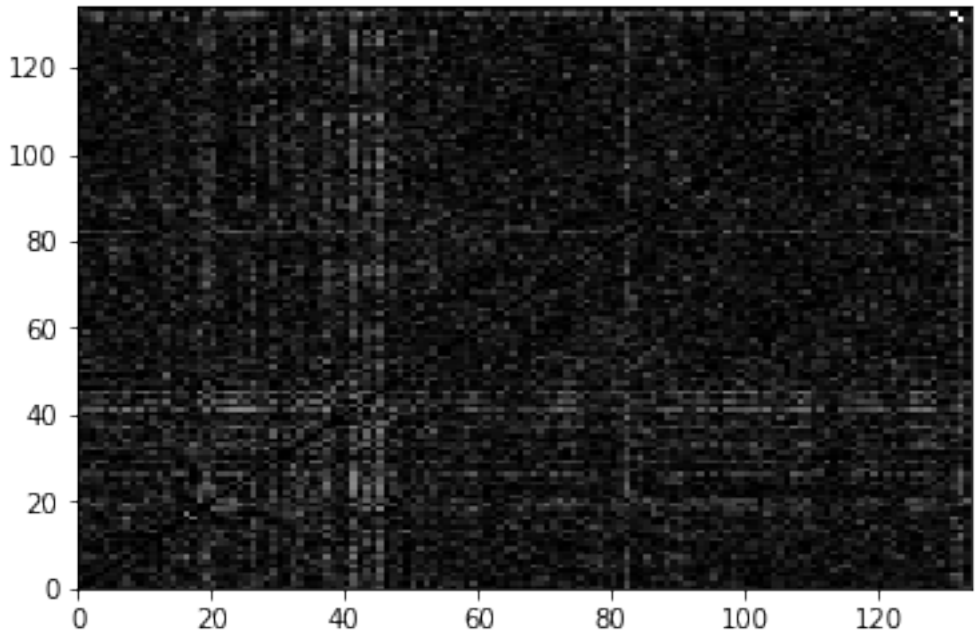

Figure 4.9: Similarity Matrix Differences, 2012/2014, Postcode Sectors

The maximum difference in similarities here is 0.89584 and the minimum is 0 - the majority of similarity differences are much closer to minimum of 0 than the maximum of 1. The median difference is 0.05685 and the interquartile range is (0.02668, 0.09932), meaning that the difference between similarities is low in most cases. As such, it would be reasonable to say that the differences between similarities are constant over time.

Here, taking the month in which a crime was committed into account, the assumption that a "customer's preferences" (i.e. the patterns of crime within a location) do not change over time appears to hold. Collaborative Filtering, therefore, can certainly be seen to be an appropriate method by which a Recommender System can be put in place for this similarity matrix.

4.3 Clustering and Visualisation Algorithms

In order to better understand the structure of the similarity matrix, ways in which the pairwise similarities generated by the method can be clustered will now be investigated. As previously mentioned, this will be accomplished by generating a series of clusters from the precomputed similarity matrices from the previous section. In this section, the details of the two different algorithms to be used for this purpose will be outlined, alongside the method by which both of these methods will be visualised and evaluated. Each of these algorithms has its origins in a separate school of thought; Spectral Clustering [89] is based on the centroid-based K-Means method and Affinity Propagation [29] on message passing algorithms. The sci-kit learn [65] implementation of both of these algorithms has been used.

4.3.1 Spectral Clustering

Spectral clustering is a popular clustering technique that often outperforms many other traditional algorithms, such as the K-Means clustering that it is based on. This technique uses the spectrum of the similarity matrix, or the set of its eigenvalues, to reduce the dimensions of the matrix before clustering the data in fewer dimensions. The goal of this algorithm is to reduce the space in such a way that locations that are *close* (i.e. locations for which the similarity between them is close to 1) are always within the same cluster and locations that are *far away* (i.e. locations for which the similarity between them is close to 0) are within different clusters. For a 2-cluster problem, the algorithm as implemented by scikit-learn will solve the normalised cut problem as first described in Shi and Malik [80] and for a multi-cluster problem, it will solve a K-way normalised cut problem as described in Yu and Shi [94]. Given that the similarity matrix to be input into this algorithm is precomputed, the general process of defining these clusters will be as follows [89]:

1. Let W be the nxn weighted similarity matrix as defined in Section 4.2.

2. From the graph produced by this similarity matrix (in which the nodes are locations and the edges the similarity between them), compute the unnormalised Laplacian matrix L.

An unnormalised Laplacian matrix L is term describing a matrix representation of a graph. This is a symmetric M-matrix that is also diagonally dominant, for which every row and column sum within the matrix is 0.

3. Then, compute the first k generalised eigenvectors $u_1, ..., u_k$ of the generalised eigenproblem $L = \lambda D$.

4. Let $U \in \mathbf{R}^{n+k}$ be the matrix containing vectors $u_1, ..., u_k$ as columns.

5. For $i = 1, ..., n$ let $y_i \in \mathbf{R}^k$ be the vector corresponding to the i^{th} row of U.

6. Cluster the points $y_i \in \mathbf{R}^k$ with the K-Means algorithm into clusters $C_1, ..., C_k$.

7. Output the clusters $A_1, ..., A_k$.

As this algorithm in the form given by scikit-learn utilises the K-means clustering algorithm as a base and this algorithm is sensitive to initialisation, it is necessary to run the clustering step several times with different initialisations. In addition, the number of clusters must be specified. Since it is unknown whether the desired or "correct" number of clusters are generated, it is unknown what number K should be set to. In our case, we have chosen to alter this parameter in order to maximise the silhouette score of the clusters - this measure will be further discussed in Section 4.3.3.

4.3.2 Affinity Propagation

Affinity Propagation [29] is a clustering algorithm based on the concept of message passing between data points. Unlike traditional centroid-based algorithms, affinity propagation does not require an estimate of the number of clusters - the estimation of the correct number of clusters is provided by the algorithm. The centres of the clusters also do not correspond to points in space between the clusters, as the locations themselves will serve as the cluster centres. By taking K locations within the dataset and taking them to be exemplars for a cluster (i.e. the best representative of the crime trends within that cluster), this algorithm forms a graph-based set of clusters using an algorithm that unlike K-Means clustering, does not require many initialisations.

The clusters created by the affinity propagation algorithm are defined in the following way. First, let $x_1, ..., x_n$ be our set of locations and s the similarity matrix as calculated in the previous section, which satisfies the condition $s(x_i, x_j) > s(x_i, x_k)$ if x_j is considered to be more similar to x_i than x_k. From the similarity matrix chosen, two new matrices are then created and zero-initialised. These matrices are:

1. The responsibility matrix, whose values $r(i, k)$ are numerical representations of how well-suited a location x_k is to serve as an exemplar for a cluster containing x_i, relative to all other locations within the dataset.

2. The availability matrix, whose values $a(i, k)$ represent the "appropriateness" of location x_i in picking x_k as its exemplar, given the preferences of all other locations within the dataset to pick x_k as their exemplar.

These matrices are then iteratively updated by the algorithm. First, the responsibility matrix is updated:

$$r\left(i,k\right) \leftarrow s\left(i,k\right) - \max_{K \neq k'} \left\{ a(i,k^{'}) + s(i,k^{'}) \right\} \tag{4.7}$$

Next, the availability matrix is updated:

$$a\left(i,k\right) \leftarrow \min\left(0, r\left(k,k\right)\right) + \sum_{i' \notin \{i,k\}} \max\left(0, r\left(i^{'}, k\right)\right) \text{ for } i \neq k \tag{4.8}$$

and

$$a\left(k,k\right) \leftarrow \sum_{i' \neq k} \max\left(0, r\left(i^{'}, k\right)\right) \text{ for } i = k \tag{4.9}$$

This method either stops at a predetermined number of iterations, or continues until the clusters remain unchanged over a number of iterations. If the responsibility for a location x_i, $r(i,i) > 0$, the location x_i will serve as an exemplar, or the centre of a particular cluster.

Two parameters in the affinity propagation algorithm affect the number of clusters produced. The first of these parameters, the damping factor, only indirectly affects the number of clusters produced by the model; the purpose of the damping parameter is to *damp* (reduce the effect of) the oscillations that can be caused by overshooting in the iterations that update the responsibility and availability matrices. The second parameter, the preference, instead alters the number of clusters directly, with higher preference values leading to the algorithm choosing a greater number of exemplars and therefore leading to the selection of a greater number of clusters.

In order to evaluate the effectiveness of these algorithms and discover the appropriate number of clusters, an appropriate empirical measure must be found that can quantify how "well grouped" the clusters are. Since it is unknown and not possible to otherwise estimate the number of location clusters that are present within the data, nor can it be estimated which locations belong to which clusters, an external evaluation of the quality of clustering produced by these three algorithms is not possible. Therefore, in order to gain an understanding of the performance of these algorithms, a way to evaluate the internal structure of the clusters will need to be found.

4.3.3 Cluster Evaluation

To solve the issue of selecting an appropriate number of clusters and evaluating the algorithm in one step, the choice was made to optimise the number of clusters with

regards to the silhouette [72] produced by the clusters in each case. The silhouette of a series of clusters is evaluated in two separate ways; firstly, considering the degree of separation between each of the clusters and secondly, the degree of separation within each of the clusters, must be evaluated. To define exactly how silhouette evaluation defines the separation between each cluster, as well as the degree of separation within clusters, the method by which an individual location within the dataset is defined to be similar to the other points within the dataset in the context of this method must be outlined.

Firstly, for any location i in the dataset, denote the cluster it has been assigned to as A. When A contains any locations other than i (i.e. the size of the cluster $>$ 1), the average dissimilarity (i.e. 0.5 - the Jaccard Similarity of Bags or 1 - the Cosine Similarity) of location i to all other locations in A is then denoted as $a(i)$. Now consider any other cluster C generated by the algorithm. For these other clusters, $d(i, C)$ is defined to be the average dissimilarity of location i to all other locations in C.

After computing $d(i, C)$ for all $C \neq A$, the minimum $d(i, C)$ is selected and denoted $b(i) = min(d(i, C)), C \neq A$. The cluster B for which this minimum is attained is on average, the closest cluster of all available clusters to A. The silhouette coefficient, $s(i)$, for each location i is therefore defined as:

$$s\left(i\right) = \frac{b\left(i\right) - a\left(i\right)}{\max\left\{a\left(i\right), b\left(i\right)\right\}} \tag{4.10}$$

,

or

$$s\left(i\right) = \begin{cases} 1 - a\left(i\right)/b\left(i\right), \ if \ a\left(i\right) < b\left(i\right) \\ 0, \ if \ a\left(i\right) = b\left(i\right) \\ b\left(i\right)/a\left(i\right) - 1, \ if \ a\left(i\right) > b\left(i\right) \end{cases}. \tag{4.11}$$

For each location i, a silhouette coefficient $-1 \leq s(i) \leq 1$ is then obtained. A coefficient close to -1 will indicate that this location is likely to have been incorrectly placed within A and would be better placed in its closest neighbouring cluster B, while a coefficient close to 1 will indicate that this location is likely to have been

correctly placed in A. A coefficient of close to 0, on the other hand, indicates that it is uncertain whether or not location i belongs in its assigned cluster A or the neighbouring cluster B. These coefficients are then taken for all locations within the dataset and the mean of these is taken to produce the silhouette score, our chosen accuracy metric for this dataset.

By maximising this silhouette score, the appropriate number of clusters (in the case of Spectral Clustering) or the appropriate parameter values (in the case of Affinity Propagation) will need to be selected in order to produce the most well-separated clusters for each algorithm.

Now that the methods by which these clustering algorithms generate clusters can be discussed and the method by which these clusters can be evaluated, clusters from the similarity matrix can be generated and computed in the previous section.

4.3.4 Cluster Visualisation

To produce a visualisation showing how these clusters are situated, a way of reducing their dimensions must be found so that they can be visualised in 2D space. To visualise these clusters, a plot of the locations has been constructed using Locally Linear Embedding to reduce the 190 x 190 similarity matrix to 2 dimensions. These locations (set as the nodes of the graph) are joined by the similarities (edges), with thicker, darker lines representing greater similarity and lighter, thinner lines representing lesser similarity. The plots shown in the results sections are produced in the same way for Spectral Clustering and Affinity Propagation. In each case, centroids are not shown but cluster assignments are.

4.4 Results

As discussed in the previous section, the next step in these calculations is to feed the similarity matrix into each of the clustering algorithms. Since both of the algorithms are based on a similarity matrix and not a distance matrix, there is no need to transform the similarity matrix in any way - it can simply be put straight into the algorithm by setting the affinity parameter equal to 'precomputed'. While

the Spectral Clustering algorithm does not require any additional parameters to be set other than the number of clusters k (which was varied in order to find the optimum configuration of clusters), the Affinity Propagation algorithm requires one extra parameter, the damping factor, to be set by the user. This damping factor decreases, or "damps" the effects of oscillations to a degree specified by the damping parameter. As previously recommended by the authors of the original Affinity Propagation paper [29], the damping factor (*damping* in scikit-learn) was set to 0.9 for all preference values.

In this section, the results of and relative effectiveness of the clusters produced by these two algorithms will be discussed and presented, beginning with Spectral Clustering. In each case, both types of similarity matrix will be fed into the algorithm. The differing results of these clustering algorithms will then be shown and discussed, both in a visual and a quantitative format, in order to produce a good picture of the relative advantages and disadvantages of each combination.

4.4.1 Spectral Clustering

Jaccard Similarity of Bags

Firstly, the similarity matrix $s_{i,j}$ containing the Jaccard Similarity of bags values for each pair of locations x_i, x_j was taken, as calculated for an aggregation of crimes across all available years within the dataset. Using this as our precomputed matrix, the Spectral Clustering algorithm was then run a number of times for different values of k. The results of these calculations for selected values of k are detailed below:

Table 4.3: Silhouette Score Results vs. Number of Clusters, Spectral Clustering

No. Clusters (k)	Silhouette Score
2	0.2447
3	0.3193
4	0.2534
5	0.2490
6	0.2538
7	0.1856
8	0.1810

From these silhouette scores, the optimum number of clusters, k, for the Spectral Clustering method is 3. Although none of these silhouette scores suggest that this

clustering method has managed to uncover a particularly strong set of well-separated clusters, the 3-cluster solution is still by far the best by this evaluation metric. To get a better idea of exactly where these clusters best approximate the clustering pattern inherent within the similarity matrix, a visualisation of these clusters can be found below in Figure 4.10.

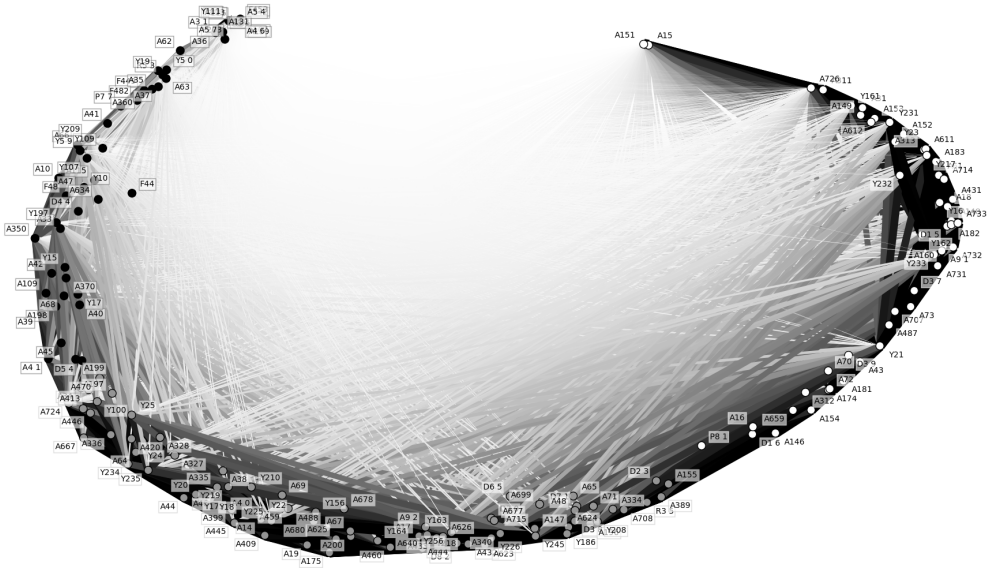

Figure 4.10: Spectral Clustering Similarity Plot, $K = 3$

The locations within this dataset have been assigned to three somewhat evenly-sized clusters, with one cluster occupying the sparse left-hand side of the graph and another occupying the dense right-hand side, while a third is located in the centre. Both the silhouette scores and the plot indicate that there is a reasonable degree of overlap between the clusters designated by the Spectral Clustering algorithm.

To further examine the nature of the silhouettes of the three clusters in each algorithm's case, the individual silhouette scores of each cluster (and by extension, each point) can be looked at in order to identify where the "problem points" may be. For the $K = 3$ case, the individual silhouette score of each of the clusters in the visualisation above can be examined.

Table 4.4: Silhouette Scores of Individual Clusters, Spectral Clustering, $K = 3$

Cluster Colour	Silhouette Score
White	0.4976
Grey	0.4310
Black	0.0006

As expected, misclassification issues appear to occur most strongly in the third (least dense) cluster, in which the intra-cluster similarities are generally lower and many small clusters of two to three individual locations can be found. This cluster does, however, contain many locations that are bordering on but not officially within the area covered by Dyfed-Powys police. These locations are likely included in the database due to the fact that the offender who committed them is from the Dyfed-Powys area, or the particular officer involved in the arrest was a part of Dyfed-Powys police. There are also many areas in which less than 20 crimes occurred over the entire observation period; since there is evidently no significant crime in these locations, it is likely not worth including them in the dataset.

To properly examine whether the inclusion of these locations is adversely affecting the clustering, these locations must be removed from the dataset and the clustering algorithm must be re-run. The silhouette scores when these adjustments are made are as follows:

Table 4.5: Inclusion Adjusted Silhouette Scores of Individual Clusters, Spectral Clustering, $K = 3$

Cluster Colour	Silhouette Score
White	0.5647
Grey	0.3901
Black	0.1131

The mean Silhouette Score across the three clusters is 0.3411433, which is a small improvement on the previous score. Although the silhouette score of the grey cluster has decreased slightly, the silhouette score of the black cluster has improved markedly, strengthening the case for not including these exceptionally low crime or out of area locations in the overall clustering. When these areas are removed, however, the optimum number of clusters as defined by the silhouette score is actually 2 and not 3; the silhouette score in the 2-cluster case is actually 0.3826099, which indicates that a more defined set of clusters has been generated. The 2-cluster case is visualised in Figure 4.11 below.

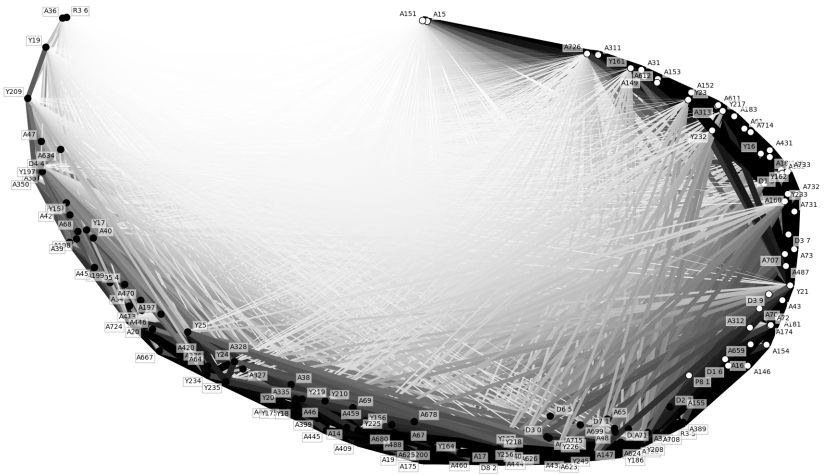

Figure 4.11: Spectral Clustering Similarity Plot, $K = 2$

From this plot, the left-hand end of the graph (which contained many locations for which the similarity between that location and all others in the dataset was very low) has been heavily affected by the removal of these possibly problematic locations.

In conclusion, when considering this particular method and similarity matrix, the best set of clusters can be found when $K = 2$ and crimes that occur in locations outside Dyfed-Powys are excluded, as well as any locations in which only a very low number of crimes occur.

TF-IDF and Cosine Similarity

Changing the precomputed similarity matrix $s_{i,j}$ to the TF-IDF and cosine similarity matrix for all locations x_i, x_j within the dataset, the Spectral Clustering algorithm can once again be investigated for a range of k. The results for a selected range of k are detailed below in Table 4.6.

Table 4.6: Silhouette Score Results vs. Number of Clusters, Spectral Clustering

No. Clusters (k)	Silhouette Score
2	0.0197
3	0.0152
4	0.0062
5	-0.0190
6	-0.0088
7	-0.0549
8	-0.0291

While none of the values of K here have produced a silhouette score that would

indicate a series of well-separated clusters, the best value of K according to this metric appears to be 2. The cluster structure for this best case scenario is visualised in Figure 4.12 below.

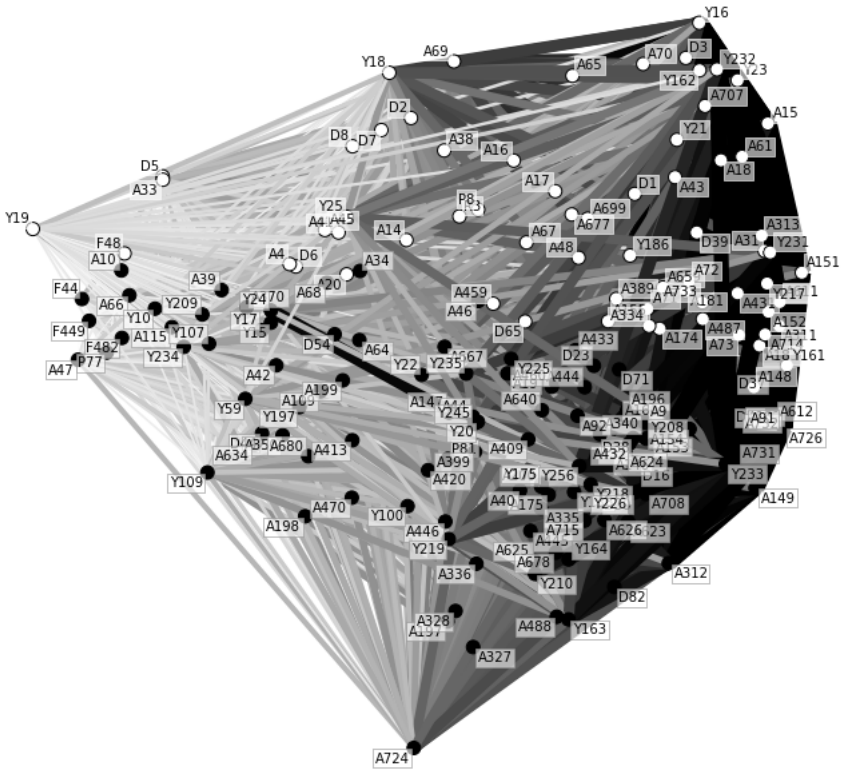

Figure 4.12: Spectral Clustering Visualisation, $K = 2$

Roughly two thirds of the locations are grouped in one cluster, while one third of the locations are placed in a second cluster. This density pattern within the clustering, where large numbers of locations are considered to be very similar to one another and a small number of locations are considered to be less similar, does not appear to be exploited by the clustering. As such, it is not surprising that the silhouette score for this algorithm indicates poor clustering performance.

In order to see whether these observations are correct, their individual silhouette scores can be calculated. Calculating these scores can aid in the identification and subsequent evaluation of issues within a more quantitative way. These scores for each of the individual clusters are given below in Table 4.7.

Table 4.7: Silhouette Scores of Individual Clusters, Spectral Clustering, $K = 2$

Cluster Colour	Silhouette Score
Black	0.1969
White	-0.0891

The observations made previously were correct; the white cluster actually has a negative silhouette score, indicating that a large number of locations within that cluster would actually be better placed in the black cluster. Although this algorithm is clearly not suitable for clustering this similarity matrix, the visualisation has uncovered an interesting similarity structure. The visualisation that a subset of locations on the right-hand side appear to be similar to a large number of other locations in the dataset, while locations on the left-hand side of the visualisation appear to be dissimilar to a large number of other locations in the dataset. This structure, while reasonably evident prior to TF-IDF vectorisation, becomes more obvious after the introduction of this weighting method.

4.4.2 Affinity Propagation

To further explore methods of clustering the complex similarity structure evident in the matrix, both of the precomputed matrices were fed into the Affinity Propagation algorithm, starting with the Jaccard Similarity matrix.

Jaccard Similarity of Bags

Instead of directly altering the number of clusters, k, with each iteration, Affinity Propagation alters the number of and distribution of its clusters by adjusting the preference value. As previously stated, in general, increasing this value increases the number of clusters and vice versa. The silhouette scores, as calculated for a selection of preference values, are detailed below in Table 4.8.

Table 4.8: Silhouette Score Results vs. Preference Value, Affinity Propagation

Preference	No. Clusters (k)	Silhouette Score
-15	2	0.2674
-5	3	0.2419
-2.5	4	0.2461
-2	3	0.2375
-1.5	6	0.1354
-1	5	0.1815
-0.75	11	0.1414
-0.5	11	0.1666
-0.25	15	0.1493
0	16	0.1528
0.25	23	0.1385
0.5	40	0.1114
0.75	96	0.0725

The solution to the optimum number and configuration of clusters for Affinity Propagation is less clear than the optimum number in Spectral Clustering. Although the silhouette score suggests that the "tightest" clusters are produced when $K = 2$ and the preference is set to -15, the silhouette scores produced by a preference of -5, -2.5 and -2 are very close to the maximum score. Therefore, in order to decide which of these clusterings can be deemed to be the most accurate, a visualisation of the clusters must be produced. These plots are shown in Figure 4.13.

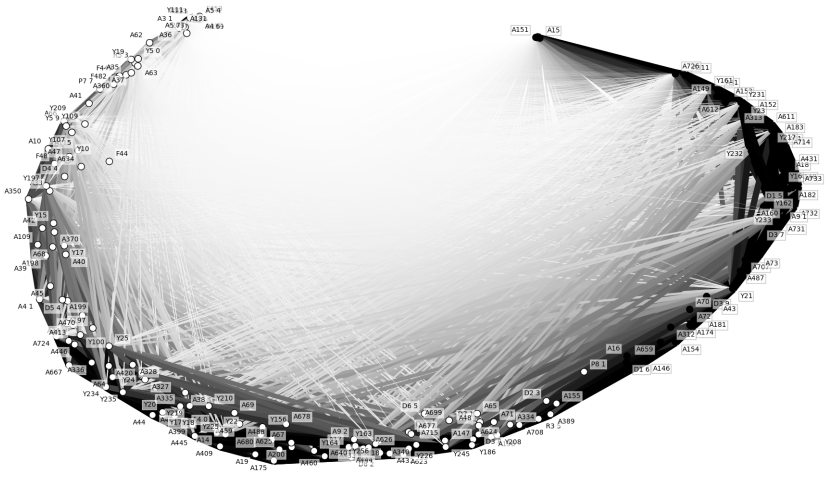

Figure 4.13: Affinity Propagation Similarity Plot, Preference = -15, K = 2

The two clusters are uneven in size, with the first cluster being much smaller in size than the second. From the many thick, dark edges present on the right hand side of the graph, it is evident that all of the locations in the first cluster are very similar to one another. While the pattern of these points having much greater similarity

to one another than other areas has been established, setting the preference value this low has caused the algorithm to miss out on some visually obvious clusters, including one that exists at the centre of the graph. Less obviously, the algorithm also misses out on a few very obvious smaller clusters at the far left of the graph.

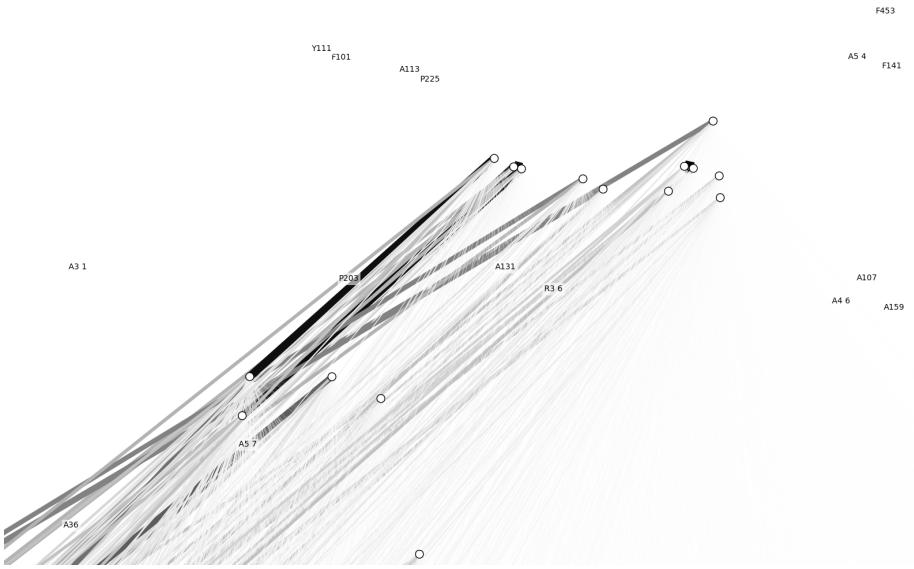

Figure 4.14: Affinity Propagation Similarity Plot, Preference = -15, K = 2

On looking more closely at this section of the graph, a small but significant cluster pattern has not been detected by the algorithm. As such, it is prudent to investigate the possibility that despite a slightly lower silhouette score, a larger number of clusters may better fit the similarity patterns within the matrix.

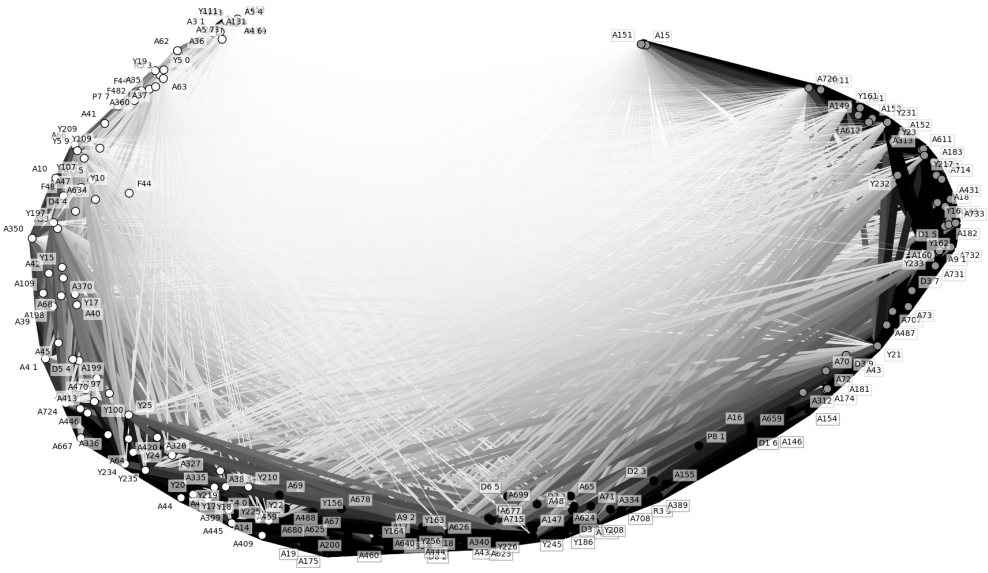

Figure 4.15: Affinity Propagation Similarity Plot, Preference = -5, K = 3

These three clusters are uneven in size. The third cluster, represented by white circles on the graph, is significantly larger in size than either of the others. However, once again, the small clusters at the edge of the graph have not been detected by the algorithm.

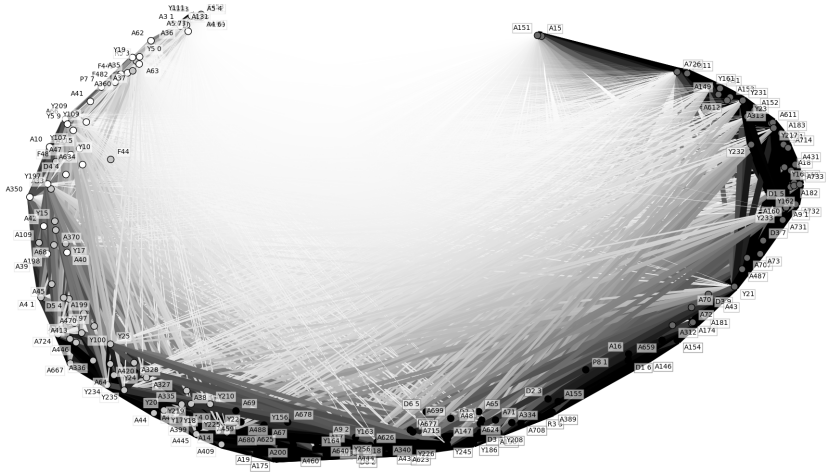

Figure 4.16: Affinity Propagation Similarity Plot, Preference = -3.5, K = 4

The four-cluster option provides more evenly-sized clusters, but some anomalous results have crept in; the white and light grey cluster points on the left-hand side of the curve are mixed in together in a random fashion, possibly indicating that fewer clusters may be required.

Returning to the small clusters found at the far left of the graph, it is evident that these small clusters may be causing some disruption to the silhouettes of the clusters generated by the algorithm. Since it is evident from the silhouette scores for the Spectral Clustering method that this had a small but positive effect on the overall clustering performance, the effect of the removal of these locations on the Affinity Propagation method should also be tested. The silhouette scores after this adjustment has been made are shown below in Table 4.9.

Table 4.9: Inclusion Adjusted Silhouette Scores, Affinity Propagation

Preference	No. Clusters (k)	Silhouette Score
-10	2	0.3842
-5	3	0.3255
-1	2	0.3183
-0.5	5	0.1958
0	8	0.1845
0.5	18	0.1285

Again, the optimum number of clusters appears to be 2, with this particular configuration of clusters being generated by a preference of -10. This configuration of clusters is visualised in Figure 4.18.

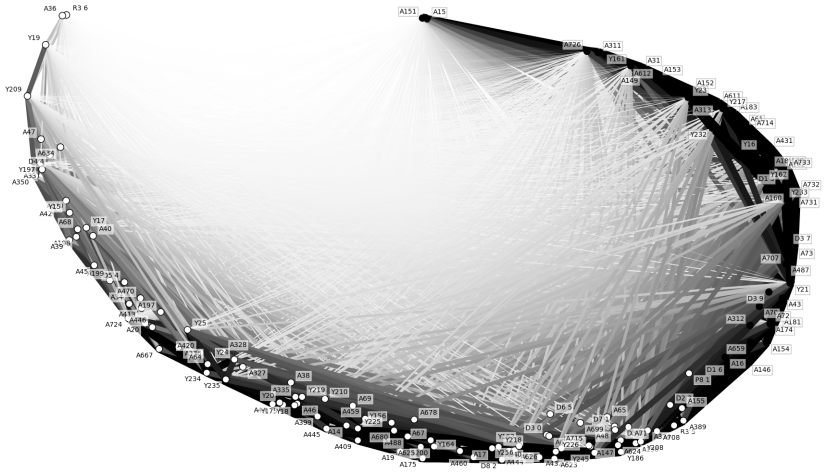

Figure 4.17: Spectral Clustering Similarity Plot, $K = 2$

Here, the locations are grouped into 2 unevenly-sized clusters, with the white cluster located on the less dense left-hand side of the graph being much larger than the black cluster. From the visualisation, the locations in the black cluster are all very similar to one another, whereas the locations in the white cluster are much more loosely related.

TF-IDF and Cosine Similarity

Changing the precomputed similarity matrix $s_{i,j}$ to the TF-IDF and cosine similarity matrix for all locations x_i, x_j within the dataset, the Affinity Propagation algorithm can once again be investigated for a range of preference values. The results for a selected range of preference values are detailed below in Table 4.10.

Table 4.10: Silhouette Score Results vs. Preference Value, Affinity Propagation

Preference	No. Clusters (k)	Silhouette Score
0.25	3	0.3432
0.275	8	0.0583
0.3	11	0.0576
0.325	15	0.0583
0.35	16	0.0573
0.375	19	0.0531
0.4	21	0.2487
0.425	26	0.2304
0.45	33	0.0509

Compared to the clusters generated by Spectral Clustering, the clusters generated by Affinity Propagation are much better. Here, the best set of clusters corresponds to a preference value of 0.25, which corresponds to a layout containing 3 clusters. This best case scenario is visualised below.

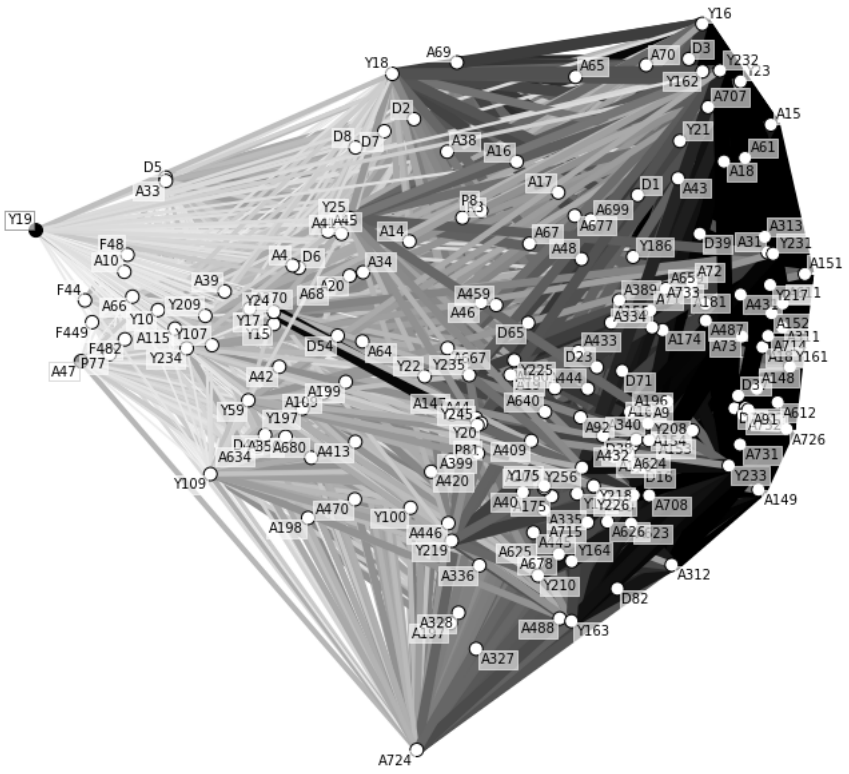

Figure 4.18: Affinity Propagation Visualisation, Preference = 0.25

The clusters generated by Affinity Propagation differ quite significantly from those generated by Spectral Clustering. From this plot, it is evident that nearly all of the points are considered to be within the same cluster. In fact, there are only two locations (Y19 and A47) that, according to this method, are considered to be dissimilar from the other locations in the Dyfed-Powys area. Although this algorithm offers a higher level of performance than Spectral Clustering, placing all of the results in one cluster leaves little of interest from a policing standpoint. Unfortunately, all other options with a reasonable silhouette score are somewhat similar in their approach to clustering - one very large cluster is formed, with a few locations being "dumped out" of the cluster. This can be seen in the silhouette scores of the three clusters, which are shown below in Table 4.11.

Table 4.11: Silhouette Scores of Individual Clusters, Affinity Propagation, Preference = 0.25

Cluster Colour	Silhouette Score
White	0.3472
Grey	0.0000
Black	0.0000

From the clustering results produced by the Affinity Propagation algorithm, there is a reasonable case that almost all of the locations within the dataset actually belong within one, loosely connected cluster.

4.4.3 Further Investigation

From these results, it can be concluded that the optimal set of clusters were produced by the Affinity Propagation algorithm, when the measure of similarity between the locations was set to be the Jaccard Similarity. The maximum silhouette score generated by any of these methods was approximately 0.38, showing that either a weak structure with a great deal of overlap is present, or that the algorithm has identified an artificial structure within the data.

From the plots and scores generated by each algorithm, it is likely that the reason these algorithms have not arrived at a good clustering structure is that there is some sort of unidentified parameter that controls the overall degree of separation of each cluster from the next. A potential explanation for this pattern will be examined, before moving on to see whether it holds in the Live Test data.

Urban-Rural Classification

From the visualisations produced by local linear embedding, significant evidence is observed that some sort of parameter or combination of parameters affects the degree of similarity between different locations, as some locations have a great deal of similarity with many other locations, while other locations are only similar to very few other locations within the Dyfed-Powys area. From observations of these plots and the distribution of the locations within them, it is likely that this parameter may well be related to the demographics of the area in question. More specifically, it seems that the clustering of these locations may be affected by the extent of the

urban development within that postcode sector. Many of the areas on the dense side of the graph are town centres or tourist areas, where many of the locations on the sparse side correspond to rural areas containing many small, interconnected villages. As such, it must be investigated as to whether or not the degree of urban development in a particular location corresponds to the temporal distribution of crime within that location.

To investigate whether these observations are correct, clusters generated by the Urban-Rural classification index will be used as a comparison, as previously defined in Chapter 2. The simplest way to see where these classification are present in the similarity matrix is to use these classifications to cluster the locations, replacing the cluster labels generated by the best case clustering scenario with labels according to the Rural/Urban classification. A visualisation of this best case scenario (Jaccard Similarity, location adjusted), where different coloured circles represent different clusters, are shown below.

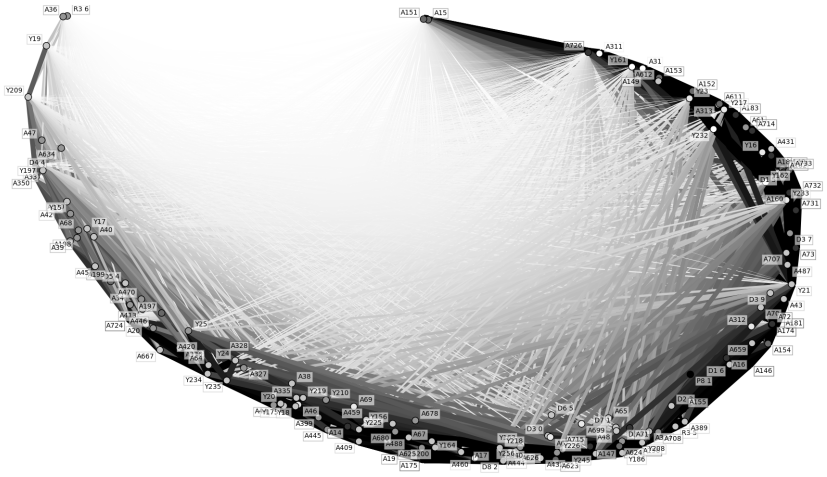

Figure 4.19: Similarity Plot of Locations, Clustered by Rural/Urban Classification

From this plot (note that the black/white/grey fill of the circle does not indicate how rural/urban a location is, as this is not an ordinal measure), it is quite evident that the solution to our clustering problem cannot be found in the rural/urban classification of the locations. In fact, clustering these points by their rural/urban classification instead of using Affinity Propagation actually produces a Silhouette Score of -0.16588, which indicates that clustering the points using this information will mean that many of the locations will end up in the incorrect clusters.

To investigate whether or not the sheer number of clusters or the extra consideration of local sparsity within the dataset may simply be the problem, it is prudent to check what will happen if the sparsity of the surrounding locations is ignored. In this case, it has been decided to reduce the number of clusters from 8 to 4 and re-plot the graph with this in mind.

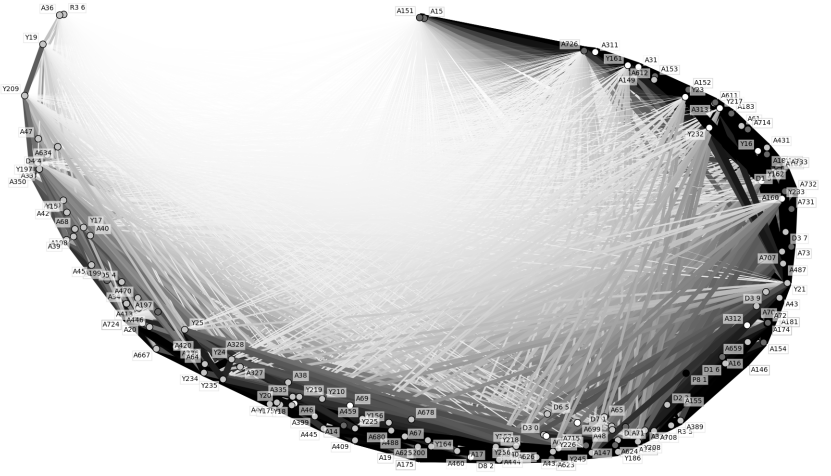

Figure 4.20: Similarity Plot of Locations, Clustered by Modified Rural/Urban Classification

Reducing the number of clusters makes little difference in the effectiveness of the clustering, except possibly on the left-hand side of the graph, which does appear to be somewhat more uniform than it was previously. The Silhouette Score here is -0.1073, indicating that many of the locations are still likely to be clustered wrongly, especially on the right-hand side of the graph.

While the Rural/Urban classifications provide a solid basis for either clustering or discovering the hidden parameter(s) that control the shape of the similarity graph, there may well be some sort of Rural/Urban divide - however, short of coming up with our own form of Rural/Urban classification method to suit the crimes within this particular dataset, it cannot be said for certain whether this is the case.

Live Test Data

Due to additional area data being present in the dataset, the similarity pattern that has been previously observed may not hold in more recent data. This could be due to biases in the missingness of the location values - it's entirely possible that locations are not missing at random and there is either some sort of systematic data

error present in the data, or a systematic bias in the way that information is input by the police.

Referring to the Jaccard Similarity matrix (colours inverted between the two matrices) below, some significant differences can be observed between the 2014 and 2017 data.

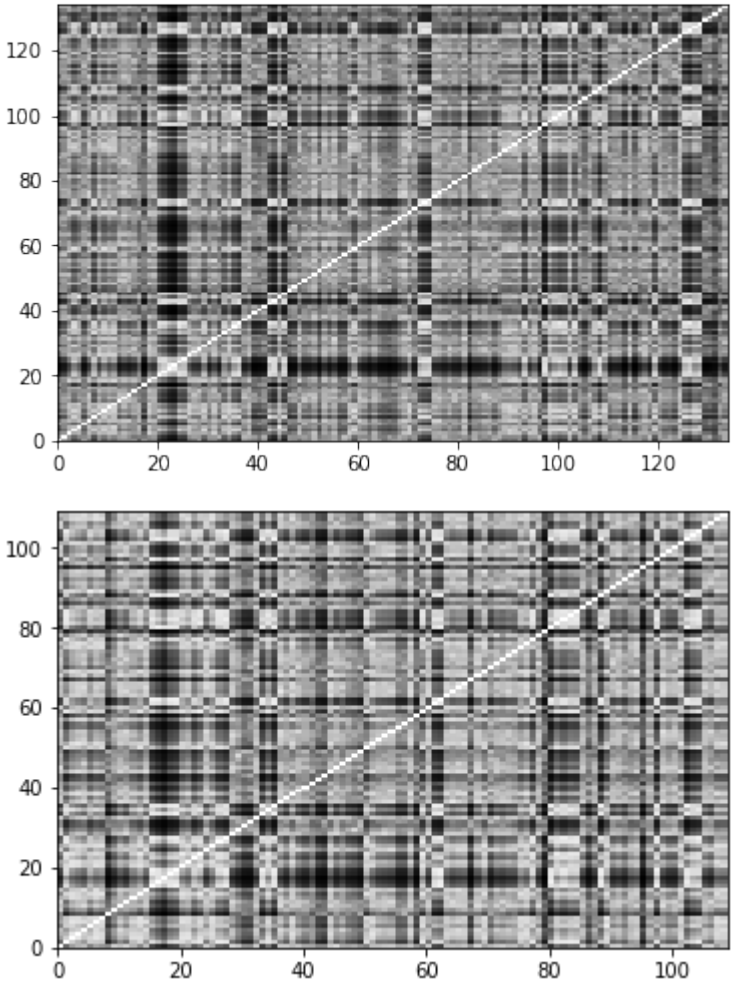

Figure 4.21: Jaccard Similarity maps, 2014 vs 2017 ("Live Test") data

This is likely to mean that the location data is not missing at random, and that the similarity matrix will have to be re-computed and re-checked for use in a live data context, as well as the clustering produced by this algorithm. This is worth bearing in mind when developing final tools for police use - due to significant changes in the data quality between 2014 and 2017, the expectations placed on this method will need to be re-assessed in order to make use of it in a live data context.

4.5 Conclusion, Limitations and Further Research

The similarity matrix produced by the chosen measure of similarity, the Jaccard Similarity of Bags, was found to be almost constant over time. Therefore, it can be concluded that if a location A is similar to a location B, it will likely always be similar to that location. From the point of view of research into Collaborative Filtering, this opens a door into further investigating the effectiveness of these types of Recommender Systems in the field of predictive policing. From a policing perspective, this is an important assertion to be able to make; knowing that if a change occurs in the crime distribution of location A, it will likely also occur in location B, will open new avenues for the police in the face of changes in criminal activity. Rather than simply reacting to the change by altering their activities in the location in question, they will be able to use the similarities between the locations in the dataset to see where similar changes in activity might also occur, altering their activities in those similar locations in preparation of the same changes in criminality in those locations. While the original aim of the research was not to produce a predictive tool, it is possible that this measure of similarity can be used to predict changes in crime across various locations, given that it is already known that a change has occurred in a particular location A.

Moving on to the clustering algorithms, it can be concluded that Spectral Clustering and Affinity Propagation, though different in their methodology, arrived at the same conclusion in this context. Although a clear and unchanging pattern is visible in the location similarity matrices, it does not appear that either of the clustering methods can properly describe or capture the nature of this pattern. In fact, even the clusters generated by each method are incredibly similar; aside from a few borderline points, almost all of the locations have been assigned to the same clusters in each case. Due to the nature of these similarities, it is likely that all (or almost all) of the locations actually belong to the same cluster, as evidenced in the results generated by the Affinity Propagation algorithm when TF-IDF and cosine similarity were chosen as the measures of similarity.

However, all is not lost. While such a low silhouette score can often indicate that the clusters are somewhat randomly allocated and are not at all well-separated, our

investigations into using the rural/urban index of an area to cluster the data shows that the clusters generated by both algorithms do provide at least some idea of how the data is structured compared to simple guesswork. The discovery of a structure in which the crime similarity between what appears to be town centres is very high and the similarity between more rural areas is much lower is also helpful; while it does not provide any immediate solutions, it does highlight a possible area in which investigations can be made in the future.

4.5.1 Limitations

In the dataset provided by Dyfed-Powys police, a number of the offences were given with inaccurate, poorly spelt or very loosely defined locations. For example, in many of the postcode columns, postcodes such as "SA15" and "SA73" were given, meaning that the offence could have occurred in any sector within the SA15 and SA73 areas. While this is not such an issue for urban areas, where multiple postcode sectors frequently exist within a small geographical area, this can be a problem in rural areas where a single postcode area (for example, LD5) could cover an impractically large geographical area. While these offences were included to the best of our ability, the limitations of this missing information were actually quite significant when considering that loosely defined or misspelled locations made up a significant proportion of the approximately 80000 records provided by the force. As such, although there were certainly enough records within the dataset for the patterns in our matrices to be reasonable, the large amount of missing location data may mean that these trends are not necessarily present in reality.

On top of this, as a result of inaccurate or missing offence location data, the clusters generated by these algorithms may not be quite as representative of the real offence patterns as they should be and in fact, the real location clusters may well look somewhat different to those generated here. For example, should many of the offences without an accurate postcode occur at a certain time or be of a type that tends to be distributed at a certain time, it is entirely possible that the absence of this data will have skewed the count distributions of crime over time significantly. While this level of disruption to the generated clusters is not likely, the unknown reason behind these missing locations could be causing significant issues for this method of

behaviour prediction. In fact, this has been noted to be the case when comparing our 2014 data vs. live data from 2017.

4.5.2 Further Research

While for our purposes, the degree of overlap between clusters generated for this dataset may not be too much of an issue, it is still possible that further research could be done. From our investigations, it has been discovered that a measure of similarity that is constant over time within our original dataset. Therefore, as long as this can be considered to be the case within the live data, the primary purpose of our investigation, which was to create a reactive tool to be used by Dyfed-Powys police, can be fulfilled by the generation of a simple k-nearest neighbours list of most similar locations. The secondary purpose of our investigation, however, may merit further study.

Chapter 5

Further Research Questions

This section outlines some further predictive policing techniques that were initially investigated but for various data or computational reasons, were deemed unsuitable for use in a policing context at this current time. For each of the two techniques investigated, this chapter will comprise the following:

1. A short review of the reasoning behind this technique's application and any prior work conducted within this area.

2. The theory behind the technique and a description of the expected inputs and outputs.

3. The results collected, limitations and why these are significant enough to warrant not making use of the technique in the current policing context.

4. A description of an ideal situation in which this technique may be usefully applied and what Dyfed-Powys police can do in order to potentially make use of this technique at a future date.

5.1 Recurrent Neural Networks for Spatio-Temporal Crime Prediction

Neural Networks, as previously investigated for the purposes of reoffender prediction in Chapter 2 of this thesis, may well have another application in this field, namely

in the area of spatio-temporal prediction. While Feedforward Neural Networks have previously been used in order to build models of reoffender behaviour, very few investigations have been made into the suitability of Neural Networks for further use in the field of predictive policing.

One study, conducted on data from the city of Pittsburgh (PA), showed that Neural Networks performed at least as well as an ordinary least squares regression model when tasked with producing an early warning system for 911 drug calls [59]. Similarly, LSTM neural networks have also been used in order to profile calls made by users in a mobile telecommunication network, with the aim of discriminating suspicious or fraudulent call patterns [2] in order to identify individuals who have been obtaining telephone services either free of charge or at a reduced rate.

In this section, the use of Neural Networks will be investigated to provide a solution to one specific problem: taking a subset of crimes at time t in location L within the Dyfed-Powys area, the question as to where and when the next crime in L is likely to occur will be addressed. One of the many questions that can be addressed in this way, given that a sequence of m crimes have already occurred in L prior to time t, is at what time $T > t$ a subsequent crime $m + 1$ is likely to occur (assuming that there is, in fact, a subsequent crime within L). Here, the aim is to produce a prediction of crime $m + 1$'s time of occurrence T given a sequence of m crimes at each time t.

To model these sequences, a particular type of neural network known as a Recurrent Neural Network (RNN) can be constructed. Unlike Feedforward neural networks, which produce static models, RNNs produce dynamic models, i.e. models that change over time as new data is input. They do this by saving the hidden state that determined the previous classification in a series. In each new step, this hidden state is then combined with new input data to produce a new hidden state and classification. The hidden states of previous steps in the RNN are therefore recycled to produce a modified successor. This means that for datasets like this, where the next element in a series may depend heavily on several previous elements in the series, RNNs can be used to predict the value of the element at the next time step.

Lipton, Berkowitz and Elkan's empirically-focused review of RNNs [51], including an overview of their many types and uses in sequential datasets, provides a good examination of the benefits and challenges of the use of these networks in a sequence prediction context. Within their review, a particular type of Recurrent Neural Network known as a Long Short-Term Memory Neural Network [41] is discussed at length. A second empirical review, conducted on several sequential datasets by Jozefowicz, Zaremba and Sutskever [45] following the development of a new architecture known as a Gated Recurrent Unit (GRU) [20], suggests that although both LSTM and GRU networks perform well, a specific architecture based on GRU exists that outperforms both the LSTM and GRU architectures and offers the best performance of the networks tested. In this thesis, the use of both of these structures will be investigated to solve the problem at hand.

5.1.1 Algorithms for Prediction

In this situation, the input of each RNN will be a sequence $x^{(1)}, x^{(2)}, ..., x^{(T)}$, where each $x^{(t)}, t \geq 0 \in \mathbb{Z}$ is a real-valued vector. Similarly, the target sequence of a RNN is a sequence $y^{(1)}, y^{(2)}, ..., y^{(T)}$, where each $y^{(t)}$ is a real-valued vector. The output, or predicted sequence of this RNN is therefore denoted by $y^{(\hat{1})}, y^{(\hat{2})}, ..., y^{(\hat{T})}$, where each $y^{(\hat{t})}$ is a real-valued vector.

At a given time t, any nodes $h^{(t)}$ with recurrent edges receive input from both the current input data point $x^{(t)}$ and also from the values of nodes $h^{(t-1)}$ in the previous timestep's hidden layer. The output $\hat{y^{(t)}}$ at each time t is then calculated given the hidden node values $h^{(t)}$ at time t. An input $x^{(t-1)}$ at time $t-1$ can therefore influence the output $\hat{y^{(t)}}$ at time t, $t+1$, $t+2$, etc. by way of these recurrent connections.

At a time t, the value of the hidden state $h^{(t)}$ is calculated from the input $x^{(t)}$ at time t and the "memory" $h^{(t-1)}$ carried over from the previous time step using the following equation:

$$h^{(t)} = f(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h) \tag{5.1}$$

where W^{hx} is the matrix of conventional weights between the input (x) and the hidden (h) layer and W^{hh} is the matrix of recurrent weights between the hidden

layer and itself at adjacent time steps. The vector b_h is a bias parameter, which allows each node to learn an offset and the layer $h^{(-1)}$, required to calculate the first hidden state, is initialised to all zeroes. The function $f()$ is usually a non-linear function such as the following 3 functions:

1. *Sigmoid*, a special case of the logistic function bounded between 0 and 1, expressed as the following equation:

$$Sigmoid(z) = \frac{1}{1 + e^{-x}} \tag{5.2}$$

2. *tanh*, the hyperbolic tangent function bounded between -1 and 1, defined as:

$$tanh(z) = \frac{sinh(z)}{cosh(z)} = \frac{e^{2z} - 1}{e^{2z} + 1} \tag{5.3}$$

3. *ReLU*, or the positive part of the argument of a function, bounded between 0 and $+\infty$.

$$ReLU(z) = z^+ = max(0, z) \tag{5.4}$$

In each case, z represents the general input into the neuron in question.

To calculate the expected output $\hat{y^{(t)}}$ of the network at time t, apply the following formula:

$$\hat{y^{(t)}} = f(W^{yh}h^{(t)} + b_y). \tag{5.5}$$

Here, W^{yh} is the matrix of conventional weights between the output (y) and the hidden (h) layer. The vector b_y, again, is a bias parameter.

Long Short-Term Memory Neural Networks

A Long Short-Term Memory (LSTM) [41] neural network is an augmented RNN that resembles a standard RNN with a hidden layer. In an LSTM network, each node in the hidden layer is replaced by a "memory cell", of which each will be referred to with the subscript κ.

Within each memory cell is a node s_κ with linear activation. In the original LSTM

paper, this is referred to as the internal state of the cell. This internal state h_κ has a self-connected recurrent edge of weight one, often known as the constant error carousel, allowing the gradient to pass through many time steps. This is what enables the neural network to store information from previous states, allowing it to access information across several time steps.

In addition to these memory cells, LSTM networks contain gating units that, like the input node, take activation from the current data point $x^{(t)}$ as well as from the previous time step's hidden layer $s^{(t-1)}$. These units are sigmoidal units that allow flow from nodes (if the value is non-zero) and disallow them (if the flow is zero) in the same way that a gate opens and closes. In this way, these networks allow different short-term memories to be recalled at different times, with some of these memories having been forged recently and some having been forged many time steps previously.

Three types of these gates, input gates, forget gates and output gates, are included within the LSTM approach:

1. Input gates, $i_\kappa^{(t)}$, are used to regulate the flow of information into the network from the input nodes.

2. Forget gates, $f_\kappa^{(t)}$, (introduced by Gers et al. [32]) allow the contents of the internal state to be evaluated, which can be very useful in continuously running networks.

3. Output gates $o_\kappa^{(t)}$ are used to calculate the value of the memory cell $v_\kappa^{(t)}$ at time t.

The LSTM algorithm proceeds according to the following six steps, which are performed at each time step t, for each memory cell κ.

Here, $g_\kappa^{(t)}$ is an input node (as opposed to the input gate $i_\kappa^{(t)}$) that is activated both from the input layer $x^{(t)}$ at time step t and, along recurrent edges, from the hidden layer at the previous time step $s^{(t-1)}$. In addition, $h^{(t)}$ is the value of the

LSTM's hidden layer at time t, while $h^{(t-1)}$ is the values output by each memory cell in the hidden layer at time $t-1$.

$$g_{\kappa}^{(t)} = \phi(W^{gx} x_{\kappa}^{(t)} + W^{gh} h_{\kappa}^{(t-1)} + b_g) \tag{5.6}$$

$$i_{\kappa}^{(t)} = \sigma(W^{ix} x_{\kappa}^{(t)} + W^{ih} h_{\kappa}^{(t-1)} + b_i) \tag{5.7}$$

$$f_{\kappa}^{(t)} = \sigma(W^{fx} x_{\kappa}^{(t)} + W^{fh} h_{\kappa}^{(t-1)} + b_f) \tag{5.8}$$

$$o_{\kappa}^{(t)} = \sigma(W^{ox} x_{\kappa}^{(t)} + W^{oh} h_{\kappa}^{(t-1)} + b_o) \tag{5.9}$$

$$s_{\kappa}^{(t)} = g_{\kappa}^{(t)} \odot i_{\kappa}^{(i)} + s_{\kappa}^{(t-1)} \odot f_{\kappa}^{(t)} \tag{5.10}$$

$$h_{\kappa}^{(t)} = \phi(s_{\kappa}^{(t)}) \odot o^{(t)_{\kappa}} \tag{5.11}$$

For LSTM without forget gates, calculations are obtained by setting $f^{(t)} = 1$ for all t. In this case ϕ is the tanh activation function and σ is the sigmoid.

In the forward pass, when the input gate is closed (i.e. has a value of zero), no activation can get in. Similarly, when the output gate is closed, no activation can get out. As such, when both gates are closed, the activation is trapped in the memory cell and does not affect the output at that time step. In the backwards pass, the constant error carousel allows backpropagation of the gradient over several time steps. The gates, therefore, learn when to let error in and when to let it out. Typically, in LSTM architectures, layers take input from both the layer below at the same time step and the same layer in the previous time step.

Gated Recurrent Units

Similarly to an LSTM network, a GRU [20] network contains gating units. Unlike LSTM networks, however, GRU networks do not contain separate memory cells. Instead, they combine the forget gate and input gate into a single update gate $\zeta^{(t)}$. This gate defines the level of memory that is kept from the previous time steps and is defined by the following equation:

$$\zeta^{(t)} = \sigma(W^{\zeta x} x^{(t)} + W^{\zeta h} h^{(t-1)} + b_{\zeta}) \tag{5.12}$$

In addition to the update gate, GRUs use another type of gate, known as the

reset gate, $r^{(t)}$. This gate defines how new input is combined with the previous memory and is defined by the following equation:

$$r^{(t)} = \sigma(W^{rx}x^{(t)} + W^{rh}h^{(t-1)} + b_r) \tag{5.13}$$

Both of these gates depend on the previous hidden state $h^{(t-1)}$ and the input $x^{(t)}$ at time t.

The activation of the GRU at time t is a linear interpolation between the activation of the previous node and the candidate activation $\tilde{h}^{(t)}$, where the candidate activation is:

$$\tilde{h}^{(t)} = \phi(W^{hx}x^{(t)} + W^{rh}(r^{(t)} \odot h^{(t-1)}) + b_h) \tag{5.14}$$

$$h^{(t)} = (1 - \zeta^{(t)}) \odot h^{(t-1)} + \zeta^{(t)} \odot \tilde{h}^{(t)} \tag{5.15}$$

5.1.2 Method and Evaluation

As stated, for a sequence of inter-crime times in a given location in the Dyfed-Powys area, the aim of the LSTM network will be to predict the next inter-event time (and consequently, the time until the next event) in that location.

Defining Offence Location

This exercise will make use of an alternative method of partitioning location to that described in Chapter 4. In order to separate the locations in a way that will make the most sense to police within the local area, the proximity to towns within the Dyfed-Powys area can be used to define which town the crime will be assigned to. This method, while perhaps less intuitive from a data science perspective, is likely to lead to a simpler way of assigning officers in response to likely offences being committed in those areas. It also offers the possibility of excluding the low proportion of crimes (both as weighted by the number of crimes in the dataset and as weighted by population) that occur within sparsely-populated rural areas, or alternatively the low number of crimes committed by individuals resident outside urban areas, enabling the police to focus on the bulk of urban-based offences that will form the most part of their everyday work.

When defining the location to which an offence is assigned in this problem, two options are available within the data provided by Dyfed-Powys:

1. The location in which a crime is committed.

2. The location in which the offender is resident.

In this way, should Dyfed-Powys police be more interested in the likely movement of offenders within a particular location than the occurrence of offences within that same location (or vice versa), they will have the ability to select the appropriate location type to solve their spceific problem. As such, both options will be considered when looking at the results of both the LSTM and GRU Neural Networks.

When considering which offences should be considered for use within the training set, the nature and geographical spread of the towns within the Dyfed-Powys area should be considered. Due to the sparsity of the Dyfed-Powys area and geographically compact nature of its towns, it is likely that a crime that occurs some 10-20km outside of a given urban area will be included in the time sequence for that area, which may have nothing to do with the general pattern within that area at all. While the presence of these "outliers" is unlikely to be too much of an issue for closely-knit groups of towns, it may be a significant issue in low-crime towns or towns that are situated within a very sparsely-populated rural area. In Figure 5.1, an example of this distance radius in different km ranges are shown as set around the centre of Carmarthen (a town within the Dyfed-Powys area).

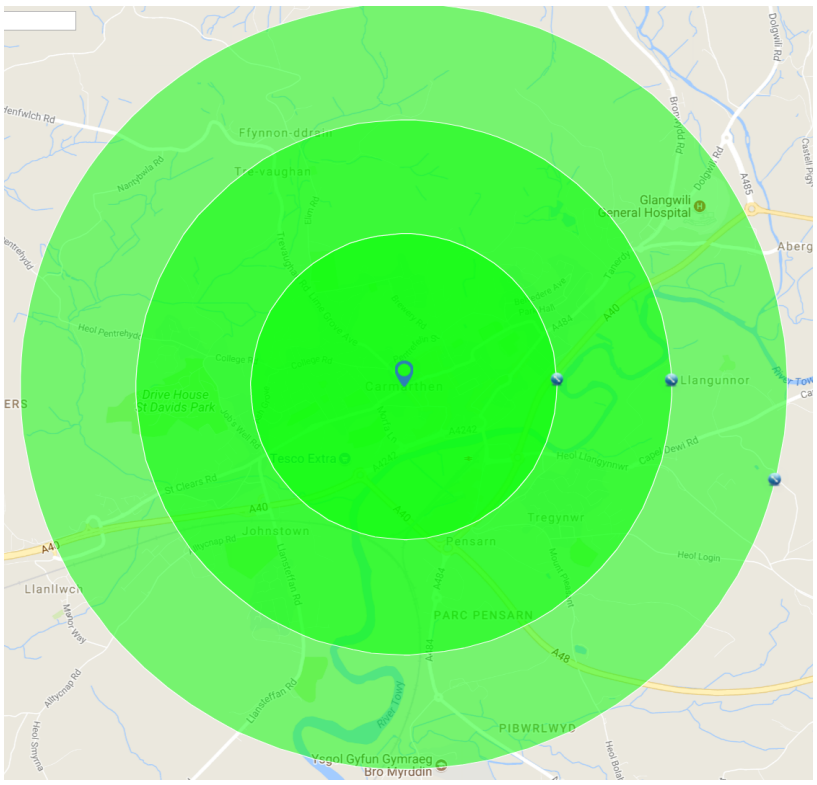

Figure 5.1: Distance Radii for Carmarthen (Carmarthenshire). Distances: 1.0km, 1.75km, 2.5km

It seems that most of the actual town of Carmarthen is contained within a 1km radius. However, since a number of small villages exist outside of this radius that could easily be considered to be part of the town, a 1km radius is likely too small. The optimum radius that includes both the town and a reasonable number of subsidiary villages seems to be somewhere between 1.75 and 2.5km, most likely around 2km. However, this radius only takes the villages within the area immediately surrounding Carmarthen into account; in order to establish the optimal radius within which offenders should be considered residents of Carmarthen, other settlements within the immediate area must also be considered.

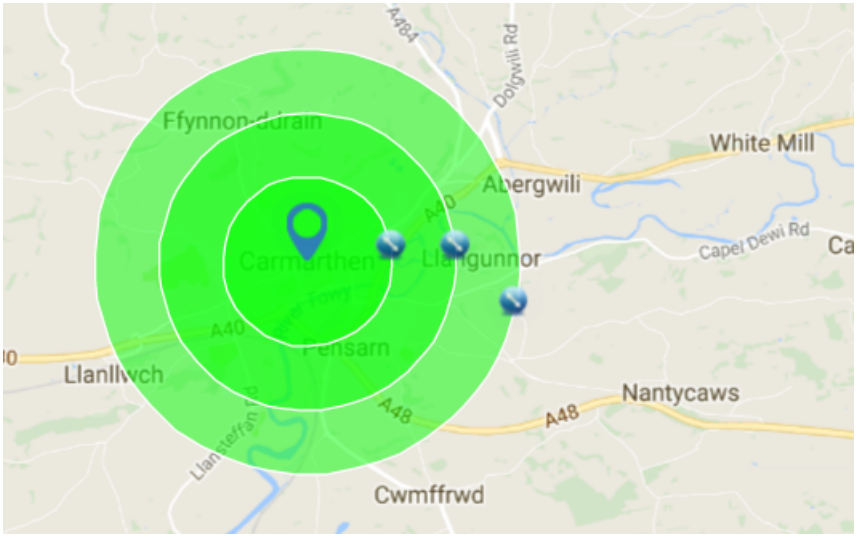

Figure 5.2: Distance Radii for Carmarthen (Carmarthenshire) and surrounding area. Distances: 1.0km, 1.75km, 2.5km

Due to the sparse nature of the surrounding area, it seems that only including the town of Carmarthen itself and the immediate area seems to be the best choice. The villages surrounding Carmarthen appear to be comparatively small and exist at too far a distance from the town itself to be reasonably considered to be part of it. As such, the decision has been made not to include these surrounding villages and instead take a 2km radius, considering the town to be a separate area. A few other towns within the Dyfed-Powys area can be treated similarly, such as Haverfordwest and the closely-situated towns of Fishguard and Goodwick.

Some towns within the Dyfed-Powys area, however, appear to be hub towns that are closely surrounded by several villages, of which some are much further away than 2km but are still clearly part of the same network. An example of a town like this is Ystradglynais (Powys), which has many villages in its immediate area, of which some are not significantly different in size to the town itself. This town and its immediate area are visualised in Figures 5.3 and 5.4 respectively.

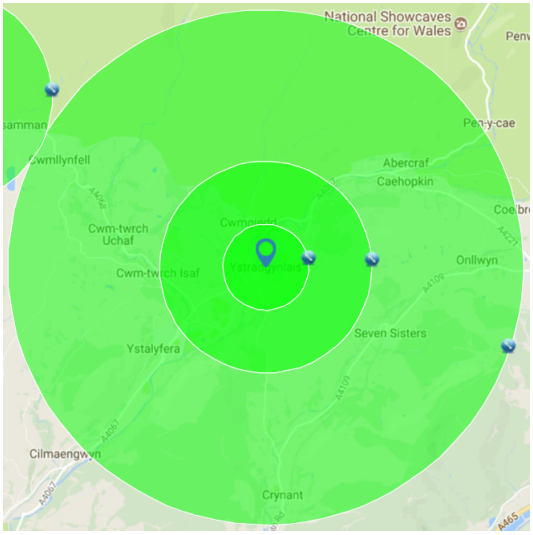

Figure 5.3: Distance Radii for Ystradglynais (Powys). Distances: 1.0km, 2.5km, 6.0km

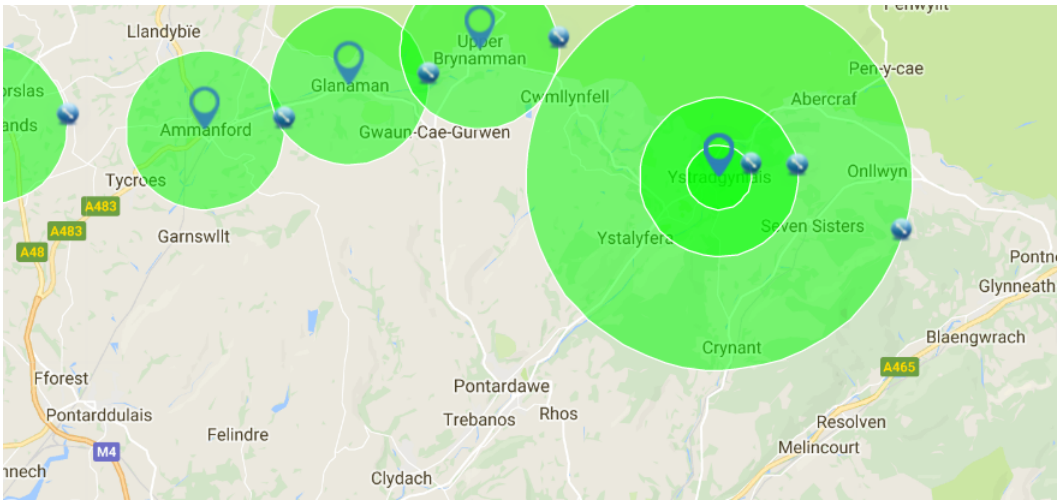

Figure 5.4: Distance Radius for Ystradglynais (Powys) and surrounding area. Distance: 2.0km

Although some offences for which the offender is resident a significant distance from the centre of any town will be included, it should still be possible to detect the patterns inherent within the time difference sequence within each location.

Defining Offence Sequences

Since both LSTM and GRU-based networks work by feeding in a set number of past training examples into the network in batches, it has been chosen to format our data in a "moving window" style. The larger this window, representing the number of past offences considered, the fewer the number of predictions that can be made on

the input data.

For high-crime locations with large reserves of past data to train on, the size of the window will not prove too much of an issue. However, for low-crime locations without these reserves of data to draw on, choosing too large a window (for example, 50 or 100 crimes) could result in only a small set of results being able to be generated, meaning that it could be difficult to decide whether this method has the potential to work for data streams from low-crime areas. As such, a balance between wanting to make use of the maximum amount of training data and wanting to make the maximum number of predictions possible must be found. This balance was checked by comparing the RMSE on the training dataset against a validation sample within the model tuning stage, then the training dataset against a holdout sample ("test set"). Throughout Section 5.1, we will focus on the performance on the training and test sets.

While conducting initial tests on this dataset, looking back to the 5 immediate previous crimes to predict the next crime was a good starting point for most locations. Making a prediction of the next crime based on more than 5 crimes significantly increased the RMSE error on the test set (representing a "holdout" sample not used for training/validation) data in most locations within the Dyfed-Powys area in the initial tests. Moreover, as a longer look-back period also necessarily results in a longer runtime, selecting the minimum possible period over which the network is able to look back on the data is crucial.

Offence Selection

Within the scope of a sequence prediction problem, many different offence time sequences can be constructed. The simplest option is to simply take a sequence of all crimes in the area, focusing on predicting any activity within that location. However, when such a sequence contains several crimes that are largely dependent on the police being in the right place at the right time (such as, for example, drug possession offences), it is uncertain as to whether these crimes should really be included in the event prediction or not. In this case, as in the Cambridge Crime Harm Index, any offences that are likely to be the direct result of police strategy (as

opposed to random chance) will not be included within the time sequence of events in a given location.

Evaluation

Before any results should be displayed for this neural network, the network (as well as the input thereof) should be properly optimised in order to give the best results. In this case, since the output of the network will be a string of numerics, a choice of metrics are available to train the network in Keras, a Python implementation of Neural Networks (see Chapter 2 for further details).

Here, as the highest level of interest is in optimising the accuracy of the predictions produced by this network and there is currently no reason to believe that the presence of outliers will significantly affect the predictions produced by the network, the simplest option is to use the RMSE (Root Mean Square Error) to evaluate both the training and test accuracy of the neural network. This is a commonly used measure of the differences between actual and predicted outputs of a model. It is defined as the quadratic mean of the differences between these actual and fitted values. A value of 0, therefore, indicates a perfect fit to the data.

Optimisation

In order to produce the best possible set of results from the Neural Networks, the layer size, depth and level of regularisation in the network must be set, as well as the length of the sequence that the network will be trained on. As these networks will be trained on a computer with limited memory space and may well be constrained by limits on available run time, it is essential that the smallest possible network that has a chance of producing a reasonable prediction is selected.

In all cases that were tested for optimisation (i.e. the LSTM and GRU networks based on either offender or offence location data), the initial number of neurons per layer in the neural network was best set as 25, while the number of layers in the neural network was best set as 3. Any more than 25 neurons in each layer and 3 layers in the neural network was found to increase the run time of the algorithm too much to offset the increase in test set accuracy, as measured by the RMSE. While this

may change in the future as the sequence to be predicted is adjusted, these initial values were found to be fairly robust in the first few tests undertaken on this dataset.

One of the biggest problems with fitting Neural Networks to datasets is the tendency of the network to overfit to the training data. As such, some sort of regularisation must be implemented within the network to regulate this overfitting. A simple way in which this can be implemented in the neural networks here is to include a dropout layer [83], which functions by randomly "dropping out" a fraction of input neurons by setting them to 0 at each update during training time. This temporarily removes the neuron and all of its connections from the network and as suggested in Srivastava et al. (as referenced above), the initial dropout rate of these networks has been set to 0.5.

Now that the initial shape and size of this neural network has been discussed, as well as the size and shape of its input data, the results of this analysis on the dataset will be presented.

5.1.3 Results

Crime Location: Multiple Offence Types

The results that will be shown in this section comprise ten distinct towns within the Dyfed-Powys area, chosen from the 59 possible towns in the four counties covered by Dyfed-Powys. This prediction error summary is designed to cover a good spread of the towns in the Dyfed-Powys area, from towns affected by seasonal fluctuations in crime (Tenby) to major towns (Llanelli), accessible small towns (Cross Hands) and remote small towns (Knighton).

Each of the LSTM layer neural networks were run for up to 250 epochs and included three LSTM layers of 25 neurons each, with dropout layers set at a rate of 0.5 included between each of the layers. These networks were then compared to a GRU neural network that was trained on the same dataset. These networks were also run for up to 250 epochs and utilised the same number of layers and neurons per layer, but due to some issues with underfitting, the percentage of neurons dropped in the dropout layers was reduced to 0.25.

The root mean squared errors for the predictions as generated for each of the chosen towns are tabulated below, alongside a measure of the total number of crimes that were considered to be attributed to that area.

Table 5.1: Neural Network Predicted Time to Next Event RMSE, All Offences Included

Location	LSTM Train	LSTM Test	Total Crimes
Aberystwyth	17.86	18.83	536
Cross Hands	18.71	19.14	572
Haverfordwest	18.16	19.40	1363
Knighton	12.59	23.57	215
Llandridnod Wells	18.08	20.12	712
Llandovery	18.80	18.97	165
Llanelli	16.74	16.92	5503
Newtown	18.97	18.40	1783
Tenby	15.51	20.14	292
Location	GRU Train	GRU Test	Total Crimes
Aberystwyth	18.05	18.83	536
Cross Hands	18.68	19.01	572
Haverfordwest	17.97	19.43	1363
Knighton	12.31	24.52	215
Llandridnod Wells	17.64	21.02	712
Llandovery	18.87	19.16	165
Llanelli	16.73	16.90	5503
Newtown	18.25	18.73	1783
Tenby	12.39	21.39	292

While these results do indicate reasonable performance for the most part, especially for rural areas where crimes are unlikely to be daily, just looking at these RMSE scores could give a false picture of the accuracy of the Neural Network if they are the result of systematic bias in the predictions. Therefore, in order to decide whether these predictions are likely to be viable, some of the individual results will need to be examined.

Here, the selected best performing location is Llanelli (Carmarthenshire) and the radial distance limit is 2km, with the LSTM Neural Network selected for its small performance boost compared to the GRU. These results can be viewed in Table 5.2 below.

Table 5.2: Time to Next Offence Predictions, Llanelli (Carmarthenshire), Offences Within 2km Radius Only

Prediction No.	Actual	Predicted
0	43.291667	19.411638
1	2.958333	19.411640
2	47.666667	19.411661
3	6.083333	19.411634
4	20.625000	19.411638
5	32.083333	19.411638
6	9.500000	19.411640
7	19.000000	19.411640
8	0.875000	19.411640
.	.	.
.	.	.
.	.	.
218	20.458333	19.411640
219	7.541667	19.411640
220	15.500000	19.411640
221	15.625000	19.411640
222	47.750000	19.411640

The prediction that the Neural Network has produced is simply (approximately) the mean Time to Event between the crimes. Without giving the network any more information than the sequence of times between each crime, therefore, to accurately model the sequence of times to event using this method has not been possible.

Crime Location: Single Offence Type

In order to test whether the addition of further information about the offence could help better predict the sequence of times to crime, the various crimes within the Dyfed-Powys area must be separated to be solely made up of one offence type. These offence types are defined and categorised under the system previously defined in Chapter 2.

The RMSE scores and predicted time to event values for a selection of these offence types and locations, representing a spread of offence types from common (Theft) to situational (Violence) to rare (Burglary) and as before, crime locations from high to low crime are included in Table 5.3. All locations here were subject to the appropriate distance radius restrictions.

Table 5.3: Neural Network Predicted Time to Next Event RMSE, Single Offence Type

Location, Offence Type	LSTM Train	LSTM Test	Total Crimes
Llanelli, Violence	19.02	19.03	398
Llanelli, Burglary (Dwelling)	18.24	17.60	56
Llanelli, Theft	18.20	19.02	127
Cross Hands, Violence	18.79	18.65	65
Cross Hands, Shoplift	18.72	21.07	15
Knighton, Violence	19.57	16.19	20
Pembroke Dock, Shoplift	21.37	22.80	104
Pembroke Dock, Burglary (Non-Dwelling)	14.12	25.22	13
Pembroke, Violence	17.82	18.59	92
Location, Offence Type	GRU Train	GRU Test	Total Crimes
Llanelli, Violence	19.62	19.45	398
Llanelli, Burglary (Dwelling)	17.65	18.04	56
Llanelli, Theft	17.38	19.58	127
Cross Hands, Violence	11.19	21.43	65
Cross Hands, Shoplift	22.65	23.38	15
Knighton, Violence	7.63	21.50	20
Pembroke Dock, Shoplift	12.59	24.45	104
Pembroke Dock, Burglary (Non-Dwelling)	10.77	22.36	13
Pembroke, Violence	17.79	18.50	92

These results seem to be reasonably consistent with those previously produced for these locations. In order to see whether or not the same problem is present here, therefore, the predictions themselves as produced by the model on the test set data must be looked into. In Table 5.4 below, the time to event predictions for dwelling burglaries in Llanelli are displayed.

Table 5.4: Time to Next Offence Predictions, Llanelli (Carmarthenshire), Offences within 2km Radius and Burglary (Dwelling) Offences Only

Prediction No.	Actual	Predicted
0	2.083333	28.956385
1	2.166667	28.956385
2	11.291667	28.956385
3	22.833333	28.956385
.	.	.
.	.	.
.	.	.
220	49.208333	28.956385
221	4.250000	28.956385
222	10.500000	28.956385

Here, the same issue has once again presented itself and that the algorithm has failed to detect any sort of pattern.

Offender Location: All Offences

Now the predictive accuracy of both LSTM and GRU neural networks has been examined on a sequential dataset based on locations as assigned by the location in which the crime was committed, the predictive accuracy of the same network for areas in which the offender was resident must be examined. As before, these networks were also run for up to 250 epochs.

Table 5.5: Neural Network Predicted Time to Next Event RMSE, All Offences Included

Location	LSTM Train	LSTM Test	Total Crimes
Aberystwyth	18.31	20.07	1528
Cross Hands	18.68	20.14	537
Haverfordwest	18.77	19.12	2029
Knighton	16.05	21.47	226
Llandridnod Wells	18.71	20.59	761
Llandovery	13.49	23.82	160
Llanelli	16.58	17.08	5828
Newtown	18.41	20.19	1979
Tenby	17.84	19.73	559
Location	GRU Train	GRU Test	Total Crimes
Aberystwyth	19.31	19.54	1528
Cross Hands	18.68	20.14	537
Haverfordwest	18.77	19.10	2029
Knighton	10.09	24.54	226
Llandridnod Wells	17.49	21.35	761
Llandovery	10.91	19.62	160
Llanelli	16.57	17.04	5828
Newtown	19.04	18.57	1979
Tenby	16.59	20.61	559

Here, the results are once again comparable to those generated by networks in the case for which an offence's location is defined by the location of the crime. Once again, to see whether these RMSE scores are at all reflective of the actual fit of the model to the training and test datasets, the neural network must be run and predictions generated, so that these can be assessed to see how well the model fits the actual times to next offence from the test set. The town of Newtown (a large town in Powys) as our example, as detailed below in Table 5.6.

Table 5.6: LSTM Time to Next Offence Predictions, Newtown (Powys)

Prediction No.	Actual	Predicted
0	6.291667	23.223602
1	47.916667	24.271477
2	0.000000	22.217215
3	41.500000	21.810350
.	.	.
.	.	.
.	.	.
584	13.166667	23.862768
585	0.000000	25.683430
586	3.583333	19.647308
587	8.291667	15.696399

Although there appears to be some improvement in the quality of time to event predictions for this particular town (i.e. the model is not just predicting one single value), the predictions produced by the model cannot truly be considered to be any better than those produced when the offences were assigned to the nearest town to the place in which the offender is resident. As this is considered by the RMSE metric to be a "good prediction", this does not bode well for the poor predictions. An example of a poor set of predictions is contained in Table 5.7 below.

Table 5.7: GRU Time to Next Offence Predictions, Llandridnod Wells (Powys)

Prediction No.	Actual	Predicted
0	36.666667	30.716028
1	0.000000	25.802147
2	21.375000	15.717809
3	44.333333	29.849543
.	.	.
.	.	.
.	.	.
219	6.666667	16.956720
220	42.875000	18.025257
221	9.125000	17.093010
222	4.416667	17.443346

The predictions of time to next offence for the town of Llandridnod Wells once again appear to be random, though perhaps not as random as the predictions generated for offender locations. It seems that here, the responsiveness of the network has increased a little, but still has a long way to go when it comes to guessing when the next offence within this location is likely to occur.

Offender Location: Single Offence Type

Once again, the time sequence of events will be altered so that each sequence is solely made up of one particular offence, as defined and categorised under the system previously defined in Chapter 2. The RMSE scores and predicted time to event values for a selection of these offence types and locations are tabulated below in Table 5.8, representing a spread of offence types from common (Theft), to situational (Violence), to rare (Burglary) and as before, crime locations from high to low crime.

Table 5.8: Neural Network Predicted Time to Next Event Raw Mean Squared Error, All Offences Included

Location, Offence Type	LSTM Train	LSTM Test	Total Crimes
Llanelli, Violence	18.53	17.97	481
Llanelli, Burglary (Dwelling)	18.96	18.89	55
Llanelli, Theft	18.15	18.83	136
Cross Hands, Violence	19.57	17.22	55
Cross Hands, Shoplift	25.05	26.21	5
Knighton, Violence	21.24	14.80	22
Pembroke Dock, Shoplift	20.00	20.72	117
Pembroke Dock, Burglary (Non-Dwelling)	19.60	20.82	6
Pembroke, Violence	17.77	19.33	90
Location, Offence Type	GRU Train	GRU Test	Total Crimes
Llanelli, Violence	19.32	18.88	481
Llanelli, Burglary (Dwelling)	14.43	21.71	55
Llanelli, Theft	18.12	18.47	136
Cross Hands, Violence	18.10	17.40	55
Cross Hands, Shoplift	18.77	25.85	5
Knighton, Violence	9.90	21.79	22
Pembroke Dock, Shoplift	20.23	22.72	117
Pembroke Dock, Burglary (Non-Dwelling)	20.49	19.87	6
Pembroke, Violence	19.08	19.44	90

Once again, the results in terms of RMSE are comparable to those generated in the previous subsections. As such, it is prudent to check whether or not the same issues with prediction are likely to occur here. As before, the predictions generated for a time sequence of dwelling burglaries in Llanelli are displayed in Table 5.9 below.

Table 5.9: Time to Next Offence Predictions, Llanelli (Carmarthenshire), Offences within 2km Radius and Burglary (Dwelling) Offences Only

Prediction No.	Actual	Predicted
0	31.750000	26.092224
1	3.333333	26.092224
2	49.583333	26.092224
3	17.666667	26.092224
4	46.583333	26.092224
5	16.375000	26.092224
6	48.333333	26.092224
7	0.416667	26.092224
8	5.833333	26.092224
.	.	.
.	.	.
.	.	.
215	26.625000	26.092224
216	39.291667	26.092224
217	28.250000	26.092224
218	15.041667	26.092224
219	10.333333	26.092224
220	16.458333	26.092224
221	20.458333	26.092224
222	23.041667	26.092224

Again, the model's tendency is to take an average time to event of sorts and predict that this will be the case in the future. This, evidently, means that these predictions are not really suitable for use in a policing context.

5.1.4 Conclusion, Limitations and Further Research

Conclusions

For this dataset, though this may not be the case for all crime datasets, it seems that using an RNN to predict time sequences of all offences within a given location in this way does not yield any appropriate results. Restricting the radius of offences included within a town does not appear to help smooth out these time sequences, although the concept of restricting the attributed crimes to simply the urban area itself would likely be beneficial from a policing point of view. Restricting the offence types to a single offence also does not help to smooth out the time sequences, or at least does not produce an appropriate set of predictions. Looking at the data, there are two possible reasons for this.

1. Firstly, it is likely that an LSTM neural network is not an appropriate way to model this dataset. The inherent limitations of this method as it relates to this dataset will be discussed further below.

2. Secondly, and perhaps more possibly, it seems likely that there is no real general pattern in the pattern of time differences, however intuitively the offences are distributed in each location.

It is, therefore, entirely possible that either:

1. Modelling the likely time to the next crime within a given location requires a more complicated approach based on a probability distribution or process, or a different use of a RNN for predictive purposes.

2. Attempting to predict the future movements of criminals within a given location by constructing a sequence of times to the next event is entirely inappropriate and therefore, predictions of future activity within rural locations such as these requires a different approach.

In the first case, it may well be that fitting a temporal distribution or process to the dataset will yield appropriate predictive results. However, since such statistical analyses are not the focus of this thesis, these will not be discussed here. A further suggestion for an alternative implementation of a RNN in this context will, however, be outlined in the following section. Prior to the discussion of an alternative method by which it may be possible to make use of this technique on this dataset, however, the limitations of the dataset in the context of this sort of prediction must be considered.

Limitations

For this method, the biggest limitation that is likely to be present within the dataset is the small size of the dataset in each of the given towns. With many towns being fairly low-crime areas, and the high-crime towns only yielding a total time to event sequence of length 1000-2000 when all offences are included within the remit of the

predictions, it is clear that Neural Networks (a method for which the learning improves incrementally with a larger number of examples) may well not be suitable as a method of pattern detection for this dataset. As the large majority of towns within Dyfed-Powys' jurisdiction are low-crime locations, RNNs are probably not the most appropriate method for this kind of time-sequence prediction task. However, in high-crime locations with a rapidly growing dataset such as Llanelli, it may well be worth attempting to predict the likely time to next crime within the town using this method.

Another obvious limitation of this method is that it cannot take into account the possible effect of ongoing offences that are occurring in other surrounding (or possibly not surrounding) towns in the Dyfed-Powys area. Although it is possible to build a Neural Network that takes the occurence of crimes within other areas into account, it is unlikely that the level of computing power that is likely to be available to Dyfed-Powys police will support such an analysis in the near future.

Further Research: Binary Event Window Prediction

In order to predict the time at which a subsequent offence is likely to occur, it is actually possible that the problem is better framed as one of binary classification, in which an event either will or will not occur within a given time window. In such a way, it can be predicted whether or not a crime is likely to occur (or in fact, whether a small or large number of crimes are likely to occur) in a given location. While it is possible to fit a feedforward neural network to this type of problem, a LSTM or GRU neural network in a similar way to the above problem is more likely to be appropriate here, as a prediction as to whether or not a crime is likely to occur in a given time window may well depend on the occurrence of crimes in a long sequence of time windows prior to that prediction. Moreover, it may depend on the occurrence of crimes in a large series of other locations in the Dyfed-Powys area. With binary indicators acting as "sensor measurements" for the occurrence of an offence in a particular time window in a particular town within the Dyfed-Powys area, it is possible to include the occurrence of these offences as predictor variables in a multivariate LSTM or GRU, then build the network in Keras. From this, a prediction as to whether or not an offence will occur within that particular time

window can be produced for the next timestep in each location.

To create appropriate time intervals in which a crime can be predicted to occur, the observation period could be (for example) divided into 8-hour windows, such that each day is (for reasons discussed in Chapter 2) considered to begin at 05:00. The 8-hour windows can, therefore, be defined for each day between 05:00 and 13:00, then 13:00 and 21:00, then 21:00 and 05:00 the following day. These time windows define morning, afternoon and overnight crime respectively and can either be taken for all detectable crimes or crimes of one particular type, depending on the desired outcome. From a policing perspective, keeping the time windows in which an offence will be predicted to occur to a small number of hours (preferably hours that are aligned with police shifts) will allow Dyfed-Powys police to react more spontaneously to crime and also to predict when an offence is likely to occur with a greater amount of certainty.

While this method certainly merits investigation, the computational and time limitations likely to be present in a police setting render this method unviable for this dataset. The main reason for these computational issues is simply the construction of the dataframe that is required to create the training and test datasets, which is considerably time-consuming and becomes increasingly large and difficult to construct as time goes on. While this issue could be solved by generating a number of dataframes that span around 200 timesteps (this is said to be the maximum number of timesteps that a LSTM neural network, and therefore in all likelihood a GRU, is able to process) and simply iteratively adding and taking away rows from the dataframe as time goes on, this method would require a large amount of testing and adjustment in a policing context - such investigations are beyond the scope of this thesis.

Further Research: Offender Survival

Returning to the concept of treating offences occurring within Dyfed-Powys as "sensor measurements" of criminal activity, it is possible to produce a second type of prediction from this kind of multivariate LSTM. By shifting the focus of the prediction from the behaviour of a location to the behaviour of an individual offender,

it is likely to be possible to predict the chance of an individual offender choosing to reoffend as they move through various time windows. Beginning with just "sensor measurements" relating to the criminal activity in the town in which the offender is resident, this can be extended to include all towns bordering on the offender's location, all towns within a certain km radius or all towns that by the measure constructed in Chapter 4, are deemed to have a certain minimum level of similarity with that particular town. In this way, the two branches of research discussed in this thesis can be joined together, allowing predictions to be made on individual offender behaviour from the likely behaviour of offenders in that same location, as well as in several other locations within the Dyfed-Powys area.

Due to the computing power and time required to construct the dataset required for this model, a full examination of the exact implementation of this technique once again lies beyond the remit of this thesis, as these constraints likely make it currently unsuitable for implementation in Dyfed-Powys. However, when considering the application of this technique to other forces or to Dyfed-Powys at a future date, it may well be worth considering fitting an RNN to a crime dataset in this way.

5.2 Area Classification Score

In this section, a stacked model will be made use of to consolidate the various factors relating to either the crime's or offender's location into a single, unified score, then use this as a factor in the Random Forests model for reoffence prediction as previously described in Chapter 2. This particular form of consolidation will take the shape of an Area Classification score, which will aim to rank the locations in Dyfed-Powys in terms of the relative offence risk of the area in question. Like the PCA previously completed in Chapter 2, this can be used as a method of reducing the number of dimensions in the model such that the minimum number of variables might be used. While it has previously been attempted to reduce the dimensionality of the WIMD variables within this model (see Chapter 2 for further information), it has not yet been attempted to truly relate these to a measure of risk.

This method is not currently commonly used within the field of predictive policing. It is most useful when considering a number of potentially predictive variables

relating to a singular topic. In this case, a number of predictive variables relating to the area will be considered in which either the crime occurs or the offender is resident, of which some have been shown in previous models to be significant. It also has the advantage over PCA (see Chapter 2) in that both numerical and categorical variables can be used as input into the model - in fact, any factor that can be made suitable for an XGBoost model can be used. Moreover, when considering restrictions on holding personal data resulting from GDPR, developing score factors that are based on statistics and counts held on an aggregate basis will become an increasingly important part of producing models that can accurately predict an offender's probability of reoffending.

While this feature will potentially require maintenance over time as the relative offence risk associated with an area alters and although a relatively small number of potentially predictive variables relating to either the crime's or offender's location will be used, it is possible that in the future (due to large volumes of publicly available location-based data at various aggregate levels on the StatsWales [36] website), Dyfed-Powys may be left with more location-related variables than they can reasonably deal with. In this case, making use of these techniques will help them to reduce the number of variables related to this subject in their model to a single aggregate measure of offence risk per location.

5.2.1 Algorithms for Prediction

The model to be built to construct and make use of this area score in the context of reoffence prediction will be a stacked model with the following three component parts:

1. An XGBoost model that will be used to estimate the relative offending propensity of the population within a given location.

2. A clustering algorithm that will take the predicted propensities and devise a series of clusters based on these propensities.

3. A Random Forest model, as previously described in Chapter 2, that will take

these clusters as an input factor and will output a probability of reoffence for a given offender based on these factors.

In the case of the XGBoost algorithm, the response variable will be the aggregate yearly offence count within each individual location; due to the choice of location, weighting these counts by population is not strictly necessary. This response will be evaluated against the actual aggregate yearly offence counts per location for its suitability, on which the focus will be on capturing the correct pattern of high/low crime counts between individual LSOAs rather than necessarily producing an accurate prediction of the exact crime counts - as such, systematic errors in the crime counts, which may be present for many reasons, will be tolerated so long as the overall pattern is captured to a reasonable degree of accuracy.

These predictions of the area's overall risk, once assessed for their suitability, will then be fed into the Affinity Propagation clustering algorithm previously described in Chapter 4 in order to produce a number of clusters representing areas that should have similar propensities to offend within their population. In this case, to make use of a small but unspecified number of clusters and therefore a reasonably coarse measure of overall location risk. This is due to the following reasons:

1. Within their jurisdiction, Dyfed-Powys are exposed to a much smaller number of locations than the average insurer would generally be exposed to for one of their products. As such, a much smaller number of clusters is likely appropriate.

2. As there is no any past reference for this problem, the approach can be taken to iterate a particular algorithm over a number of cluster choices, or an algorithm can be made use of that will automatically select the number of clusters for us, so long as individual parameters are tuned in order to select the optimum number of clusters. From a policing use perspective, it would generally be preferential to use the automatic selection method as this would be much simpler for analysts within Dyfed-Powys police to bring into regular use.

Once these clusters have been generated, they will be joined back on to the original dataset via a code assigned to the location. This will then be input into the current

best-performing Reoffender model from Chapter 2 (Random Forests) in the place of the relevant area-related WIMD factors, be these relating to either the location in which the crime was committed or the location in which the offender is resident. The performance of this model can then be evaluated in the same way as all models previously constructed in Chapter 2.

5.2.2 Method and Evaluation

The method by which this model will be defined and constructed is detailed below. Firstly, the precise method by which locations are partitioned will be defined, then the suitability of using past years' crime counts relating to the occurrence of particular types of crimes within an individual area will be discussed. Finally, the performance of the initial XGBoost regression model and the final Random Forests classification model will be evaluated.

XGBoost: Defining Offence Location

Instead of Postcodes or nearest towns, the location of a crime or an offender will be defined by the LSOA (Lower Level Super Output Area) in which they are either resident or have committed the crime. These areas are partitioned such that they have approximately the same population (1500) per LSOA, which means that there is no need to consider weighting the area by population when looking at offence counts per area. For the purposes of this exercise:

1. Locations in which the crime was committed, including all available crime LSOAs within Dyfed-Powys. Only a very small number of crimes within the original dataset are committed outside Dyfed-Powys jurisdiction and those that are tend to occur within bordering areas. As such, all available locations have been included.

2. Locations in which the offender was resident, including all available LSOAs in Wales. Offenders resident outside of Wales will not be considered for the purposes of this exercise, but offenders within Wales will be - this is due to the lack of consistent deprivation statistics between England and Wales.

All offence types will be included within the counts for the input dataset, subject

to general conditions already stipulated in Chapter 2 (as a reminder, TIC offences and those that do not lead to prosecution are not included in these counts). Once again, it must be borne in mind that the location data within the original Dyfed-Powys dataset, which has been used in order to test this method, is incomplete and therefore, this analysis inherently carries the following risks:

1. Systematic bias in the locations in which crimes have a missing or an indeterminate location - it may, for example, be more common to have a missing postcode for crimes within certain towns (particularly those that encompass multiple postcode locations), or a certain type of location.

2. Systematic bias in the dates for which crimes with missing or indeterminate locations are recorded - it may, for example, be more common to see missing postcodes in earlier years, which may artificially depress the yearly counts for those years in which a greater number of crimes with missing or indeterminate locations are recorded.

3. Changes in data collection methods altering the general patterns of aggregate counts. This is likely to be less of an issue if these methods vary between location, but could negatively impact the predictions output by the final Random Forests model if, for example, the under-reporting of crimes within one location lead to it being considered to be a low-risk area when it is really a high risk area (and vice-versa).

Due to a number of location uncertainties present within the original dataset as provided by Dyfed-Powys, it is uncertain as to whether or not the performance of this algorithm would be improved or would deteriorate in a live setting and can only state that points 1 and 2, while a concern in the original dataset, are not likely to be a concern in a live environment.

Point 3, however, is likely to be a concern in the live environment if Dyfed-Powys cannot be sure that all of its forces are adhering to the same data collection standards. As such, for this model stack to be appropriate for use in a live environment, it must be reasonably certain that consistency in data capture between the various forces is deployed across the Dyfed-Powys area. For the purposes of this thesis, this particular assumption will hold.

XGBoost: Defining Time Periods

Within our original dataset as first provided by Dyfed-Powys, 12 months of crime data will be used to train the XGBoost regression and the next 12 months to test its accuracy on unseen data. This test will be completed for 5 consecutive years (2009 to predict 2010, 2010 to predict 2011 and so on until 2013 to predict 2014), for all crimes as described above. These particular time windows have been chosen due to changes in:

1. The recording of crime.

2. The structure of the database.

3. Overarching police strategy throughout the period.

Due to changes in these three factors over time, it is unlikely that multiple years of data will be used to build a reliable Area Classification Score at this point. Moreover, as crime counts from previous years as factors within the training dataset will be used, it is entirely possible that the algorithm will overfit to past counts, which will most likely only be relevant for those locations whose population and strategy has not significantly changed over time. While whole calendar years will be used for the purpose of this proof of concept analysis, this model stack is more likely to be trained on a rolling monthly basis and as such, would expect the model to be trained on the latest 12 months of count data available to Dyfed-Powys police. In our dataset, this is the entire calendar year of 2014.

In Table 5.10 below, the performance of those models will be compared with ex-

tra past counts added as factors to those which simply train on the previous year's data.

Table 5.10: Comparative Performance of Aggregate Crime Count Predictions from WIMD, Location of Crime, RMSE

Years	Test RMSE (+ 0 Past Years)	Test RMSE (+ 1 Past Year)
Train 2012, Test 2013	21.21591	21.46737
Train 2013, Test 2014	15.16637	17.57449
Years	Test RMSE (+ 2 Past Years)	Test RMSE (+ 3 Past Years)
Train 2012, Test 2013	21.80892	21.81006
Train 2013, Test 2014	16.83547	16.13347

As no improvement in performance (and in most cases, a slight deterioration) was observed in the fit of the model to unseen data with these factors added, it has been decided not to include these past counts in the predictive factors of this dataset. Moreover, in all cases, the variables with the highest importance in these models were those previous years' crime counts, meaning that it is likely the model was significantly overfitting to these factors and ignoring other factors that may well be more predictive on unseen data.

The main issue with only making use of the previous year's crime counts in order to predict the following year's is that many of the LSOAs in question may have either a very low number of or no crimes attributed to them in a particular year. In order to reduce the effect of this particular issue on the results of the data, it has been decided to group all crimes that have a crime count of less than 20 within a particular location into a general "low-crime group". These are removed before the XGBoost is allowed to fit to the data.

In a live setting, therefore, the expectaion would be for Dyfed-Powys to make use of the most recent WIMD variables (this will now be later than 2014) and the corresponding most recent 12 months of crime counts as input data, excluding those locations with less than 20 crimes in the previous year, into the XGBoost algorithm. The XGBoost model should then be trained on that data, clustered and the clusters joined on to the latest batch of offender data suitable for training (the nature of which has been discussed at length in Chapters 2 and 3).

XGBoost: Defining Appropriate Variables

The variables to be included in the dataset to be input into XGBoost will be all the variables that have been obtained from the WIMD dataset, except those that already describe some sort of crime propensity within the LSOA. These have not been included as they have not been properly back-dated - while it would be possible to include variables relating to crime propensities or crime counts by LSOA for the correct years, these have not been included in this model as it is likely that the model will overfit to these factors.

Despite not including these overall crime propensity indicators in our final model, a report on a small matter of interest from an XGBoost model is intended to include these indicators of criminal activity. The Top 5 variable importances from that model are detailed in Table 5.11 below.

Table 5.11: Top 5 Variable Importances from WIMD Data, Location of Crime, Train 2013, Test 2014

Variable Name	Importance
CrimDam_per100	0.48648
Violence_per100	0.40922
KS4_Points	0.02035
Absentee_Percentage	0.01751
DeathRate_per100000	0.01452

When looking at the variable importances here, the Criminal Damage and Violence per 100 individuals within an LSOA are the most predictive variables - this is unsurprising, as these counts will contribute directly towards the overall count of offences in the LSOA area. What is interesting, however, is that these two types of offences are selected as being a far more important predictor of the offence counts per location than any other crime type. Two possible conclusions can be drawn from this information:

1. That the majority of offences within the Dyfed-Powys area are simply more likely to fall into one of these two categories, meaning that the propensity of the individuals within an LSOA to commit these offences would in fact be a good proxy for the overall propensity to offend.

2. That the propensity of the individuals within each LSOA to commit one of these offence types relates directly to their propensity to offend in general, meaning that it may be better constructing an area score based around one or both of these offence types separately.

Without determining which of these is the case or (most importantly) how these offence categories are determined in the WIMD data, however, it will be difficult to determine whether or not a peril rating-style set of area classification scores intended to predict an individual's propensity to offend in different ways should be made use of per LSOA. For the purposes of this thesis, this question will be left open as it is likely a larger number of potentially predictive area-related factors will be required to produce an offence type-level area score.

Evaluation

The first method of evaluation for this offence score will be to evaluate the predictive power of the initial XGBoost model on the aggregate reoffence counts. This will be done by examining the overall RMSE of the predictions, then analysing the spread of actual vs. fitted results to determine if these overall RMSE errors are due to poor model fit or some sort of systematic difference between one year and the next. It is entirely possible that, for example, overall crime counts in the Dyfed-Powys area might increase or decrease substantially from one year to the next, causing an overall systematic error in the predicted crime counts year on year. So long as this movement is generally systematic and does not occur in several individual locations (i.e. locations moving suddenly from being a low to a high crime area), it is unlikely to significantly affect the performance of the clusters as a factor in the final model and so is unlikely to be worrying.

In this case, the effectiveness of the clustering algorithm by its performance will be evaluated in the Random Forests model for reoffence prediction. This will be evaluated in the same way as it was in Chapter 2 and its results (in terms of the various metrics detailed within that Chapter 2, Section 4) compared against the "best case" results from Chapter 2, Section 5.

5.2.3 Results

This results section will begin with the results of the XGBoost model designed to predict crime counts, as evaluated for both the location in which the crime occurs and the location in which the offender is resident. It will then continue to an evaluation of the final Random Forest, which will be completed in the same manner as Chapter 2's evaluation. Finally, a comparison of these results against the original "best case" results will be completed.

XGBoost - Crime Locations

In this section, the predictive accuracy of crime count predictions produced by the XGBoost algorithm will be evaluated in order to determine whether these can be reasonably be used in the final reoffending model. Here, the number of crimes will be predicted that occur within a location. The errors in these predictions for 5 consecutive Train/Test years are detailed in Table 5.12 below.

Table 5.12: Aggregate Crime Count Predictions from WIMD, Location of Crime, RMSE

Years	Train RMSE	Test RMSE
Train 2009, Test 2010	9.9008	22.8262
Train 2010, Test 2011	10.1818	27.6983
Train 2011, Test 2012	9.0396	18.8574
Train 2012, Test 2013	10.6337	21.2159
Train 2013, Test 2014	12.5978	15.1664

The RMSE of the training set is relatively low in each case, at a maximum of 10 crimes. The RMSE of the test set is generally higher, with the predictions from 2010 on 2011 data having the highest RMSE on the test set. The model that has been constructed in order to predict the 2014 crime counts generalises the best to the unseen data - this is most likely due to the better lineup between the location state captured by the 2014 WIMD data and the one present at the time of the 2014 offences. The lower RMSE on the 2012 data, however, cannot be explained by this - in this case, it's possible that the distribution of crime counts in 2011 and 2012 responded similarly to the deprivation state of the LSOA and that little changed in terms of policing strategy between those two years.

In order to properly interpret the overall results, it is necessary to look more closely

at the distribution of predicted vs. actual crime counts within the area to see whether there are any systematic errors in the predictions of crime counts and moreover, whether the overall actual crime counts per LSOA are high enough to support this level of test set error. The 2012-2013 and 2013-2014 years will be selected as examples - at first, an examination of the fitted vs. actual comparisons between the predictions made on 2013 data from 2012 data will be provided.

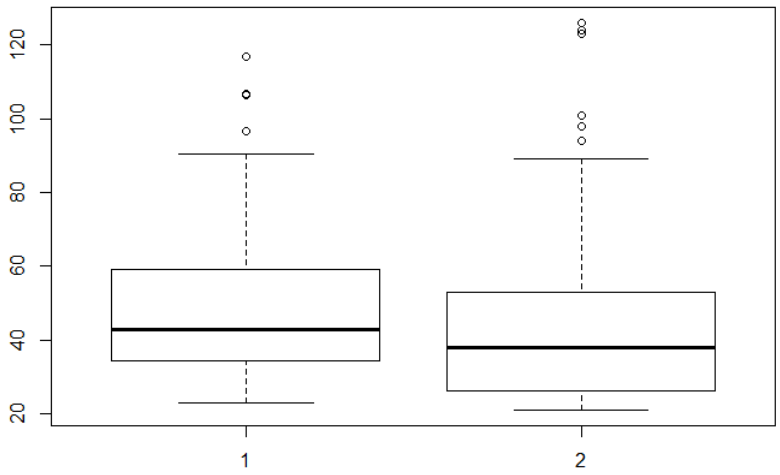

Figure 5.5: Fitted (1) vs. Actual (2) Boxplot, Train on 2012 to predict on 2013

Here, the median of the fitted crime counts is very close to the actual median - both are around 40 crimes per year per LSOA. The interquartile range of predictions is, however, placed much higher for the fitted values than the actual values and the maximum number of crimes predicted is also higher than the actual. This could be due to an overall decrease of around 500 crimes over all LSOA areas between 2012 and 2013 - in this case, so long as the distribution of crimes stays relatively consistent between 2012 and 2013, the predictions here will still be useful as a risk score.

To determine whether or not this is the case, the scatterplot of fitted vs. actual crime counts by LSOA for 2013 will be examined. A line, indicating the case in which fitted crime counts = actual crime counts, is provided on each of the plots for comparison.

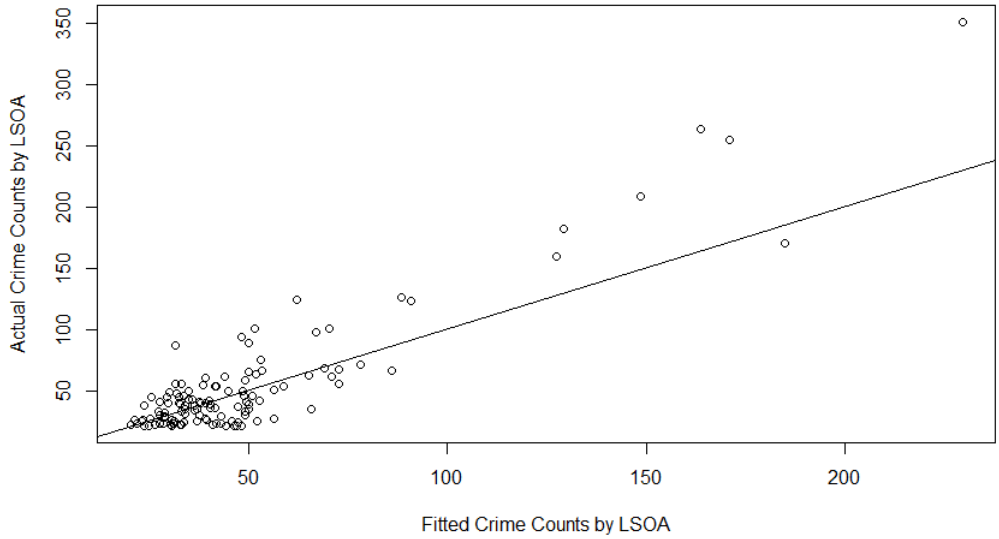

Figure 5.6: Fitted vs. Actual Scatterplot, Train on 2012 to predict on 2013

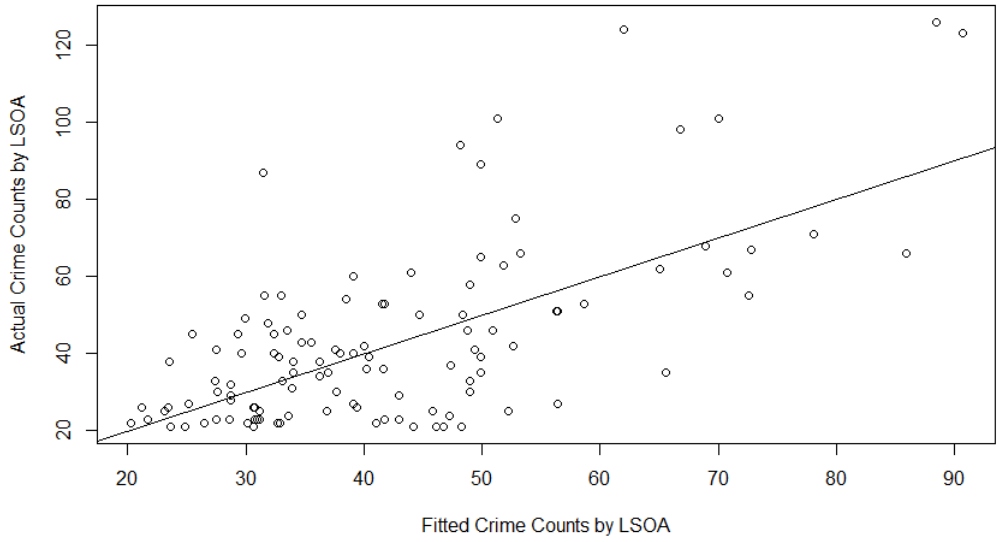

Figure 5.7: Fitted vs. Actual Scatterplot, Train on 2012 to predict on 2013, LSOA with < 150 crimes only

While Figure 5.6 appears to show a reasonable correspondence between actual and fitted values with a small systematic overestimation of the total number of crimes per LSOA, on focusing on those LSOAs with less than 120 crimes per location, it is evident some substantial prediction errors are occurring for those areas predicted to have less than 80 crimes per LSOA. In many cases, the model will significantly overestimate the number of crimes per LSOA and in a few, it will sig-

nificantly underestimate the level of criminal activity in the area. The correlation coefficient between the fitted and actual results is 0.6544, indicating a relatively weak correlation.

The best performing model, which was trained on 2013 data to predict the 2014 counts, is detailed below. A boxplot of the Fitted and Actual results is included below.

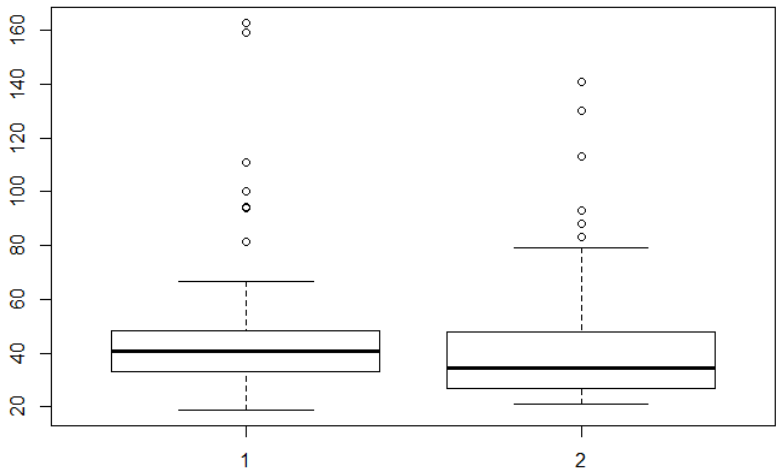

Figure 5.8: Fitted (1) vs. Actual (2) Boxplot, Train on 2013 to predict on 2014

The interquartile range of fitted counts is much smaller than that of the actual counts - this means that overall, the model has a tendency to predict far more closely to the median crime count than would be actually observed. In general, the model will produce more extreme results than those actually observed at the top and bottom end - the range of fitted count values is much larger than the range of actual counts.

While the predictions made on 2013 for 2014 appear to be somewhat reasonable, there may be an issue present in the fitted results whereby too many predictions are erroneously made around the median crime count.

In order to check whether this is likely to be the case, the scatterplot of fitted vs. actual results must be checked for this particular set of years. Again, a line

representing the case in which fitted crime counts = actual crime counts is provided on each of the plots for comparison.

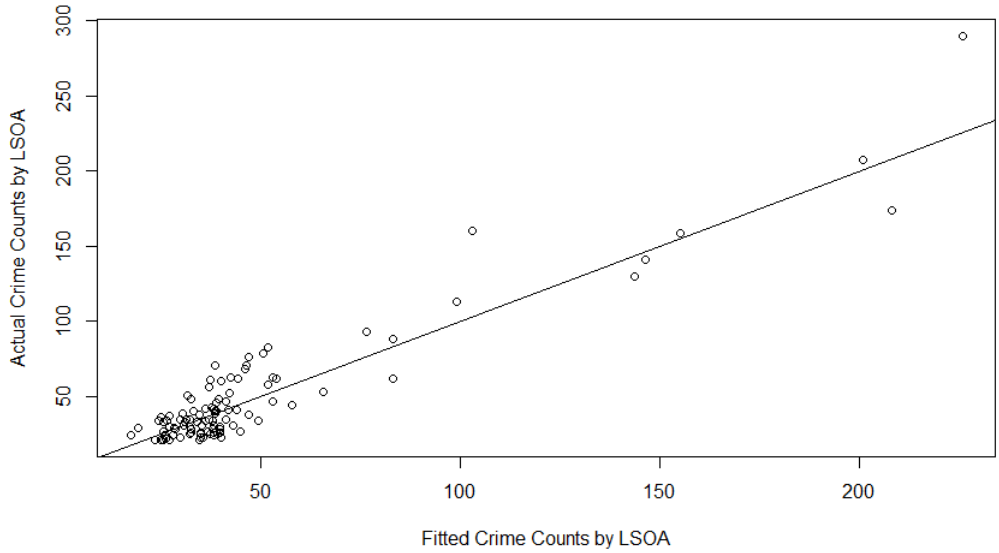

Figure 5.9: Fitted vs. Actual Scatterplot, Train on 2013 to predict on 2014

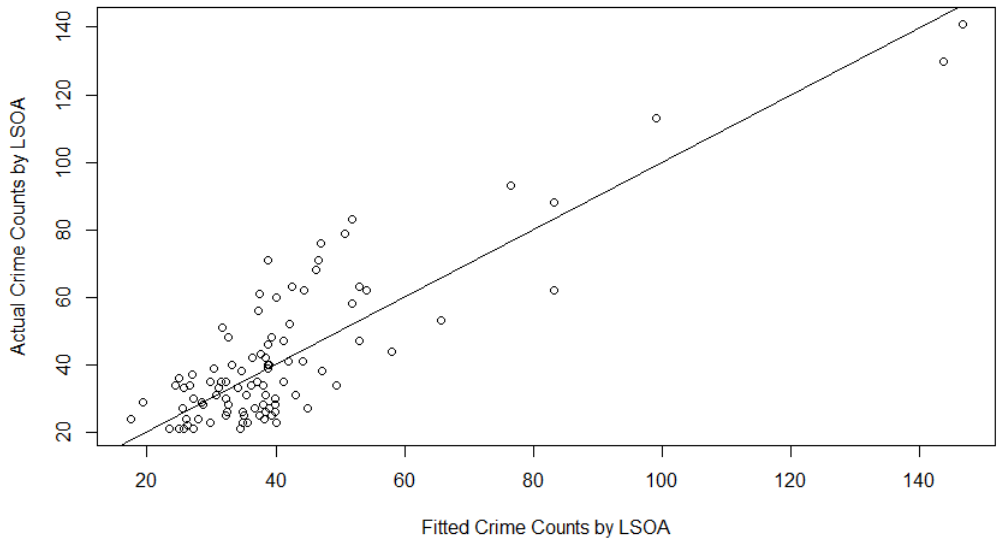

Figure 5.10: Fitted vs. Actual Scatterplot, Train on 2013 to predict on 2014, LSOA with < 120 crimes only

Compared to the 2012 - 2013 plots, a much larger number of locations are experiencing low crime counts. Once again, the fitted values are systematically lower than their actual counterparts. When focusing on those LSOA with less than 150 crimes, there are no particular trends in the under/overpredictions made and for the

most part, there appears to be a reasonable scatter around a straight line, meaning that some sort of consistent pattern is likely to be found, even at very low crime rates. This is reaffirmed by a much stronger correlation coefficient of 0.8475.

Evidently, therefore, there is some disruption to the predictive patterns at the "low likelihood of offending" end - however, it is still possible that these issues may not be too disruptive when it comes to predicting an offender's probability of reoffending. It will be examined whether this is likely to be the case in Section 2.3.4 - for now, however, this analysis will be repeated for the location in which the offender is resident.

Before an appropriately constructed set of clusters is fed into the Reoffender prediction algorithm, the locations that have been clustered together should be reported on, based on the latest year of training data (in this case, 2014), as well as the variables that are considered to be the most important in determining the level of crime that is committed within an LSOA. A list of these variable importances is given in Table 5.13 below.

Table 5.13: Aggregate Crime Count Predictions from WIMD, Location of Crime, Variable Importances

Variable	Importance
Education_NoQuals_Percentage	0.2696
Health_DeathRate_per100000	0.2511
Employment_Benefits_Percentage	0.1728
Health_Cancer_per100000	0.1217
Education_KS4_Points	0.0365
Health_LowBW_Percentage	0.0344
Education_Absentee_Percentage	0.0334
Income_Deprivation_Percentage	0.0312
Education_KS2_Points	0.0175
Employment_LTSick_per100000	0.0116
Education_KS4L2_Percentage	0.0112
Education_NotInHE_Percentage	0.0088

The propensity of an LSOA to have crimes committed within its borders is most heavily affected by the percentage of individuals resident within the area that have no formal qualifications and the death rate per 100,000. The percentage of individuals within the area on employment benefits and the cancer rate per 100000 are also reasonably important within this context.

The mean crime counts by LSOA of the 13 clusters produced by the Affinity Propagation algorithm are as follows:

Table 5.14: Mean No. of Crimes per LSOA

Cluster	Mean No. of Crimes per LSOA
1	44.28571
2	135.50000
3	35.10000
4	62.28571
5	29.85714
6	37.83333
7	113.00000
8	53.00000
9	26.50000
10	53.71429
11	81.00000
12	28.83333
13	27.76923

XGBoost - Offender Locations

In this section, the predictive accuracy of crime count predictions produced by the XGBoost algorithm will be evaluated in order to determine whether these can reasonably be used in the final reoffending model. Here, the focus will be on predicting the number of crimes that are committed by offenders within a given location.

Table 5.15: Aggregate Crime Count Predictions from WIMD, Location of Offender, RMSE

Years	Train RMSE	Test RMSE
Train 2009, Test 2010	3.32687	17.25617
Train 2010, Test 2011	4.20642	14.13780
Train 2011, Test 2012	4.05053	14.14595
Train 2012, Test 2013	3.44800	13.53449
Train 2013, Test 2014	10.2455	12.09478

When compared to the RMSE scores based on the LSOAs in which crimes were committed (given in the previous subsection), the Train RMSE scores are reasonably similar, but the predictions made on the unseen dataset are somewhat closer to the actual values in all cases. There is also much less variation in the RMSE scores on the test dataset in general. Once again, the best unseen data prediction is found for

the train in 2013, test on 2014 case - this, again, is not surprising due to the nature of the WIMD 2014 data.

In order to determine whether these particular errors are reasonable, however, the results themselves must be examined more closely - this will begin with the fitted results for 2013, trained on 2012 data.

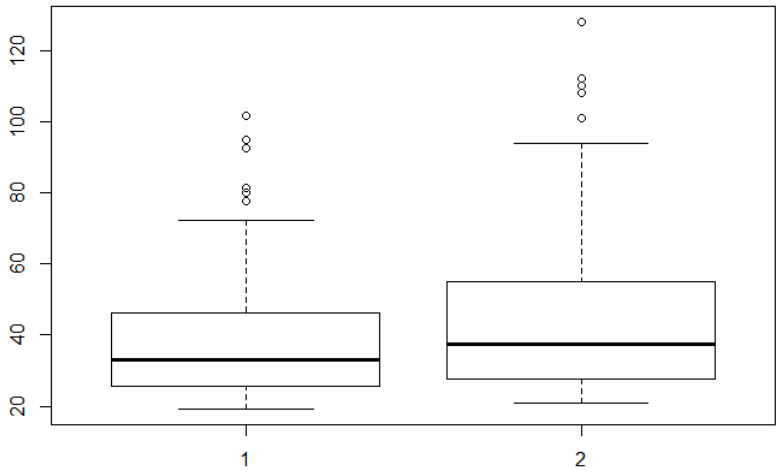

Figure 5.11: Fitted (1) vs. Actual (2) Boxplot, Train on 2012 to predict on 2013

The median of the fitted crime counts is very close to the actual median - both are around 35 crimes per year per LSOA in which the offender is resident. The interquartile range of predictions is, however, placed much lower for the fitted values than the actual values and the maximum no. of crimes predicted is also lower than the actual. Again, this also could be due to an overall decrease of around 500 crimes over all LSOA areas between 2012 and 2013. To determine whether or not this is the case, the scatterplot of fitted vs. actual crime counts by LSOA for 2013 is provided below.

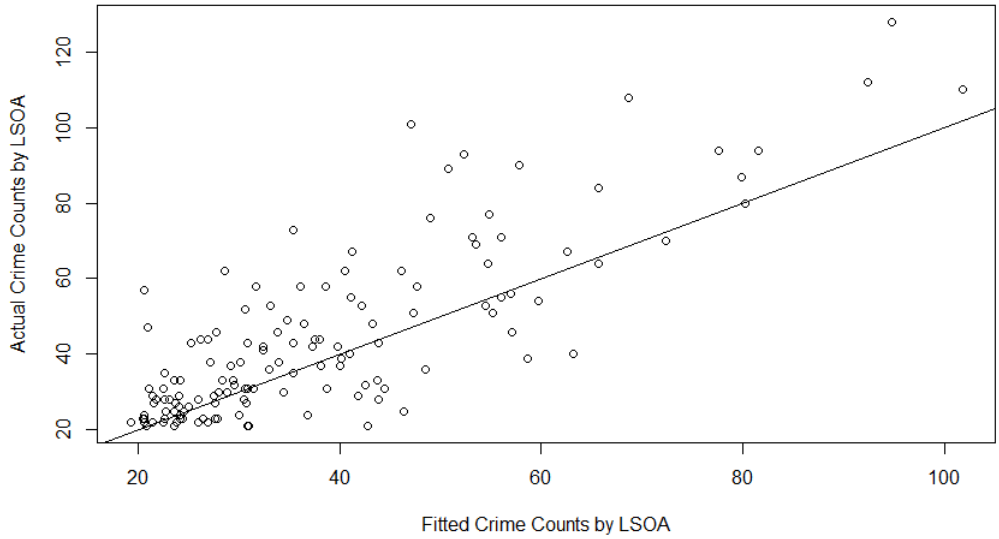

Figure 5.12: Fitted vs. Actual Scatterplot, Train on 2012 to predict on 2013

In general, there are not too many large discrepancies between the actual and fitted counts and that (as previously stated), the fitted counts for 2013 are systematically slightly lower than the actual crime counts for that year - this comes despite an overall decrease in the actual total crimes reported in 2013. This model seems to produce a slightly tighter scatter than the comparable model produced for crime locations, reflecting the overall lower RMSE. While there are a few cases where a large discrepancy can be observed between the fitted and actual crime counts, overall it does not appear that there are any particular areas of concern when it comes to the model's ability to capture the patterns in the crime counts in the 2013 data. This is backed up by the correlation coefficient between the fitted and actual results, which is 0.8162.

Having examined the fitted and actual results for 2013 unseen data, the same examination will be presented for unseen data from 2014.

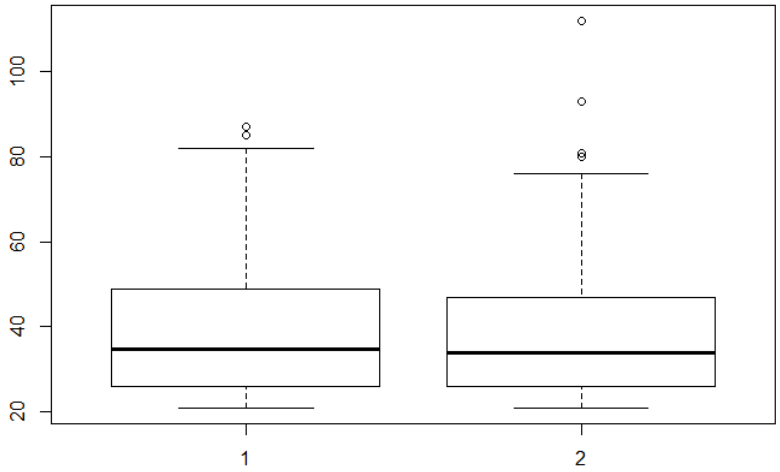

Figure 5.13: Fitted (1) vs. Actual (2) Boxplot, Train on 2013 to predict on 2014

The first observation that can be made is that the interquartile range and median crime count for both the fitted and actual results are very close - while the interquartile range is slightly wider for the fitted results, indicating a slightly larger dispersion of results than would occur in reality, the predictions produced by the 2013-trained model are very close to the actual data.

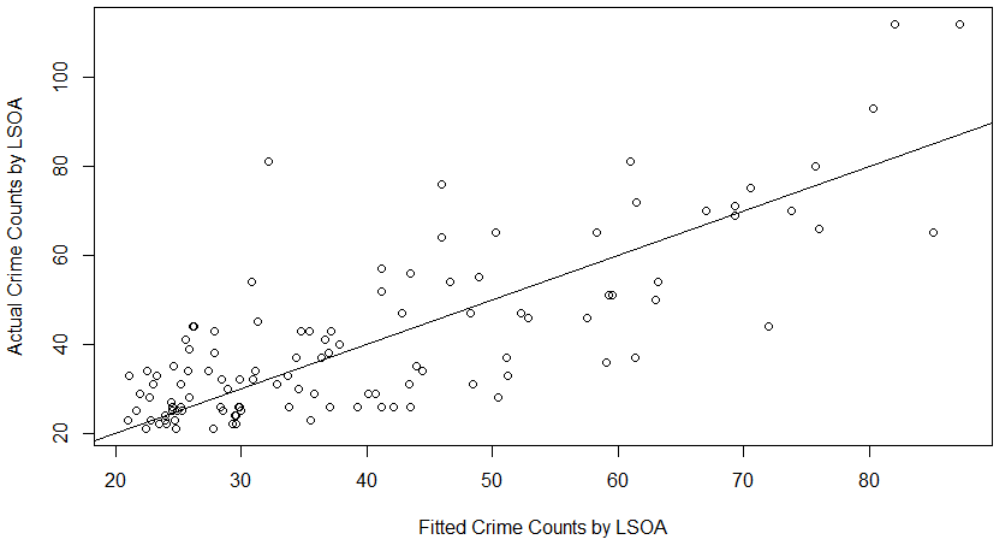

Figure 5.14: Fitted vs. Actual Scatterplot, Train on 2013 to predict on 2014

A larger degree of scatter in the results is present than in the case when the

location of the crime was considered to be the location in which the crime was committed. For the most part, however, a clear pattern in the fitted and actual results is observed and there appears to be a reasonable degree of positive correlation between the fitted and actual results - this is backed up by the correlation coefficient between the fitted and actual results, which is 0.7795.

Table 5.16: Aggregate Crime Count Predictions from WIMD, Location of Offence, Variable Importances

Variable	Importance
Employment_Benefits_Percentage	0.5659
Education_KS4L2_Percentage	0.0889
Education_Absentee_Percentage	0.0763
Health_DeathRate_per100000	0.0555
Education_KS2_Points	0.0468
Education_KS4_Points	0.0365
Income_Deprivation_Percentage	0.0352
Education_NotInHE_Percentage	0.0339
Health_Cancer_per100000	0.0296
Employment_LTSick_per100000	0.0111
Health_LowBW_Percentage	0.0104
Education_NoQuals_Percentage	0.0099

Here, the propensity to commit crime by offenders resident within a given LSOA is most heavily affected by the percentage of the population within that LSOA who are on Employment Benefits.

The mean crime counts by LSOA of the 11 clusters produced by the Affinity Propagation algorithm are as follows:

Table 5.17: Mean No. of Crimes per LSOA

Cluster	Mean No. of Crimes per LSOA
1	34.66667
2	35.00000
3	54.30000
4	33.14286
5	47.00000
6	77.25000
7	43.22222
8	27.45455
9	29.63158
10	65.80000
11	96.33333

Random Forests - Reoffence Predictions

After removing the raw WIMD variables and replacing these with the created Area Classification scores, the Random Forests model as built in Chapter 2 was re-tested and compared the results of each of these models. The comparative metrics are detailed below - please see Chapter 2 for the definitions.

Table 5.18: Random Forest Evaluation Metrics. Parameters: No. Trees = 500

Metric	Raw Features	Clustering Added
Validation Accuracy	0.8204	0.8177
Training Set AUC Score	0.9924	0.9536
Training Set Importance	0.9847	0.9071
Test Set AUC Score	0.8084	0.7999
Test Set Importance	0.6169	0.5998
First-Time Offenders Test Set AUC Score	0.7062	0.7114
First-Time Offenders Test Set Importance	0.4124	0.4228

It appears that making use of these area scores does not result in any significant increase or decrease in the model's performance on unseen data. While the model appears to overfit less to the training data and produce slightly better predictions on first-time offenders, the fit to the validation and unseen sets made up of all offenders is slightly worse.

Therefore, it is uncertain as to which of these models is preferable - this will depend on whether the analysts within Dyfed-Powys police would rather make use of a more complicated modelling process with fewer factors or a simpler process with a larger number of factors. In this case, when considering the issue of model complexity, the replacement of raw features with clustering can be considered to both increase and decrease the level of complexity in the model. While it does increase the level of complexity in terms of the number of models and steps required to produce a prediction of an individual's reoffence, it dramatically decreases the number of variables required as model input.

5.2.4 Conclusion, Limitations and Further Results

Conclusions

Firstly, regarding the results produced by the LSTM in Section 5.1, it is evident that LSTMs are not appropriate for the sequence prediction task intended. While there

could be many reasons for these issues, including the size of the dataset, the overall length of each time sequence and the issues pertaining to several missing location data records within the dataset, it is evident that making use of this method to predict the times to next crime within a given location in the Dyfed-Powys area is unlikely to be effective. While there may be several other effective potential applications of LSTM networks for Dyfed-Powys data, these are beyond the scope of this thesis and likely beyond the scope of what the police force are currently able to bring into use at this point in time.

From the second investigation within this dataset, the Area Classification score generated for the original dataset appears to be quite effective as a replacement for the various WIMD variables brought in from the external data. While there are some systematic variations brought about by year on year increases and decreases in the overall level of crime within Dyfed-Powys as a whole, the XGBoost algorithm appears to be able to capture the overall crime count patterns between the various LSOAs in Dyfed-Powys. Therefore, it is evident that the XGBoost algorithm produces a suitable estimate of the crime counts in the LSOAs within Dyfed-Powys and from the results of the Random Forests algorithm, it is also evident that the clusters produced by the Affinity Propagation algorithm are also suitable for use in predicting reoffender behaviour.

When looking at both the original "best-case" algorithm (as described in Chapter 2) and the "best-case" algorithm containing the Area Classification score, the Random Forests algorithm's performance on the dataset in either case is comparable - while making use of an Area Classification score appears to result in slightly better performance on First-Time Offender data and less overfitting to the training data, making use of the raw data as-is would make more sense if the police wanted improved validation and test data overall. However, in either case, the difference is negligible.

Therefore, the choice as to whether or not an Area Classification score should be used will simply be down to practical considerations. While it would be more elegant to make use of a stacked model as the number of variables would be reduced, it is likely to be simpler for analysts within Dyfed-Powys police to make use of a

single model that directly uses a set of raw variables to produce a series of predictions. As such, it has been chosen not to incorporate the Area Classification score into the final model - maintaining two models is likely to be far more complex than maintaining one and since there is no great decrease in model complexity or increase in model performance resulting from the creation of the Area Classification score, it is unlikely to be worth maintaining both models in a live context.

Limitations

As previously stated, within the original dataset given to us by Dyfed-Powys, many of the crimes have missing or inconsistent locations. This makes it difficult to know whether or not any algorithm heavily based on location variables is going to be viable on a live dataset - it is entirely possible that systematic errors in the data process that originally created the database will have made some locations appear to be low-crime when they are not, or comparatively high-crime when they are not. While to the best of our knowledge, the clustering that has been produced on our original dataset will also apply to the live data, it is uncertain if this will be the case due to the location data inconsistencies between the two datasets. This is a major limitation for both of the algorithms, especially the LSTM due to the sequential nature of the input required.

While LSTM networks may have many potential applications for Dyfed-Powys police at a future date, it is likely that the computing power required to run LSTM networks will severely limit the effectiveness of this algorithm in a live environment. Moreover, the datasets required to be input into LSTM networks are likely to be large and unwieldy - these would need to be stored and processed on a server rather than processed via csv test files, as was originally the case here.

Further Research

It is possible that by including all aggregate information from all Welsh police forces, it may be possible to obtain better predictions of reoffence counts by offender location - when those offences committed by out of area individuals cannot easily be taken into account, it can be difficult to estimate counts by offender location.

It is also possible that by including further Welsh government data, which is also keyed on LSOA, it may be possible to find further factors that are more consistently predictive of the aggregate reoffence count. While a number of deprivation factors have already been included in this analysis, the deprivation within an area is far from the only general factor that could affect the general propensity of its population to commit offences. This data is publicly available and as such, can easily be obtained by Dyfed-Powys police at a future date.

Creating alternative location partitions might also help in finding a more stable reoffence count pattern. The LSTM predictions may, in fact, be improved by separating offences by LSOA instead of by town and there is also potential that the town-centric location partition method may be effective with regards to an Area Classification. While there are significant advantages to making use of the LSOA partitions (including equal populations per area and data consistency), it may in fact make more sense to use alternative partitions in the future. For example, when dealing with LSOAs that are very close to the border of the Dyfed-Powys area, it may make more sense to group several neighbouring LSOAs together and take the aggregate average of their crimes. For those that are further away (for example, in Birmingham or Bristol, two areas that appeared to make several appearances in the dataset), it may be appropriate to simply designate the area as a city, though this may cause issues with weighting the location by population at a future date.

Chapter 6

Conclusion

6.1 Summary of Results

6.1.1 Models for Reoffence Prediction

When considering the three binary classification models that were tested on this dataset within Chapter 2, it was clear that the best performance on unseen data was achieved by the Random Forests algorithm, both in the case in which the output of the model was a binary classification and the case in which the output was a probability of reoffence. While it is possible that comparable performance may be generated by altering the formats of the input variables or further tuning the parameters of the XGBoost algorithm, Random Forests is currently our best performing algorithm in terms of predictive performance. As shown in the live test results, this algorithm will need to be deployed with care and the success of its predictions will likely depend (at least in part) on the way in which the input dataset is engineered. With due consideration put in place for the limitations of the Random Forests algorithm, however, there is no reason why this algorithm cannot be used to accurately predict reoffender behaviour in the Dyfed-Powys area. Therefore, for its benefits in terms of simplicity as well as predictive accuracy, it has been chosen to deploy the Random Forests algorithm as outlined in Chapter 2 in Dyfed-Powys - as stated in the Introduction, it will underpin a new diversionary scheme within the force, which has recently been allocated funding within the budget.

Based on the comparison between the predictive accuracy of the Probability Forest and the Classification Forest,it can be concluded that the Probability Forest is likely

to be the better option. Moreover, with an output that is not quite as black and white as a yes/no classification, using a Probability Forest will give a greater degree of agency in the way that these results are utilised, giving Dyfed-Powys police the option of taking these probabilities and forming their own classification system (e.g. High/Medium/Low Risk), should they wish to do so. The use of Principal Components Analysis to reduce the dimensionality of the input dataset, while investigated in Chapter 2, has been determined to be largely unnecessary in this context; the potential increase in predictive accuracy afforded by this model did not offset the loss of information to the police in its implementation. As such, the version of the model that has been deployed in Dyfed-Powys does not contain a PCA analysis.

In Chapter 5, the augmentation of the Random Forests model with an Area Classification was discussed. This was created using both XGBoost for crime count predictions and Affinity Propagation for clustering the predicted crime counts. While the Area Classification model has not been deployed for use in Dyfed-Powys at this current time, it is clear that making use of this model stack does have some benefits, both in terms of reducing the dimensionality in the model and increasing the amount of information that can be obtained from it in terms of general offence occurrence. Although this model would be a more elegant solution in the longer term, once the analysts within Dyfed-Powys have got to grips with the first deployment of the classification model.

In Chapter 3, the Random Forests classification and probability models were further developed into a survival model, allowing for more granular predictions of reoffence likelihood to be made for each offender. This survival model, which can handle the large number of diverse risk factors necessary for prediction, produces a convincing set of results on both cold start and non-cold start data, meaning that this model can certainly be said to be suitable for the purposes of predicting offender survival. Specifically, it was found that splitting the individual trees using the maxstat splitrule was the most effective method of producing predictions of offender survival, reducing the runtime of the algorithm while increasing the accuracy of its predictions.

In all contexts in which the Random Forests model was used, the list of variable

importances generated by the model will be extremely important in a real-world setting. However, it is uncertain how exactly each of these individual factors will alter the survival of an offender - due to the model's non-linear nature, the variable importance measure created by the Random Forests model can give no indication of how its value will affect the predicted reoffence outcome. While this may be an issue for officers more familiar with the OGRS system (which, due to the linear nature of the algorithm employed, will be able to give such an indication), the ways in which the values of each individual factor affect the reoffence outcome should become apparent through sufficient testing and application of police intuition.

6.1.2 Models for Spatio-Temporal Crime Prediction

Moving on to the task of predicting the location in which an offence is likely to occur, the memory-based collaborative filtering recommender system discussed in Chapter 4 yielded some interesting results. Firstly, and most instructively, a pattern of similarities was found between the locations in Dyfed-Powys that held almost constant over time. In the case of the Jaccard Similarities of Bags as generated from a count matrix of offences, this pattern was sufficiently consistent month on month to consider it to be a "constant pattern". In the case of the cosine similarities as generated from a TF-IDF vectorisation of that count matrix, however, this pattern was only found to be sufficiently consistent year on year. The distinction between these is important and will need to be considered in their deployment within Dyfed-Powys - currently, in order to exploit the monthly as well as yearly consistency, the Jaccard Similarity of Bags has been selected for deployment in Dyfed-Powys. If this pattern is found to be inconsistent over several months in a live context, however, it would be recommended for Dyfed-Powys police to consider testing both methods to see which produces the most useful measures of location similarity.

Exploiting the patterns found in this matrix through the production of location clusters proved more difficult, with neither of the Affinity Propagation or Spectral Clustering algorithms yielding a particularly tight set of clusters. This could be due to many factors, including the poor reporting of location data or the measure of similarity used not being the most appropriate for this dataset, but it may also be that this crime count data is simply not suited to produce well-separated sets of

Gaussian clusters. In fact, a large degree of overlap between clusters is likely not too much of an issue in this context, given that the temporal distribution of crimes in most urban areas seem to be very similar to one another, while in most rural areas they are very different to one another. Strangely, however, there does not seem to be any direct link between the clusters generated by the algorithm and the rural-urban index of the locations themselves, nor do the generated clusters seem to be best explained by this rural-urban index. This will likely merit further investigation from Dyfed-Powys police.

While the LSTM Neural Networks in Chapter 5 did not yield any appropriate results, it is possible that approaching the problem from a different angle (as suggested in the conclusion of Chapter 5, Section 1) may yield preferable results to the original time to event prediction idea. While this method may well take more time and effort to set in motion, taking this approach may provide better predictions in the long term. However, given the results suggested in Chapter 5, it is possible that another, perhaps probability distribution based, method of producing these predictions should be considered. This research is out of scope for this thesis, but may form the basis of future research in locations with higher crime rates.

6.2 Future Recommendations for Dyfed-Powys Police

In order to extend and improve upon the methods analysed in this thesis, it is necessary that recommendations are made as to how and where Dyfed-Powys police can improve these methods to make better use of the concepts outlined in this thesis in a practical policing context. These improvements, of which many were previously mentioned at the ends of each individual chapter, will be summarised here, beginning with the possible improvements to be made to the reoffence research conducted in Chapters 2, 3 and 5, before moving on to the location-based research conducted in Chapters 4 and 5.

6.2.1 Models for Reoffence Prediction

The first recommendation that I would make to Dyfed-Powys police, or any other police force intent on following the methodology discussed within this thesis, would be to consider adjusting the 3-year time limit in the reoffence classification case to suit the needs of the officers involved. While the 3-year time limit does provide a good measure of a near-eventual probability of reoffence, it is possible that within a policing context, it may be more appropriate to look at the probability of an offender's reoffence within a lower or higher time limit. As such, this limit should be adjusted to meet the needs of Dyfed-Powys, or in fact any other police force that intends to use the recommended method to predict the likelihood of reoffence for the offenders within their database.

Within the context of the reoffence classification model, it should also be discussed as to whether or not this time limit is to include possible prison time (as is the current default), or to exclude prison time from these limits, thereby setting the offender's start date to be their release from prison if they have been incarcerated as the result of that offence. Similarly, it should also be discussed as to whether or not an offender's period of incarceration should be included or excluded from the predictions of survival. As previously mentioned in Chapter 2, due to the unavailability of this data, it is impossible for this thesis to assess the effects of including or excluding this data from the predictions made. As such, this particular task must be left to further research.

In the case of both the reoffence classification and survival models, the treatment of probation periods should also be considered carefully, as if an offender is subject to probation, it may well result in an alteration of their behaviour with regards to their propensity to reoffence. Including the presence of a probation period could be done in several different ways, with the most obvious being a simple indicator variable that indicates whether an offender was put under probation or an indicator of the probation period that the individual offender was put under. By including the influence of probation periods into the reoffence likelihood predictions, it is possible to see the effect of the treatment of offenders (at least in terms of their monitoring following an offence) on their likelihood of survival in different time periods, as well

as their eventual probability of reoffence.

As the dataset grows over time, it may be possible that a point is reached at which predictions generated by a Neural Network become viable and perhaps even more accurate than those produced by Random Forests. Should, for example, Dyfed-Powys want to make use of larger national offence datasets to fit future models, it may be worth them testing the Neural Network approach at a future date, as the more data a Neural Network can train on, the more its performance will improve. It is possible that this may also be the case for the survival model, though as yet it is uncertain whether or not such a model would be appropriate for the prediction of offender survival.

Finally, the most simple way in which Dyfed-Powys police can make improvements to the performance of these models is through the addition of further data. As indicated by many past studies, including further demographic information on the offender could significantly boost the accuracy of the predictions. This sort of data could prove extremely helpful in gaining a deeper understanding of the factors that affect the probability of an offence leading to a further offence, or an offender's survival following an offence. This could take two forms:

1. Objective data, such as the offender's marital status, employment status or medical history.

2. Subjective data, such as an officer's opinion on the likely causes of the reoffence.

Due to recent changes in data protection regulations via GDPR, objective information would need to be treated carefully. This information may well need to be completely depersonalised and stored on aggregate if it is not volunteered (which in most cases, it is not). The research completed in Chapter 5 on the production of an Area Classification Score may therefore have wider benefits once a larger amount of objective data is collected, particularly if a large amount of information is collected on aggregate by LSOA. Subjective data, however, would likely not be bound by such regulations; as these are subjective opinions on the part of the officers involved in

the case, they would not be considered sensitive personal information. As such, this information (which is already collected in the form of "Alcohol-Related" and "Drug-Related" statistics) could be collected and used freely at the police's discretion.

Although collecting data in either of these forms would require a small amount of extra investment on Dyfed-Powys' part to make the collection of this sort of data a routine part of their investigations on new offenders, it may well be worth collecting this extra data from offenders to both understand and have the ability to react to the individual social factors that affect reoffending. The simplest way for Dyfed-Powys police to begin collecting this data would be to include more subjective statistics in the current data collection form; this would have the additional benefit of transferring officer knowledge upstream, where it can be quantified and made use of in a more structured manner.

6.2.2 Models for Spatio-Temporal Offence Prediction

The most important recommendation that I must make to Dyfed-Powys police, should they wish to use location-based prediction methods in the future, is to improve the automatic creation and entry of location data into the police database. For postcode data, this could be as simple as introducing an automatic postcode check into the entry form so that invalid UK postcodes cannot be entered into the system. As the geocode data is based on this postcode data, this will also solve the problem of incorrect or missing geocodes within the dataset. Incorrect town data, which is sometimes present due to spelling errors, could be solved by simply introducing a drop-down menu into the data entry system, as it is unlikely that any new towns or villages will be created in Dyfed-Powys in the near future. While this will not alter the data already within the database and it does appear that significant improvements have been made since the original dataset was delivered for testing, such improvements to the entry system will improve the accuracy of future data and allow Dyfed-Powys police to validate the results of Chapter 4's clustering algorithm in the long term.

Although the two methods of calculating similarity were chosen as the best and most appropriate methods of producing a similarity matrix for this particular dataset, it is

still possible that there may be a better method of defining the similarities between locations. The simplest way to alter these definitions of similarity is to alter the boundaries that separate the locations in Dyfed-Powys. It is possible, for example, that locations within Dyfed-Powys are better partitioned by town than postcode sector as in Chapter 5, or that they are better partitioned by a series of radii generated by a series of Geocode centres. Such optimisations are beyond the scope of this thesis, but may provide long-term improvements to the recommender system as generated for this dataset.

Finally, to improve the success of an LSTM model, or offer an alternative perspective on the factors that are likely to affect an offence, it is possible that more location-based factors can be added. The most simple addition that could be made is the addition of meterological data, which likely has a great impact on the occurrence of crimes within a location (and may, in fact, have a lot more to do with the occurrence of crimes than a past pattern of offences within a certain area) due to the weather's impact on criminal movements within a location. It is intuitive to assume that, for example, the likelihood of many offences occurring may drop significantly when the weather is poor due to a decrease in criminal movements during those periods of time. An RNN would also have the opportunity to detect the effect of suddenly changing weather; for example, the effect of good weather on crime may be less potent if there are a number of consecutive days with good weather, with the greatest boost to activity occurring on the first day of the improved weather. While this data is as yet unavailable for these purposes, should Dyfed-Powys police be able to acquire this data at a later date, it would be prudent to consider testing its effectiveness as a predictor of criminal movements within a location.

In conclusion, while I believe that sufficient research has been completed to deploy a series of finished models to Dyfed-Powys police, it is clear that far more research and development can be done in this area. For the police, obtaining further data via improved data collection methods or the use of further external datasets could provide a simple method of improving what is already there. Once this data is in place, it is likely that significant improvements can be made to the models deployed within Dyfed-Powys, though care must be taken not to overcomplicate the models such that they become unusable. The biggest challenge for Dyfed-Powys, therefore,

is to determine exactly what data they would ideally like to collect, how they wish to collect it and how it should be stored. Once they have decided how this will be done, they will need to decide how this data should be used - at this point, they may return to the results of this initial research to form guidelines as to how the initial models should be refreshed. As their data collection methods improve and the volume of data collected increases, it is extremely important that Dyfed-Powys consistently improve on the models that make use of this data. By constructing a process by which these models are iteratively improved over time, Dyfed-Powys can help to ensure the safety of the communities it protects as the world moves towards an increasingly data-driven future. As such, my final (and most important) recommendation from this research is that Dyfed-Powys invests as heavily as it can into the process of data collection within the force - with many police forces both nationally and internationally now moving into the sphere of predictive policing, this is how Dyfed-Powys can best make sure that maintains its position at the frontier of this fast-developing field.

Bibliography

[1] https://tex.stackexchange.com/questions/132444/diagram-of-an-artificial-neural-network. Access Date January 2018.

[2] Olusola Adeniyi Abidogun. Data mining, fraud detection and mobile telecommunications: Call pattern analysis with unsupervised neural networks. Master's thesis, University of the Western Cape School of Computer Science, August 2005.

[3] Andre Altmann, Laura Tolosi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *BMC Bioinformatics*, 26(10):1340–1347, April 2010.

[4] Joaeran Beel, Bella Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, November 2016.

[5] Brent B. Benda. Survival analysis of criminal recidivism of boot camp graduates using elements from general and developmental explanatory models. *International journal of offender therapy and comparative criminology*, 47(1):89–110, February 2003.

[6] Brent B. Benda. Gender differences in life-course theory of recidivism: A survival analysis. *International journal of offender therapy and comparative criminology*, 49(3):325–342, June 2005.

[7] Yoshua Benigo. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[8] James Bennett and Stan Lanning. The Netflix prize. *KDDCup '07*, August 12th 2007.

[9] Richard Berk, Lawrence Sherman, Geoffrey Barnes, Ellen Kurtz, and Lindsay Ahlman. Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning. *Statistics in Society Series A*, 172(1):191–211, January 2009.

[10] Richard A. Berk. An impact assessment of machine learning risk forecasts on parole board decisions and recidivism. *Journal of Experimental Criminology*, 13(2), June 2017.

[11] Richard A. Berk and Justin Bleich. Statistical procedures for forecasting criminal behavior. *Criminology and Public Policy*, 12(3):513–544, August 2013.

[12] Richard A. Berk, Susan B. Sorenson, and Geoffery Barnes. Forecasting domestic violence: A machine learning approach to help inform arraignment decisions. *Journal of Empirical Legal Studies*, 13(1):94–115, March 2016.

[13] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[14] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[15] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[16] Laurent Candillier, Frank Meyer, and Francoise Fessant. Designing specific weighted similarity measures to improve collaborative filtering systems. In *International Conference on Data Mining 2008: Advances in Data Mining, Medical Applications, E-Commerce, Marketing and Theoretical Aspects (ICDM '08)*, pages 242–255, 2008.

[17] Jonathan Caulkins, Jacqueline Cohen, Wilpen Gorr, and Jifa Wei. Predicting criminal recidivism: A comparison of neural network models with statistical methods. *Journal of Criminal Justice*, 24(3):227–240, 1996.

[18] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016.

[19] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. xgboost: Extreme gradient boosting, January 2017. https://cran.r-project.org/web/packages/xgboost/xgboost.pdf.

[20] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Benigo. Learning phrase representations using rnn encoder decoder for statistical machine translation. *arXivpreprintarXiv* : 1406.1078, 2014.

[21] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, June 2017.

[22] David Cox. The regression analysis of binary sequences (with discussion). *Journal of the Royal Statistical Society Series B (Methodological)*, 20(2):215–242, 1958.

[23] David R. Cox. Regression models and life tables. *Journal of the Royal Statistical Society, Series B (Methodological)*, 34(2):187–220, 1972.

[24] Abhinandan S. Das, Mayur Datar, Shyam Rajaram, and Ashutosh Garg. Google news personalisation: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pages 271–280, 2007.

[25] 2011 rural urban classification - methodology, July 2016. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239477/RUC11methodologypaperaug_28_Aug.pdf.

[26] Timothy Dozat. Incorporating nesterov momentum into adam, 2016.

[27] Julia J. Dressel. Accuracy and racial biases of recidivism prediction instruments, May 2017. Dartmouth Computer Science Technical Report TR2017-822.

[28] Grant Duwe and Pamela J Freske. Using logistic regression modeling to predict sexual recidivism: The minnesota sex offender screening tool-3 (mnsost-3). *Sex Abuse*, 24(4):350–377, August 2012.

[29] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(number):972–976, 2007.

[30] Ariya Fujita and Shigeru Shimada. Forecast of criminal character by text mining of suspicious person behavior report. http://id.nii.ac.jp/1004/00000143/.

[31] Edmund A. Gehan. A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52:203–223, 1965.

[32] Felix A. Gers, Jurgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[33] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006.

[34] George Giaglis and George Lekakos. Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors. *Interacting with Computers*, 18(3):410–431, 2006.

[35] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992.

[36] Welsh Government. https://statswales.gov.wales/Catalogue. Welsh Government website containing detailed official data on Wales.

[37] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, and Orlando de Jesus. *Neural Network Design*. Martin T. Hagan, 2nd edition edition, September 2014. http://hagan.okstate.edu/NNDesign.pdf.

[38] Zachary Hamilton, Melanie-Angela Neuilly, Stephen Lee, and Robert Barnoski. Isolating modeling effects in offender risk assessment. *Journal of Experimental Criminology*, 11(2):299–318, June 2015.

[39] John R. Hepburn and Celesta A. Albonetti. Recidivism among drug offenders: A survival analysis of the effects of offender characteristics, type of offense, and two types of intervention. *Journal of Quantitative Criminology*, 10(2):159–179, June 1994.

[40] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, January 2004.

[41] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[42] Philip Howard, Brian Francis, Keith Soothill, and Les Humphreys. Ogrs 3: The revised offender group reconviction scale. Ministry of Justice Research Summary, July 2009. `https://core.ac.uk/download/pdf/1556521.pdf`.

[43] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008.

[44] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2 edition, 2002.

[45] Rafael Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the* 32^{nd} *International Conference on Machine Learning*, volume 37, 2015.

[46] Harrell FE Jr, Califf RM, Pryor DB, Lee KL, and Rosati RA. Evaluating the yield of medical tests. *JAMA*, 247(18):2543–2546, May 1982.

[47] Hyunzee Jung, Solveig Spjeldnes, and Hide Yamatani. Recidivism and survival time: Racial disparity among jail ex-inmates. *Social Work Research*, 34(3):181–189, September 2010.

[48] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[49] Merry Morash Kristy Holtfreter, Michael D. Reisig. Poverty, state capital, and recidivism among women offenders. *Criminology & Public Policy*, 3(2):185–208, March 2004.

[50] Berthold Lausen and Martin Schumacher. Maximally selected rank statistics. *Biometrics*, 48:73–85, March 1992.

[51] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning, October 2015.

[52] Yuan Y. Liu, Min Yang, Malcom Ramsay, Xiao S. Li, and Jeremy W. Coid. A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending. *Journal of Quantitative Criminology*, 27(4):547–573, December 2011.

[53] Jorge M. Lobo, Alberto Jiminez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17:145–151, 2008.

[54] David Lovell, Gregg J. Gagliardi, and Paul D. Peterson. Recidivism and use of services among persons with mental illness after release from prison. *Psychiatric Services*, 53(10):1290–1296, October 2002.

[55] James D. Malley, J. Kruppa, Abhijit Dasgupta, Karen G. Malley, and Andreas Ziegler. Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, January 2012.

[56] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*, chapter Scoring, term weighting and the vector space model. Cambridge University Press, April 2009.

[57] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI '03)*, pages 263–266. ACM, ACM Press, 2003.

[58] Melanie-Angela Neuilly, Kirsten M. Zgoba, George E. Tita, and Stephen S. Lee. Predicting recidivism in homicide offenders using classification tree analysis. *Homicide Studies*, 15(2):154–176, May 2011.

[59] Andreas M. Olligschlaeger. Artificial neural networks and crime mapping, 1998.

[60] Pew Center on the States. State of recidivism: the revolving door of america's prisons. The Pew Charitable Trusts Washington DC, April 2011.

[61] Turgut Ozkan. *Predicting Recidivism through Machine Learning.* Doctor of philosophy in criminology, University of Texas at Dallas, May 2017.

[62] Susan W. Palocsay, Ping Wang, and Robert G. Brookshire. Predicting criminal recidivism using neural networks. *Socio-Economic Planning Sciences*, 34(4):271–284, December 2000.

[63] Michael J. Pazzani and Daniel Billsus. *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2007.

[64] K. Pearson. On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, 2(6):559–572, 1901.

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhoffer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perror, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[66] Isabelle Percy, Alexander Balinsky, Helen Balinsky, and Steve Simske. Text mining and recommender systems for predictive policing. In *Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018, Halifax, NS, Canada*, ACM 2018, pages 15:1–15:4, August 2018.

[67] Marlon O Pflueger, Irina Franke, Marc Graf, and Henning Hachtel. Predicting general criminal recidivism in mentally disordered offenders using a random forest approach. *BMC Psychiatry*, pages 15–62, March 2015.

[68] Marlon O Pflueger, Irina Franke, Marc Graf, and Henning Hachtel. Predicting general criminal recidivism in mentally disordered offenders using a random forest approach. *BMC Psychiatry*, 15(62), March 2015.

[69] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *RecSys '09 Proceedings of the third ACM conference on Recommender systems*, pages 385–388, October 2009.

[70] Welsh Government Statistics and Research. Welsh index of multiple deprivation. Technical report, Welsh Government, `http://gov.wales/statistics-and-research/welsh-index-multiple-deprivation/`, 2014.

[71] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.

[72] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[73] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1996.

[74] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00)*, pages 158–167, 2000.

[75] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *ECommerce '99*, 1999.

[76] Matthias Schmid, Marvin N. Wright, and Andreas Ziegler. On the use of harrell's c for clinical risk prediction via random survival forests. *Expert Systems with Applications*, 63:450–459, November 2016.

[77] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, pages 210–217, 1995.

[78] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08 Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266, October 2008.

[79] Lawrence Sherman, Peter William Neyroud, and Eleanor Neyroud. The cambridge crime harm index: Measuring total harm from crime based on sentencing guidelines. *Policing*, 10(3):171–183, April 2016.

[80] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[81] C.O.S Sorzano, J. Vargas, and A. Pascual-Montano. A survey of dimensionality reduction techniques, 2014.

[82] Andrew L. Spivak and Kelly R. Damphouse. Who returns to prison? a Survival analysis of recidivism among adult offenders released in Oklahoma, 1985 – 2004. *Justice Research and Policy*, 8(2):57–88, December 2006.

[83] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, June 2014.

[84] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, pages 8–25, 2007.

[85] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):1–25, January 2007.

[86] Luis Teran and Andreas Meier. A fuzzy recommender system for eelections. In *Proceeds of the Electronic Government and the Information Systems Perspective (EGOVIS 2010)*, 2010.

[87] Nikolas Tolenaar and Peter van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Statistics in Society A*, 176(2):565–584, February 2013.

[88] Stephen J. Tripodi, Johnny S. Kim, and Kimberly Bender. Is employment associated with reduced recidivism? the complex relationship between employment and crime. *International Journal of Offender Therapy and Comparative Criminology*, 54(5):706–720, July 2009.

[89] Ulrike von Luxburg. A tutorial on spectral clustering. *Stat Comput*, 17:395–416, 2007.

[90] Emmanoiul Vozalis and Konstantinos G. Margaritis. Analysis of recommender systems' algorithms. In *Proceeds of the Sixth Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2003)*, pages 263–266. ACM, ACM Press, 2003.

[91] Mingjun Wang and Matthew S. Gerber. Using twitter for next-place prediction, with an application to crime prediction. In *2015 IEEE Symposium Series on Computational Intelligence*, December 2015.

[92] Marvin N. Wright, Theresa Dankowski, and Andreas Ziegler. Unbiased split variable selection for random survival forests using maximally selected rank statistics. *Statistics in Medicine*, 36(8):1272–1284, January 2017.

[93] Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1), March 2017.

[94] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*, 2003.

[95] Yan Zhou and John J. McArdle. Rationale and applications of survival tree and survival ensemble methods. *Psychometrika*, 80(3):811–833, September 2015.

Appendix A

Dyfed-Powys Dataset: Independent Variables Used in Analysis

Variable	Description	Possible Values	Variable Type
DayOfWeek	The weekday on which the offence was committed	Valid Day of Week	Categorical
Month	The month in which the offence was committed	Valid Month of Year	Categorical
AlcoholRelated	Is the offence alcohol related?	True, False	Binary
BurgValue	Value of goods stolen by offender	0+	Integer
CrimDamValue	Value of criminal damage committed by offender	0+	Integer
DrugRelated	Is the offence drug related?	True, False	Binary
MDAClass	UK Government classification of drug seized	Class A (A), Class B (B), Class C (C), Unknown (U), No Drug (0)	Categorical

MultipleOffences	Whether or not the offender was arrested for more than one offence on that day	True, False	Binary
OffenceCat	Type of offence	See Chapter 2	Categorical
OffenceRacist	Is the offence racially motivated?	True, False	Binary
OffenceTriable	Severity of offence, as considered in court	Summary, Either Way, Indictable	Categorical
Outcome	Offence follow-up	See Chapter 2	Categorical
RelatedToVictim	How was the offender related to their victim?	Family, Other, Unknown	Categorical
HaversineDist	Haversine Distance between offender's home location and the location of their crime	0+	Float
AgeCommitted	Age of offender at time of offence	0+	Integer
OffenderSex	Sex of offender	Male (M), Female (F), Unknown (U)	Categorical
VictimSex	Sex of victim	Male (M), Female (F), Unknown (U), No Victim (0)	Categorical
VictimAge	Age of victim	0+	Integer
NoPreviousArrests	Total number of previous arrests	0+	Integer
Fine	No. of fine payoff days assigned to crime based on Cambridge Crime Harm Index[79]	See Chapter 2	Integer

Sentence	No. of sentencing days assigned to crime based on Cambridge Crime Harm Index[79]	See Chapter 2	Integer
PrevFine	No. of fine payoff days assigned to offender's previous crimes based on Cambridge Crime Harm Index[79]	See Chapter 2	Integer
PrevSentence	No. of sentencing days assigned to offender's previous crimes based on Cambridge Crime Harm Index[79]	See Chapter 2	Integer
PrevBurg	No. of burglaries previously committed by the offender	0+	Integer
PrevDrug	No. of drug-related offences previously committed by the offender	0+	Integer
PrevIndict	No. of indictable offences previously committed by the offender	0+	Integer
PrevSex	No. of sex offences previously committed by the offender	0+	Integer
PrevViolence	No. of violent offences previously committed by the offender	0+	Integer
PrevWeapons	No. of weapon-related offences previously committed by the offender	0+	Integer
Crime Income Perc	Percentage of low-income households within the area in which the crime was committed	0+	Float

Off Income Perc	Percentage of low-income households within the area in which the offender is resident	0+	Float
Crime EmpBenefits Perc	Percentage of households in receipt of employment benefits within the area in which the crime was committed	0+	Float
Off EmpBenefits Perc	Percentage of households in receipt of employment benefits within the area in which the offender is resident	0+	Float
Crime EmpLTSick per100000	Number of people per 100000 who are considered to be long-term sick within the area in which the crime was committed	0+	Float
Off EmpLTSick per100000	Number of people per 100000 who are considered to be long-term sick within the area in which the offender is resident	0+	Float
Crime HealthDR per100000	Death rate per 100000 people within the area in which the crime was committed	0+	Float
Off HealthDR per100000	Death rate per 100000 people within the area in which the offender is resident	0+	Float
Crime HealthCancer per100000	Cancer rate per 100000 people within the area in which the crime was committed	0+	Float

Off HealthCancer per100000	Cancer rate per 100000 people within the area in which the offender is resident	0+	Float
Crime HealthLBW Perc	Percentage of low birth weights recorded within the area in which the crime was committed	0+	Float
Off HealthLBW Perc	Percentage of low birth weights recorded within the area in which the offender is resident	0+	Float
Crime EducKS2 Pts	Average points earned at Key Stage 2 by schoolchildren within the area in which the crime was committed	0+	Float
Off EducKS2 Pts	Average points earned at Key Stage 2 by schoolchildren within the area in which the offender is resident	0+	Float
Crime EducKS4 Pts	Average points earned at Key Stage 4 by schoolchildren within the area in which the crime was committed	0+	Float
Off EducKS4 Pts	Average points earned at Key Stage 4 by schoolchildren within the area in which the offender is resident	0+	Float

Crime EducAbs Perc	Percentage of unauthorised absences by schoolchildren within the area in which the crime was committed	0+	Float
Off EducAbs Perc	Percentage of unauthorised absences by schoolchildren within the area in which the offender is resident	0+	Float
Crime EducKS4L2 Perc	Percentage of schoolchildren achieving level 2 qualifications within the area in which the crime was committed	0+	Float
Off EducKS4L2 Perc	Percentage of schoolchildren achieving level 2 qualifications within the area in which the offender is resident	0+	Float
Crime EducNotInHE Perc	Percentage of schoolchildren who did not progress to higher education within the area in which the crime was committed	0+	Float
Off EducNotInHE Perc	Percentage of schoolchildren who did not progress to higher education within the area in which the offender is resident	0+	Float
Crime EducNoQuals Perc	Percentage of schoolchildren leaving school with no qualifications within the area in which the crime was committed	0+	Float

Off EducNoQuals Perc	Percentage of schoolchildren leaving school with no qualifications within the area in which the offender is resident	0+	Float
Crime Burglary per100	Burglary rate per 100 people within the area in which the crime was committed	0+	Float
Off Burglary per100	Burglary rate per 100 people within the area in which the offender is resident	0+	Float
Crime Violence per100	Violent crime rate per 100 people within the area in which the crime was committed	0+	Float
Off Violence per100	Violent crime rate per 100 people within the area in which the offender is resident	0+	Float
Crime Theft per100	Theft rate per 100 people within the area in which the crime was committed	0+	Float
Off Theft per100	Theft rate per 100 people within the area in which the offender is resident	0+	Float
Crime CrimDam per100	Criminal damage rate per 100 people within the area in which the crime was committed	0+	Float
Off CrimDam per100	Criminal damage rate per 100 people within the area in which the offender is resident	0+	Float

Crime Fire per100	Fire rate per 100 people within the area in which the crime was committed	0+	Float
Off Fire per100	Fire rate per 100 people within the area in which the offender is resident	0+	Float
Crime ASB per100	Anti-social behaviour rate per 100 people within the area in which the crime was committed	0+	Float
Off ASB per100	Anti-social behaviour rate per 100 people within the area in which the offender is resident	0+	Float
Crime UR01IND	Urban/Rural classification of the area in which the crime was committed	See Chapter 2	Categorical
Off UR01IND	Urban/Rural classification of the area in which the offender is resident	See Chapter 2	Categorical