A Novel Approach for Cross-Selling Insurance Products Using Positive Unlabelled Learning

Tomas Sidorowicz
School of Mathematics
Cardiff University
Cardiff, United Kingdom
SidorowiczTM@cardiff.ac.uk

Pedro Peres
ActiveQuote Ltd
Global Reach, Dunleavy Drive
Cardiff, United Kingdom
pedro.peres@activequote.com

Yuhua Li
School of Computer Science
Cardiff University
Cardiff, United Kingdom
LiY180@cardiff.ac.uk

Abstract—Successful cross-selling of products is a key goal of companies operating within the insurance industry. Choosing the right customer to approach for cross-purchase opportunities has a direct effect on both decreasing customer churn rate and increasing revenue. Unlike sales data of general products, insurance sales data typically contains only a few products (e.g., private medical insurance, life insurance, etc), it is highly imbalanced with a vast majority of customers with no cross-purchasing information, highly noisy due to varying purchase behaviour between different customers, and has no ground truth for knowing if the majority customers are truly non-cross-sell customers or they are missed opportunities. These data challenges render the building of machine learning models for accurately identifying potential cross-sell customers extremely difficult. This paper proposes a novel approach to solve this challenging problem of cross-sell customer identification using Positive Unlabelled (PU) learning in conjunction with advanced feature engineering on customer demographic data and unstructured customer question-response texts through topic modelling. We implement a bagging approach to iteratively learn the positive samples (the confirmed cross-sells) alongside random sub-samples of the unlabelled set. The proposed approach is extensively evaluated on real insurance data that has been newly collected from a leading insurance company for this study. Evaluation results demonstrate that our approach can successfully identify new potential opportunities for likely cross-sell customers.

Keywords—positive unlabelled learning, cross-sell, bagging classification, topic modelling, text similarity.

I. Introduction

Cross-sell is the selling of additional products or services to existing customers. This type of sale involves a certain risk-reward as successful implementation can increase customer longevity and reduce churn (i.e., the loss of customers), while unsuccessful implementation can weaken the customer relationship [1]. As such it is seen as a key strategic priority in many financial sector industries [2].

Modelling of past customer data can often be used to optimise the cross-sell recommendation process. This is rarely a straightforward process however.

One issue is that much of the data is in the form of question-response text. This needs to be processed in an interpretable and effective manner. For this we will look to topic modelling and a text similarity metrics.

More problematic issues arise when we look at class-specific properties. Within the insurance product dataset originally collected for this study, Private Medical Insurance (PMI) and Life insurance were rarely bought together—less than 0.5% of customers—making the data highly imbalanced. Further, there appears to be no negative class as customers that were approached and had turned down cross-purchase opportunities were not recorded. Finally, early modelling using a range of binary classifiers as well as state-of-the-art dimension reduction techniques such as Uniform Manifold and Projection (UMAP) we unable to separate between the single-sell/cross-sell customers; there appears to be no anomalous customer features or purchasing behaviour defining the cross-sell class.

Imbalanced data is a relatively straightforward problem to deal with, either on an algorithmic level (adjusting model weights or careful model selection) or a data sampling level (over/under sampling of data) [3]. Likewise, no negative labels are not a problem given enough anomalous data with which to train a one-class model [4]. Finally, learning algorithms can often identify anomalous high-dimensional clusters if given enough training data for each class. Any one of the problems alone can be dealt with in a straightforward fashion—but it is the combination of all three that means we must take a more specialised approach.

This approach takes the form of Positive Unlabelled (PU) learning. Framing the problem this way allows us to label the confirmed cross-sells as the positive class, and the remaining PMI and Life customers as an unlabelled (positive or negative) class.

PU learning naturally arises as a solution to a variety of important problems and thus has attracted a significant level of academic attention [5]. This approach—though previously explored—has not yet been applied to the cross-sell recommendation problem within the financial industry. Recommendation systems as a whole usually fall under the umbrella of either: content-based, collaborative filtering, knowledge-based or hybrid systems [6]. Such systems were not appropriate as a solution to our problem, as they rely on a higher diversity of items than the two insurance policies present in our dataset.

PU learning approaches can vary significantly. Bekker and Davis [5] identified four distinct techniques: two-step, biased learning, incorporation of the class prior and relational approaches. Our approach will focus on biased learning. We performed tests implementing the two-step technique, however we found that it showed inferior performance. The final two techniques rely on *knowledge of the label frequency* and the *open world assumption* respectively—assumptions that were unfeasible given our dataset.

We hypothesise that PU learning can be applied to simultaneously handle the non-ideal characteristics of our insurance customer data, and create a functional recommender system to identify likely potential cross-sell opportunities.

Specifically, using this approach we will only learn the positive examples, therefore the data imbalance is not a problem. The unlabelled set assumption means we do not need a negative class. The lack of anomality is dealt with by using

This project is funded by Innovate UK and Welsh Government. (KTP11894).

a PU bagging meta-estimator to iteratively perform large amounts training using the positive sample with random sub-samples of the unlabelled set. Finally, Out-Of-Bag (OOB) probability scores are obtained from the trained model. These represent likelihood of cross-sell and can be used to validate the model. Once validated the pre-trained model can be applied to new opportunities as part of a recommender system.

The key contributions of this paper are the following: (1) An analysis of PMI, Life and cross-sell customer distributions on Latent Dirichlet Allocation (LDA) topics, derived from unstructured questions-response data and scored on their similarity to these topics. (2) The building of a functioning PU bagging classifier to identify our current cross-sell customers. (3) Validation of the classifier using OOB probability scores on a dataset with hidden labels. (4) The building of a recommender system to identify opportunities for potential cross-sells in unlabelled or unseen data given a predetermined *decision threshold*.

The remainder of the paper is organised as follows. Section II focuses on related work about recommendation systems within insurance, PU learning and natural language processing (NLP). In Section III we will outline our work: using an NLP approach to transform and analyse our unstructured text data, as well as describing our PU bagging classifier and recommendation system. Section IV will involve experimentation and evaluation of our meta-classifier with some comparisons between different architectures, as well as modelling recommended cross-sells within the unlabelled set. Finally, in Section V we will provide a conclusion to our work.

II. Related Work

A. Topic Modelling and Text Similarity

First looking at NLP approaches—specifically topic modelling and text classification—there is a huge body of research as a result of its wide applicability. Blei *et al*. [7] first applied LDA within machine learning, noting it's success in dealing with the *hard-clustering* and *overfitting problem* encountered by previous models. Jacobi *et al*. [8] and Nikolenko *et al*. [9] both successfully applied LDA to model large amounts of journalistic text and qualitive research respectively. A comparative study by [10] highlighted one key weakness of LDA when dealing with shorter text sequences, proposing a new technique—LDA-U.

Two key methods of text classification will be investigated in this paper. Primarily we will look to the *zero-shot* approach—assigning topic similarity for responses without any explicit training. The zero-shot technique was first proposed by [11] for images, and adapted by [12] for text. This initially involved pretraining the model on the relationships between sentences and sentence tag embeddings. Yin *et al*. [13] built on this proposing a *textual entailment framework* achieving state-of-the-art results without annotated labels using Natural Language Inference (NLI).

The second method is *cosine similarity* of Sentence Bidirectional Encoder Representations from Transformers (S-BERT) embeddings—calculating the cosine angle between transformed text and label. BERT was first proposed by Google researchers [14], using deep bidirectional representations of text to achieve state-of-the-art results in a diverse range of NLP tasks. The model has been adapted in work by [15] and [16], proposing BART and S-BERT respectively. BART performs well for NLI tasks [13] making use of BERT with an autoregressive decoder. S-BERT reduces the computational overhead of BERT on semantic text similarity tasks from 65 hours to 5 seconds by using a *Siamese* network structure.

B. Positive Unlabelled Learning

PU learning [5] deals with learning problems where the unlabelled data can contain both positive and negative examples. It builds a learner that only has access to positive examples and unlabelled data. The earliest PU learning approach, the two-step technique, involves iteratively labelling likely positive and negative data points from the unlabelled dataset until reaching a convergence point where all data is fully labelled. Problems arise as there is no way to verify the proportion of actual classes within the unlabelled dataset, therefore separating between likely positive and negative groups is usually estimated using domain knowledge. Bagging on the other hand has gained in popularity, showing promising performance [17]. Mordelet and Vert [18] highlight the unstable nature of classification in a PU learning setting, which can be successfully exploited using the bagging ensemble approach. Bagging falls under the biased PU learning approach, which treats the unlabelled examples as negatives examples with class label noise (as the unlabelled examples may also contain positive examples, hence noisy) [5].

There are a variety of previously successful examples applying PU learning. Primarily, Li *et al*. [19] combined collective classification with PU learning to formulate a *Collective Positive and Unlabelled Learning* algorithm, achieving state-of-the-art results at classifying fake reviews left on restaurants in Shanghai. Yang *et al*. [20] implemented a PU learning algorithm with weighted support vector machines (SVM's) in disease gene discovery, using a positive set of diseased genes and a set of unlabelled genes, partitioned by suspected positives or negatives. Li *et al*. [21] applied PU learning to the data stream classification problem, extracting high-quality likely positive and negative micro-clusters from unlabelled data.

C. Insurance Recommender Systems

Though cross-sell recommendation systems have been studied extensively within the literature, their use in the insurance market is far more limited. There are a few significant examples; Qazi *et al*. [22] used a Bayesian Network to predict likely insurance products for both new and current customers. Desirena *et al* [23] proposed a recommender system based on a two-stage neural network architecture (Collaborative Learning followed by Survival Analysis) to investigate varying insurance cross-sell opportunities. More general insurance recommendation systems includes work by [24] that used item-item collaborative filtering to predict additional riders to policyholders, and [25] that used association rule mining to find the best life insurance policy for any single person.

III. The Proposed Approach

A. Text Preprocessing

The data used in the research is collected from an insurance company providing PMI and Life insurance, strictly complying with UK GDPR. It contains customer data and conversation scripts between customers and sales advisers. Therefore the first component of the proposed approach is to transform the unstructured responses to insurance advisor questions into numerical features representing topic similarity.

Using LDA we will model six topics for three distinct response categories: *reason for looking*, *medical history* and *occupation*. These three categories represent the most commonly asked questions; while six topics is selected as a good balance between capturing response complexity without adding unnecessary computation.

Fig. 1. The first word Cloud for the *reason for looking* response.

Performing LDA returns topics and their probable coherence. Each topic can be viewed as a *Word Cloud* where size denotes coherence. Using this we can finalise our topics. Fig 1. shows a word cloud for the *reason for looking* response—the first of 18 in total for all categories. This cloud represents specific topics such as *mortgage* or *family*, as well as more generic words such as *insurance* and *cover*. Deriving topics from the Word Cloud implies some non-empirical interpretation. Due to the repetition and similarity of topics we conclude this was a necessary step.

Our final topics are shown in Table I for our trio of response variables, where we can observe a diverse range of different subjects that were discussed by customers.

Having finalised our topics we will move onto text classification. We will utilise two methodologies: zero-shot classification and cosine similarity of S-BERT embeddings. Without a validation set for testing the performance of our classifiers, we hypothesise that using a dual model approach will provide increased reliability as we can compare and contrast results between models.

TABLE I. FINAL TOPICS

Topic	***Reason for Looking***	***Medical History***	***Employment***
1	Mortgage or Family	Pain or Anxiety	Managed or Self-Employed
2	Waiting Time	Heart Related or Asthma	Director or Consultant
3	Access to Treatment	Surgery for Injury	Engineer or Software Engineer
4	Want a Diagnosis	Cancer	Retired
5	Policy Premium	Thyroid Related	Teacher or Civil Servant
6	Peace of Mind	Blood Related	Unemployed

Our first approach, zero-shot classification, will utilise BART for NLI to assess the inference relations between responses and topics. The second approach, cosine similarity of S-BERT embeddings, involves transforming response and topics into sentence embeddings, and calculating the cosine angle between the two. Both inference relations and cosine angles represent topic similarity for responses.

The entire transformation from unstructured responses to topic similarity scores is shown in Fig 2. Response data R is passed to an LDA pipeline where it is tokenised, lemmatised and transformed into a bag-of-words representation with highly common words removed. Following this the LDA model is applied from which—as mentioned previously—we derive six distinct topics T that show high coherence. The two similarity functions are then applied returning topic similarity through both inference relations S_I and cosine similarity S_C for *each response* in R.

The reasons for measuring topic similarity are twofold. Primarily, it allows us to perform *customer segmentation analysis*, observing the overall topic similarity for PMI, Life and cross-sell customer groups, in order to create customer profiles. An example of such analysis on *reason for looking* is shown in Fig 3. where we can observe PMI customers favouring *access to treatment* while Life customers favour *mortgage or family* in terms of responses. Both similarity metrics seem to follow the same distributions on topics, though zero-shot classification is clearly more generous with scoring. Secondly, measurements of topic similarity will act as a numerical representation of the question-response data that can be used in our classification model.

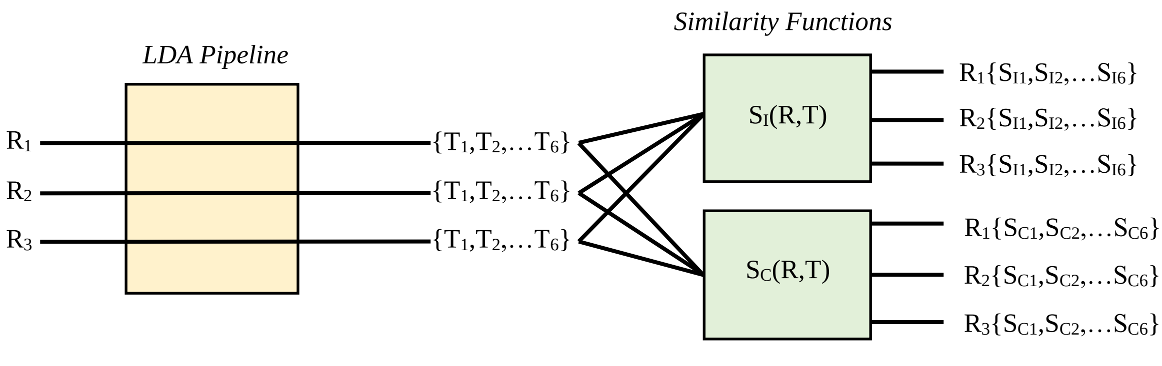

Fig. 2. Overall structure of the text preprocessing. The yellow section shows the LDA pipeline where six topics are derived from unstructured responses. Green sections show the two text similarity functions that measure the inference relations S_I and cosine similarity S_C between responses and topics.

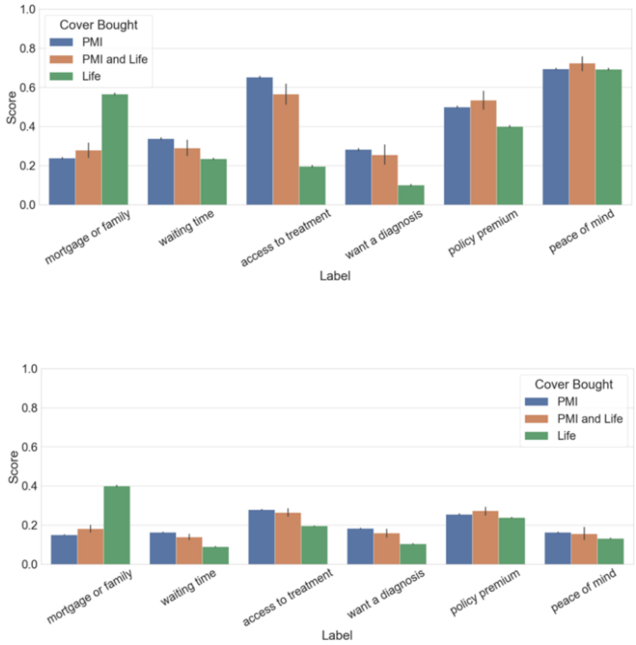

Fig. 3. Bar chart for customer segmentation analysis on the *reason for looking* responses showing the mean similarity scores assigned by zero-shot classification (above) and cosine similarity of S-BERT embeddings (below), with topics as categories and customer groups as coloured bars.

B. *Positive Unlabelled Bagging Classifier*

In this section we will outline our model: a PU bootstrap aggregator, that can successful classify cross-sell customers without training on a negative class. PU learning relies on the Selected Completely At Random (SCAR) assumption [5]—which holds for our data, as the set of positive labels can be framed as a uniform subset of the whole set of positive instances. The methodology for our approach was first proposed by [18], and our implementation is based on the work by [26]. The key idea is to train a classifier C to differentiate between a known positively labelled sample L of size m, and a random sub-sample of size $m^* = m$ taken from the unlabelled class U uniformly and with replacement. The classifier is applied to OOB points at each iteration in order to return a probability score for likelihood of positive class membership. The aggregator then repeats the process through I=*1000* iterations with the same positively labelled sample and different unlabelled sub-samples; finally taking the mean of the scores as the final prediction F for each data point. These predicted probabilities can be used to rank customers or classify them based on a decision threshold.

As mentioned previously, our model falls under the biased learning approach described by [5]. One other feasible method would be the two-step approach outlined by [27]. This involves identifying a set of likely positive and negative examples from the unlabelled class, and building a classifier to iteratively identify similar examples using this information as a base. This approach was not utilised as within our data exploration it was not immediately obvious that there existed a set of likely negative examples, and making assumptions about customers likely would lead to a poorly functioning model. We will however perform tests using the two-step technique to confirm this hypothesis.

We tested three different binary classifiers inside the bagging meta-estimator to try and identify an optimal model, these being: a neural network, a random forest and a light gradient boosting machine (LightGBM). The choice of these learners is a non-trivial one, and we will systematically go through each of the learners to discuss why they were chosen to tackle this problem:

- We implemented the *neural network* primarily for its ability to pick up on complex nonlinear relationships and deal with potential interactions between predictor variables [28]. One further advantage of the neural network is the level of structural complexity: our model makes use of the sigmoid activation function in the output layer, a hidden layer of 20 nodes with the ReLU activation function, a binary cross-entropy loss function and the Adam version of Stochastic Gradient Descent as our optimizer. A potential disadvantage of the neural network is potential over-fitting [29] as the network adapts poorly to new data. We will attempt to deal with this by early stopping of training.

- The *random forest* is an ensemble of unpruned decision trees that uses averaging of its ensemble to control for any potential over-fitting and reduce variance [30]. This learner is interpretable, in the sense that we can obtain a straightforward examination of feature importance through measurement of impurity decrease amongst the trees.

- Gradient boosting involves again using decision trees, added instead using a gradient descent procedure to sequentially reduce bias. *LightGBM* uses *Gradient Based One-Side Sampling* to exclude non-important data instances with low gradients and *Exclusive Feature Bundling* to bundle mutually exclusive features. This allows the algorithm to achieve higher speed and performance than other state-of-the-art gradient boosting models (XGBoost and pGBRT) [31].

To test this model, we cannot look to standard classification metrics. Assessing how well the model classifies examples will be difficult with such a severe class imbalance. More importantly though, is the fact that standardised metrics such as OOB error, accuracy or the Receiver Operating Characteristic (ROC) curve all rely on the creation of a confusion matrix—something that isn't possible without labelled negative instances. Instead, we will take a less orthodox approach and will hide 100/207 positively labelled cross-sells within a random sub-sample of unlabelled data points of the same size (n=100). These 200 data points will form our test set and will be used to assess how well the model picks up on actual cross-sells in comparison to unlabelled customers. This approach likely cannot achieve perfect accuracy however, as there will be at least a few positive instances hidden within the unlabelled set. In practicality this means getting the model to rank the 200 hidden/unlabelled customers by their likelihood of cross-sell, and measuring what percentage of the customers in the top 100 ranking are the hidden cross-sells.

Finally, hyperparameter optimisation was performed using randomised search on our learners and meta-estimator. We favoured this method as it has been shown to have greater efficiency with comparable performance to grid search [32].

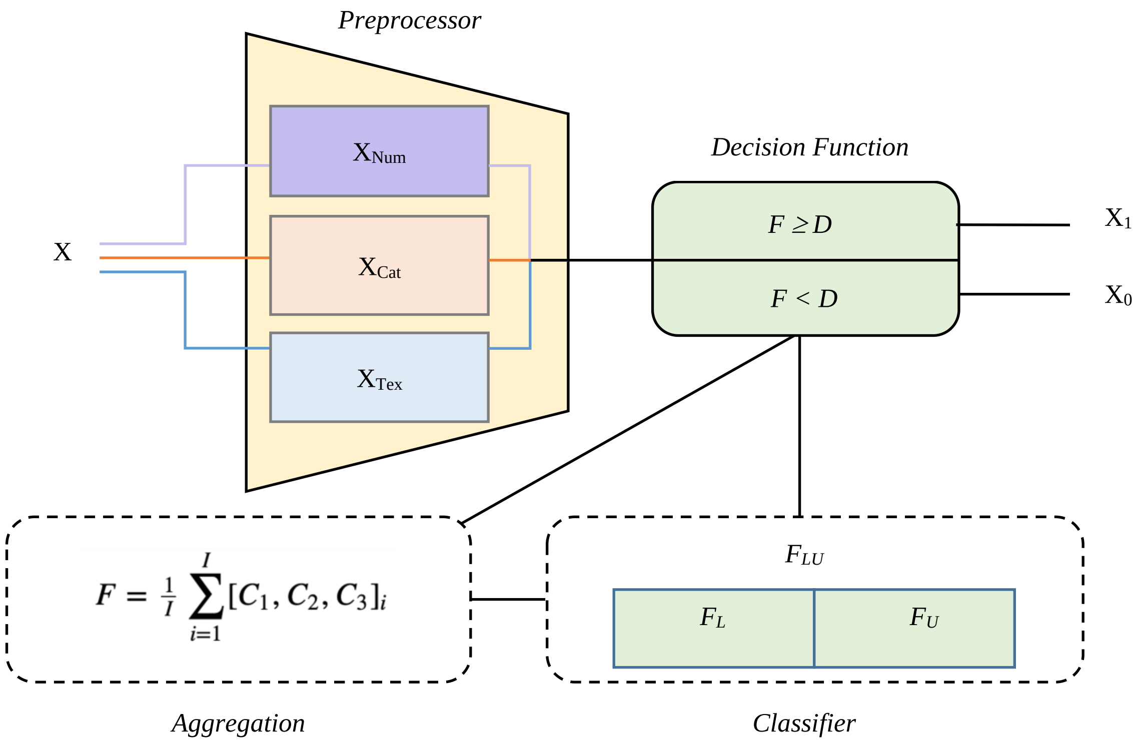

Fig. 4. Full Pipeline of the Recommendation System: the yellow section indicates the preprocessing pipeline, with coloured components inside indicating seperation of the data through the pipeline. Green sections indicate classification, where we are trying to separate our two classes. Dashed lines indicate steps where learning takes place—all of which would occur in a pre-trained model outside of the regular operations of the recommendation system.

C. Customer Recommendation System

The final step in our PU modelling is to create a functioning algorithm that is capable of taking new customers with all of the numeric, categorical and text-based features associated with them, and using these to predict the probability of cross-sell for said customers. The structure of this system is shown in Fig 4.

Each customer X should have most of if not all this data available (features with significant missing values were not used in our approach) as a result of their interaction with an advisor. These features are then run through a pre-processing pipeline, transforming them into standardised numeric features.

Specifically, this pipeline is split into three parts:

- The numerical features X_{Num} use k-nearest neighbours (KNN) for imputation of missing data and are scaled between the values 0 and 1 to prevent feature bias.
- The categorical features X_{Cat} imputes missing values using the most frequently occurring and one-hot encoding values to create binary columns.
- The text features X_{Tex} assign scores based on topic similarity using the inference and cosine metrics outlined previously.

With this data we use our pre-trained PU meta-classifier to differentiate between the positive and unlabelled data points, in order to make a prediction about the new customer X. This classifier has been trained using the PU bagging approach, learning the differences between labelled *L* and unlabelled *U* classes, and aggregating OOB predictions over *I=1000* iterations. In Fig 4. the components of the pre-trained model are placed inside dashed lines to emphasise that they occur outside of the regular scoring mechanism of the system. The model is retrained weekly on all available data, instead of constantly in order to avoid high computational overhead and therefore low system speed.

The pre-trained model makes an aggregate prediction which takes the form of a probability *F* (likelihood of cross-sell) and an advisor can decide whether or not to approach the new customer about cross-purchasing—depending on whether they are above a pre-set threshold *P* or are within their top ranked group of customers *N*. *P* and *N* therefore make up our decision function—*D*. The list [*C1*, *C2*, *C3*] in Fig 4. shows the list of viable learners (neural network, random forest, LGBM) that can be selected for the bagging classifier, with the neural network (*C1*) set as the default. The output of the system are two groups which a customer is assigned to: X_1 which denotes customers recommended for cross-sell approach, and X_0 being the reverse—customers that are not recommended.

As we can see the recommendation system provides a streamlined and structured approach to taking raw customer data and applying the relevant operations to return our ideal customers for cross-sell approach.

IV. Experiments

In this section we perform experiments on our dataset, with dimensions made up of 23675 PMI, 21323 Life and 207 cross-sell customers—for a total of 45205 customers. Each customer has 35 descriptive numerical, categorical and textual features—which increase to 110 after adding topic closeness columns for our unstructured fact-response data, and encode our categorical features. For those customers that made multiple purchases or renewals, we use only the initial purchase, ensuring purely unique customers in our dataset.

We will apply our learners inside the meta-classifier outlined in Section III to our dataset. We provide a comparison against a naïve classifier; one that would on average identify half of the points correctly as our hidden cross-sells.

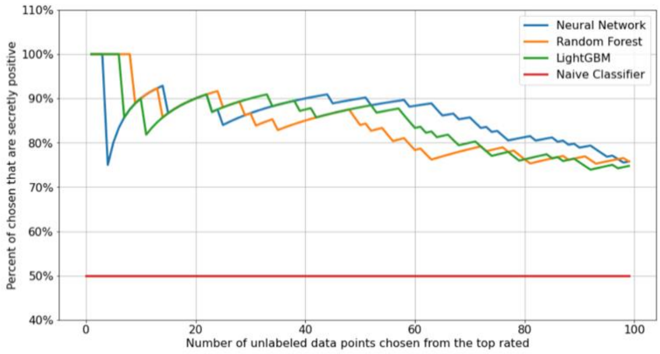

Fig. 5. Comparitive performance of the different learners, inside the begging meta-estimator, at identifying the hidden cross-sell customers compared to the naïve classifier.

In Fig 5. we can see that all of our learners inside the meta-estimator significantly outperform the baseline naïve classifier. We have an upper threshold of 100% for the first few customers—indicating perfect accuracy—which then drops down to a lower threshold of 75% as we move towards the 100th ranked customer. This downward trend is expected, since as we move through the rankings there are fewer and fewer hidden cross-sells for the model to find. Given the performance we can be confident in using this model to find and score new potential cross-sell opportunities within the customer data.

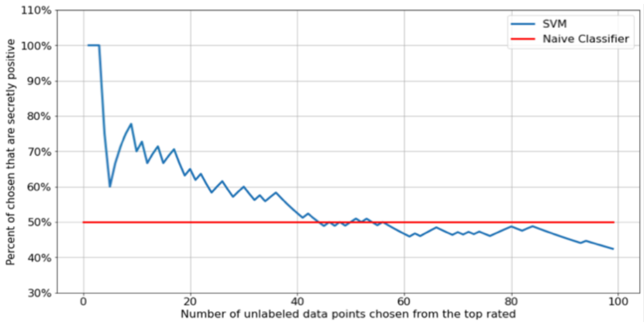

Fig. 6. SVM in the bagging meta-estimator against naïve performance at identifying hidden cross-sell customers.

The original paper for a PU bagging meta-estimator proposed using an SVM as the learner [18]. We tested this learner, which can be seen in Fig 6. showing poor performance against the naïve model. For this reason, alongside the complexity of correctly optimising parameters, we decided against using the SVM learner in our system.

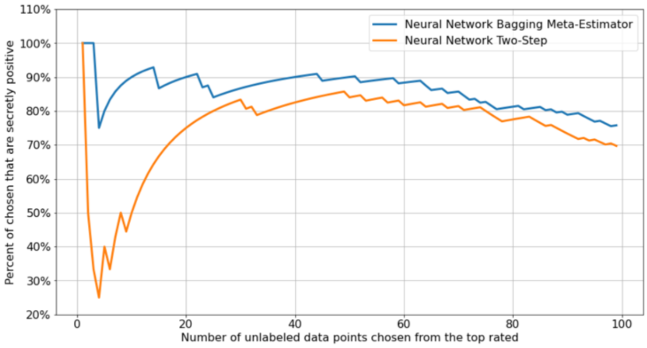

Fig. 7. Performance of the bagging approach against two-step at identifying hidden cross-sells with the neural network.

In Fig 7. we implemented the two-step approach from [26] with our neural network learner to test it against the bagging meta-estimation. The results provide evidence that the bagging meta-estimator shows better and more consistent results than the two-step approach. This is likely a result of not assuming true positive or negatives within the unlabelled set, which could bias the model towards identifying certain customer groups as cross-sells. Further, the greater iterative power of the bagging meta-estimator without a convergence point likely contributed to increased effectiveness. In our experiments the two-step approach utilised very few iterations to reach the convergence point outlined in [27] where likely positive and negative sets are fully scored and labelled.

TABLE II. Experimental Results

Learner	***Proportion of Hidden Cross-Sells Identified***	***AUC***
NN (Bagging)	**76%**	**85.23**
RF	**76%**	82.85
LGBM	75%	83.62
SVM	42%	54.44
NN (Two-Step)	70%	74.30

Table II summarises the final percentage of actual cross-sells captured and Area under the Curve (AUC) for each of the classifiers. AUC is included to measure consistency. We can see the bagging meta-estimator with a neural network (NN) and random forest (RF) share the highest performance for capture, with the neural network having a greater AUC.

TABLE III. Impurity Based Feature Importance

Feature	***Average Importance***
Pain or Anxiety (Medical Zero-Shot)	**0.037**
Age	0.033
Surgery for Injury (Medical Zero-Shot)	0.029
Times Renewed	0.028
Blood Related (Medical Zero-Shot)	0.026
Blood Related (Medical S-BERT)	0.026
Introducer Sales	0.026
Annual Premium	0.024
Heart Related or Asthma (Medical Zero-Shot)	0.023
Cancer (Medical Zero-Shot)	0.021

As mentioned previously, the random forest provides us with impurity-based importance for our input features. The highest 10 ranked features in terms of importance are shown in Table III. From this we can observe that medical responses as a group show significant importance. Further age, traditionally a prime factor with regards to profiling customers within insurance, is key with regards to the models understanding of cross-sell class. Finally, the introducer sales and premium of policies show some importance. These points are significant as future analysis of cross-sell customers will likely yield greater success if it focuses on medical related, age, premium and introducer-based factors.

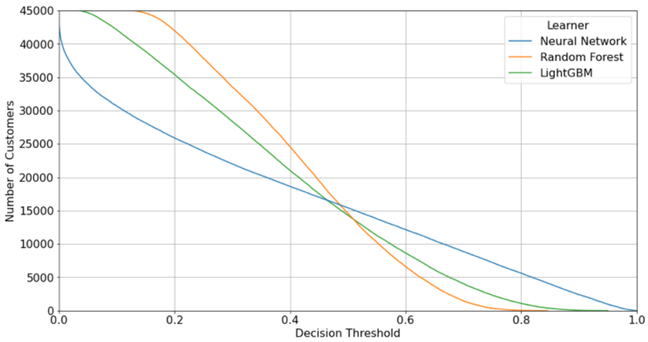

Fig. 8. Decision threshold *P* plotted against the Number of Customers *N* for each of the three learning algorithms inside the bagging meta-clasifier.

Finally, we can plot out our decision threshold against the number of customers for each classifier. To define our decision function *D* we can choose the top *N* ranked customers by the model, or we can assign a fixed decision threshold *P* that customers must be greater than. We would optimally capture a section of the customer base that balances both the number of customers and a high decision threshold. We can visualise the relationship between *N* and *P* with a plot—shown in Fig 8. It is important to note that the neural network was the most generous with allocating high cross-sell probabilities, while the random forest appears to be the strictest.

V. Conclusion

This paper proposes a novel approach for identifying cross-sell opportunities within PMI and Life insurance customer data using PU learning. It first builds on previous work in topic modelling and text similarity to provide analysis and pre-processing of the unstructured customer data to extract features from customer-advisor interaction scripts. Experimental results demonstrate that a bagging meta-estimator combined with a strong learner (neural network, random forest or LightGBM) shows promising results at identifying cross-sell customers. This has been integrated into a recommendation system that can be used to assign cross-sell probability scores to current or new insurance customers, to support advisors for improved cross product selling.

One limitation of this study is the lack of positively labelled data with which to form a full test set. In future, we will apply the recommendation system to real-time customers in order to test its functionality and increase the number of confirmed cross-sells for testing. Further we would aim to integrate other insurance policies such as Income Protection. This would increase the scope of the model both in terms of customer data and diversity of products—likely increasing performance.

A further limitation is the lack of a publicly available dataset for reproducibility of results. Due to the private nature of our dataset complying with UK GDPR however, we conclude that this was not possible.

References

[1] Kamakura, W.A., 2008. Cross-selling: Offering the right product to the right customer at the right time. *Journal of Relationship Marketing*, *6*(3-4), pp.41-58.

[2] Li, S., Sun, B. and Montgomery, A.L., 2011. Cross-selling the right product to the right customer at the right time. *Journal of Marketing Research*, *48*(4), pp.683-700.

[3] Ganganwar, V., 2012. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, *2*(4), pp.42-47.

[4] Chalapathy, R., Menon, A.K. and Chawla, S., 2018. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*.

[5] Bekker, J. and Davis, J., 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, *109*(4), pp.719-760.

[6] Khanal, S.S., Prasad, P.W.C., Alsadoon, A. and Maag, A., 2020. A systematic review: machine learning based recommendation systems for e-learning. *Education and Information Technologies*, *25*(4), pp.2635-2664

[7] Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research, 3, pp.993-1022.

[8] Jacobi, C., Van Atteveldt, W. and Welbers, K., 2016. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital journalism*, *4*(1), pp.89-106.

[9] Nikolenko, S.I., Koltcov, S. and Koltsova, O., 2017. Topic modelling for qualitative studies. *Journal of Information Science*, *43*(1), pp.88-102.

[10] Jónsson, E. and Stolee, J., 2015. An evaluation of topic modelling techniques for twitter. *University of Toronto*.

[11] Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C.D. and Ng, A.Y., 2013. Zero-shot learning through cross-modal transfer. Advances in Neural Information Processing Systems 26 (NIPS 2013).

[12] Pushp, P.K. and Srivastava, M.M., 2017. Train once, test anywhere: Zero-shot learning for text classification. arXiv preprint arXiv:1712.05972.

[13] Yin, W., Hay, J. and Roth, D., 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. arXiv preprint arXiv:1909.00161.

[14] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[15] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[16] Reimers, N. and Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[17] Jaskie, K. and Spanias, A., 2019, July. Positive and unlabeled learning algorithms and applications: A survey. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)* (pp. 1-8). IEEE.

[18] Mordelet, F. and Vert, J.P., 2014. A bagging SVM to learn from positive and unlabelled examples. Pattern Recognition Letters, 37, pp.201-209.

[19] Li, H., Chen, Z., Liu, B., Wei, X. and Shao, J., 2014, December. Spotting fake reviews via collective positive-unlabeled learning. In *2014 IEEE international conference on data mining* (pp. 899-904). IEEE.

[20] Yang, P., Li, X.L., Mei, J.P., Kwoh, C.K. and Ng, S.K., 2012. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, *28*(20), pp.2640-2647.

[21] Li, X.L., Yu, P.S., Liu, B. and Ng, S.K., 2009, April. Positive unlabeled learning for data stream classification. In *Proceedings of the 2009 SIAM international conference on data mining* (pp. 259-270). Society for Industrial and Applied Mathematics.

[22] Qazi, M., Fung, G.M., Meissner, K.J. and Fontes, E.R., 2017, August. An insurance recommendation system using bayesian networks. In

Proceedings of the Eleventh ACM conference on recommender systems (pp. 274-278).

[23] Desirena, G., Diaz, A., Desirena, J., Moreno, I. and Garcia, D., 2019, December. Maximizing Customer Lifetime Value using Stacked Neural Networks: An Insurance Industry Application. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 541-544). IEEE.

[24] Rokach, L., Shani, G., Shapira, B., Chapnik, E. and Siboni, G., 2013, March. Recommending insurance riders. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 253-260).

[25] Gupta, A. and Jain, A., 2013. Life insurance recommender system based on association rule mining and dual clustering method for solving cold-start problem. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(7), pp.1356-1360.

[26] Wright, R., 2017. Positive-unlabeled learning. [online] Roy Wright. Available at: https://roywrightme.wordpress.com/2017/11/16/positive-unlabeled-learning/ [Accessed 29 September 2021].

[27] Kaboutari, A., Bagherzadeh, J. and Kheradmand, F., 2014. An evaluation of two-step techniques for positive-unlabeled learning in text classification. Int. J. Comput. Appl. Technol. Res, 3, pp.592-594.

[28] Tu, J.V., 1996. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology, 49(11), pp.1225-1231.

[29] Livingstone, D.J., Manallack, D.T. and Tetko, I.V., 1997. Data modelling with neural networks: advantages and limitations. Journal of computer-aided molecular design, 11(2), pp.135-142.

[30] Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

[31] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, pp.3146-3154.

[32] Bergstra, J. and Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of machine learning research, 13(2)