Cost-sensitive Churn Prediction in Fund Management Services

James Brownlow[1,2], Charles Chu[1,2], Bin Fu[1], Guandong Xu[2⋆], Ben Culbert[1,2], and Qinxue Meng[1]

[1] Colonial First State, Sydney, Australia, 2000
{James.Brownlow, Charles.Chu, Bin.Fu, Ben.Culbert, Qinxue.Meng}@cba.com.au
[2] Advanced Analytics Institute, UTS, Sydney, Australia, 2007
Guandong.Xu@uts.edu.au

Abstract. Churn prediction is vital to companies as to identify potential churners and prevent losses in advance. Although it has been addressed as a classification task and a variety of models have been employed in practice, fund management services have presented several special challenges. One is that financial data is extremely imbalanced since only a tiny proportion of customers leave every year. Another is a unique cost-sensitive learning problem, i.e., costs of wrong predictions for churners should be related to their account balances, while costs of wrong predictions for non-churners should be the same. To address these issues, this paper proposes a new churn prediction model based on ensemble learning. In our model, multiple classifiers are built using sampled datasets to tackle the imbalanced data issue while exploiting data fully. Moreover, a novel sampling strategy is proposed to deal with the unique cost-sensitive issue. This model has been deployed in one of the leading fund management institutions in Australia, and its effectiveness has been fully validated in real applications.

Keywords: Customer retention, Churn prediction, Cost-sensitive classification, Imbalanced data

1 Introduction

Fund management services refer to the institutions that help customers achieve their wealth goals by providing them with a range of investment options, i.e., funds. Since a customer could have an investment of thousands or even millions of dollars, it is vital for them to retain their valuable customers. To this end, a practical approach is to predict which customers would quit, i.e., churners as soon as possible, then a retention campaign which targets these potential churners could be launched. Churn prediction can be viewed as a binary classification task, which is one of the fundamental concepts in data mining. Basically, a set of customers classified as *churner* or *non-churner* aka a training set is used to

⋆ Corresponding author

learn a predictive model, which is used to predict churn probabilities of customers whose classes are unknown. Nowadays, churn prediction is receiving increasing attention from both academia and industry. A multitude of methods such as boosting [1], random forest [2], and neural network [3] etc. have already been investigated and employed for churn prediction in various applications, including telecommunication [1, 4], online community [5], and social game [6], and so forth.

Despite these achievements in other industries, the fund management industry has its own particular challenges, meaning that existing methods cannot be employed directly. One is that financial data is even more imbalanced compared with other industries. Sampling techniques like undersampling are commonly used to cope with this issue [7]. However, how to sample a set of informative and diverse subsets still needs further investigation. Another major challenge is that a unique cost-sensitive problem is presented. Costs in existing applications are either class-dependent or instance-dependent [8]. However, the cost of churn prediction in financial industry belongs to neither of them. On one hand, costs of wrong predictions for churners should be proportional to their account balance, so these costs are instance-dependent. One the other hand, wrong predictions for non-churners could be the same loss which should be less than the loss associated with any churner, so these costs are class-dependent. Thus the cost here actually is a hybrid of class-dependent cost and instance-dependent cost. To our knowledge, there are few approaches for dealing with this special type of cost at the moment.

To tackle these challenges, we propose a novel approach based on ensemble learning for churn prediction in this paper. Specifically, multiple balanced subsets are sampled from the original dataset, multiple classifiers are then learnt and combined using these subsets. Although similar paradigms have been used in [2, 9], we introduce a new sampling strategy that consists of two separate sampling steps with different weighting mechanisms for two classes respectively. The advantages of our approach include: (1) this novel sampling strategy uses different weighting mechanisms for different classes, thus the special cost-sensitive issue can be handled properly; (2) sizes of the subsets are determined randomly, so they are varied instead of being fixed as in [9], this additional randomness could increase the diversity of classifiers and achieve better performance accordingly. Gradient boosting machine [10] is used in our approach to learn the classifiers to improve the performance further. To summarize, this paper makes the following three main contributions. (1) A new weighting mechanism and sampling strategy is proposed to deal with the imbalanced data and special cost-sensitive problem. (2) The concrete process of how this approach has been deployed in real production is introduced. (3) Extensive experiments with real-world data have been conducted to validate the effectiveness of our approach.

The rest of this paper is organised as follows. Section 2 previews related work. Section 3 gives the notations used throughout this paper as well as a formal formulation of the learning task. Section 4 introduces the specific implementation of our proposed model in the real scenario. Section 5 presents the experimental results, followed by conclusions in Section 6.

2 Related Work

2.1 Churn Prediction

Over the last decade, churn prediction has been applied in various fields, e.g., telecommunication, social networks, and mobile application [4, 6, 11]. In most cases, it is solved as a classification problem through learning a predictive model using a set of customers whose classes are known. A customer is usually represented as a vector of features, and the relationship between a customer's features and class could be captured by the model. Generally, there are two keys to learn a good model, one is how to define a set of discriminative features that could cover underlying factors, and the other is how to determine the form of model that is suitable for current data. Every particular application has its distinctive data from which features can be derived. For example, business data and operation data are exploited in the telecom industry [4], question and comment data are analysed in online question answering services [12], etc. Although each application has its unique features, existing classification methods can be used in these applications commonly. Popular methods such as boosting, random forest and logistic regression have already been employed [1, 4, 12], and a comprehensive review of methods used in the telecom industry is also given in [13].

2.2 Imbalanced Data and Cost-Sensitive Learning

Imbalanced data must be carefully handled otherwise the learning process would be skewed towards the majority class while the minority class is ignored. Two primary strategies can be employed to cope with imbalanced data, i.e., method transformation and data transformation [14]. The former adapts learning methods to enable them to handle imbalanced data directly. For instance, a skew-insensitive splitting criteria is adopted in decision tree [15]. By contrast, the latter aims to obtain balanced datasets, so existing methods can be used without adaptation. For example, oversampling and undersampling techniques obtain balanced data via varying the size of data of one particular class [7]. Since useful information could be missed in undersampling, ensemble learning based methods have become popular recently. These methods follow the same paradigm in which multiple subsets are sampled, but differ from each other in terms of weighting mechanisms in the sampling process [2, 9, 16]. Cost-sensitive learning is closely related to imbalanced data and has been used as a weighting mechanism to make data balanced [17]. As stated above, there could be a class-dependent cost or an instance-dependent cost. A classical strategy of dealing with class-dependent cost is to define a cost matrix and determine predictions using Bayes optimal rule [18]. In addition, weighting instances according to their relevant costs is another typical strategy of encoding costs into the learning process [19].

From aforementioned work, it can be observed that assigning appropriate weights to instances is a critical way of dealing with imbalanced data as well as the cost-sensitive learning issue. Inspired by the EasyEnsemble method [9], our approach also adopts the ensemble learning paradigm to obtain balanced subsets

as well as take full advantage of available data. The key difference is that a novel weighting mechanism based on customers' balance is designed in our approach to handle the special cost-sensitive issue.

3 Problem Formulation

Let $P = \{(x_i, 1)\}_{1 \leq i \leq |P|}$ be a dataset of minority class in which x_i denotes the ith customer whose class is 1, i.e., churner. Similarly, Let $N = \{(x_i, 0)\}_{1 \leq i \leq |N|}$ be a dataset of majority class in which every customer x_i' class is 0, i..e, non-churner. The size of N should be considerably larger than the size of P, i.e., $|P| \ll |N|$. A customer x is represented as a feature vector $x = \langle x_1, x_2, \ldots, x_n \rangle$, and these features could be demographic information and behavioural patterns extracted from historical transactions.

The task of classification is to learn a predictive model f based on a training set $D = P \bigcup N$. Essentially, a model f is a function that establishes a mapping from instance space to class space, i.e.,

$$f(x) \rightarrow c, c \in \{0, 1\} \tag{1}$$

Given an instance x, c is the class predicted for it by f. The output f could also be a real value $y(0 \leq y \leq 1)$ which indicates the probability of $c = 1$.

In order to learn a good model, aforementioned imbalanced data and the cost-sensitive learning issue must be handled properly. An effective strategy to handle imbalanced data is undersampling. Specifically, a subset N' is sampled from N, and a model is then learnt based on training set $D' = N' \bigcup P$. Usually we choose $|N'| = |P|$, so D' is balanced. One issue of undersampling is that the majority of N is excluded, resulting in much useful information being unexploited. Hence, recent methods often follow the paradigm of integrating ensemble learning with sampling as shown in Algorithm 1.

Algorithm 1: Ensemble of multiple samplings

Data: Training set N and P, iteration number t
Result: Multiple classifiers $f = (f_1, f_2, \ldots f_t)$

```
for k ← 1 to t do
    sample P_i from P according to weights of instances in P;
    sample N_i(|N_i| = |P_i|) from N according to weights of instances in N;
    learn a model f_i using D_i = N_i ⋃ P_i
return f = (f_1, f_2, ... f_t);
```

As shown in Algorithm 1, every classifier f_i is learnt using a balanced dataset D_i, and N is fully exploited through multiple samplings. Methods that follow this paradigm differ mainly on: (1) how to set the weights of instances in N and P, (2) the size of N_i and P_i, and (3) the method used to learn classifiers. For example, in the EasyEnsemble method, N_i is sampled evenly from N with every instance having the same weight, P_i is simply set as P so that $|N_i| = |P_i| = |P|$, and AdaBoost [9] is used to learn classifiers. Our approach also adopts this

paradigm, and the remaining problem is how to design weighting mechanisms, determine sizes of subsets, and combine multiple classifiers to deal with the special cost-sensitive issue. The solution is introduced in following section.

4 Model Design and Implementation

In this section, our proposed approach is introduced. Particularly, the framework and steps of its implementation in practice are also presented.

4.1 Our Learning Approach

Two types of wrong predictions could possibly happen, i.e., predicting a churner as a non-churner and predicting a non-churner as a churner. The former is costly because failing to identify a churner could lead to loss of all his or her money. The more money he or she has, the greater the cost will be. Consequently, the cost of a wrong prediction for churners should be proportional to their account balance. However, the latter would not incur much loss and has nothing to do with customers' account balance. Hence it is reasonable to set the cost of wrong predictions for non-churners as a fixed value. With this assumption, the weight w_i assigned to every instance x_i in dataset N and P is set according to Equation (2) in our approach.

$$w_i = \begin{cases} \frac{1}{|N|} & \text{if } x_i \in N \\ \frac{b_i}{\sum_{x \in P} b_x} & \text{if } x_i \in P \end{cases} \tag{2}$$

Here b_i is x_i's account balance. It can be seen from Equation (2) that weights assigned to churners are proportional to their individual account balance, while weights assigned to non-churners are the same which is a class level value.

Next, instances should be sampled from N and P according to their weights to take costs into consideration when learning models. Instances with greater weights would appear more times in the new training set, thus the likelihood of making wrong predictions for them is reduced. Here a key point is how to determine the sizes of sampled subsets. Instead of setting $|N_i|$ and $|P_i|$ always as $|P|$, we use a straightforward method to introduce randomness in the sizes of subsets. Specifically, when sampling a subset P_i from P, a subset P' of size $|P|$ is sampled according to Equation (2) firstly, then these instances which exist in P but not in P' will also be added into P' to form P_i. In this way, the size of P_i is a random value which ranges from $|P|$ to $2|P| - 1$. A subset N_i of size of $|P_i|$ is then sampled from N, so that $(P_i \bigcup N_i)$ is a balanced dataset. We can see that now the imbalanced data and the cost-sensitive issue are well addressed in this way.

Furthermore, Xgboost [10], which is a popular implementation of the gradient boosting machine model, is employed in our approach to learn models. It is an additive model which consists of multiple submodels, and every submodel is obtained through minimizing the residuals produced by previous models. Now, all the key issues are solved, and the details of our approach are specified in Algorithm 2.

Algorithm 2: Our proposed approach

Data: Training set N and P, iteration number t
Result: Multiple classifiers $f = (f_1, f_2, \dots f_t)$

1 **for** $i \leftarrow 1$ **to** t **do**
2 sample $P'(|P'| = |P|)$ from P using weights according to Equation (2);
3 $P_i = P' \bigcup (P \setminus P')$;
4 sample $N_i(|N_i| = |P_i|)$ from N using weights according to Equation (2);
5 $f_i \leftarrow Xgboost(N_i \bigcup P_i)$
6 return $f = (f_1, f_2, \dots f_t)$;

The output of Xgboost for binary classification is a real value in $[0, 1]$ which denotes the probability of being a churner. After obtaining multiple models, we simply use the average of their outputs as the final prediction for x as shown in Equation (3).

$$f(x) = \frac{1}{t} \sum_{i=1}^{t} f_i(x) \tag{3}$$

Here f_i is the ith model learnt in the ith iteration.

It can be observed that our approach has several advantages: (1) weights based on account balance are introduced, so it is less likely to make wrong predictions for high value churners; (2) line 3 of Algorithm 2 indicates the size of every subset is randomly determined, so models learnt using these subsets would be more diverse and the performance could be improved via reducing variance accordingly; (3) the size of subset N_i is larger than $|P|$, so more information about the majority class could be exploited when learning models compared with other methods like EasyEnsemble.

4.2 Model Implementation

Our approach has been applied in a fund service company in Australia. In this section, how to prepare data and define features in practice is introduced.

Data Sources. Multiple sources of data regarding various entities exist in reality, and data from heterogeneous sources should be integrated to get a comprehensive understanding of customers. In our implementation, the primary types of data that have been exploited are: (1) customer demographic information, (2) customer behaviour, such as call log and online system login, (3) account status, (4) transaction records, (5) fund performance such as daily records of fund price, (6) insurance records, and (7) interaction with advisers such as records of adviser fees etc.

Feature Engineering. Six types of features as below are extracted.

(1) **Customer demographic features**. These features provide information regarding customers' profiles, such as *gender*, *age*, and *occupation* etc.

(2) **Customer behavioural features**. Customers' past behaviours or interactions with a company contain some useful clues for their future behaviours. Typical features of this type includes *call frequency*, *survey rating*, and so on.

(3) **Account level features**. Two types of account level features are extracted. The first one relates to an account's current status, such as *tenure* and *balance*. the other describes an account's changing trend in the past, i.e., *balance change*, and *option change*.

(4) **Fund performance**. Customers are usually sensitive to their investment returns. Therefore, we also extract features like *fund performance* to measure the growth rate of a customer's investment in the past year.

(5) **Adviser and dealer features**. Although we do not have much data about advisers and dealers, we can infer their features through customers associated with them. Features such as *number of customers*, and *number of churn customers* are constructed under the assumption that if many customers who belong to an adviser have left, other customers belonging to the same adviser are also likely to leave in the near future.

(6) **Employer features**. We also extract a set of features regarding employers. Features like *number of employees*, *number of churn employees* are extracted to measure the impact of an employer on its employees.

Around 120 features are defined totally. For every customer, his or her final feature vector is the combination of features of all above 6 types. The overall framework of model implementation is outlined in Fig. 1.

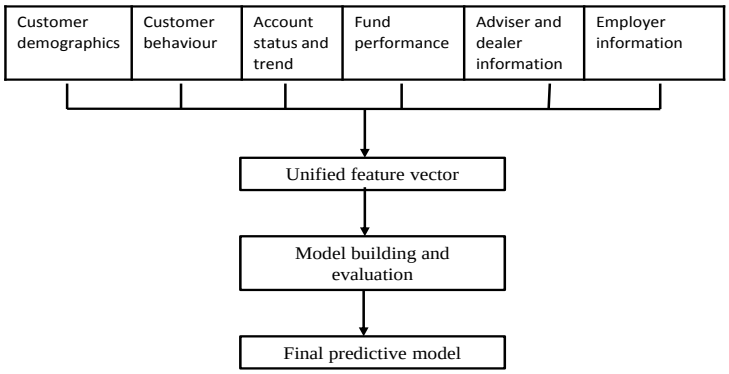


Fig. 1: Framework of model implementation

As shown in Fig. 1, for every customer, multiple sets of features are extracted from different perspectives. These features are then combined into a single feature vector. Therefore, a unified view which covers the influential factors as much as possible is obtained, increasing the probability of building a reliable model.

5 Experiments

In this section, extensive experiments using data from real applications are conducted to validate our approach's effectiveness.

5.1 Datasets

Datasets of four different funds are used in experiments. They are *retail superannuation*, *corporate superannuation*, *pension*, and *investment*. To generate these datasets, data between Jan 2016 and Dec 2016 (observation window) are extracted to generate features that are introduced in previous section, and data between Jan 2017 and Jun 2017 (label window) are extracted to determine classes. A customer is classified as a churner if his or her account is closed in the specified label window, otherwise is classified as a non-churner. The purpose here is to use a customer's information in the past one year to predict his decision in the next six months.

After excluding outliers and customers whose accounts are opened within the observation window because they do not have sufficient historical data, Table 1 gives the summary of the four datasets in detail.

Table 1: Description of datasets

Dataset	$\|D\|$	$\|N\|$	$\|P\|$	*churn ratio*
retail super	220000	210320	9680	4.4%
corporate super	260000	243300	16640	6.4%
pension	135000	128925	6075	4.5%
investment	160000	152480	7502	4.7%

In Table 1, $|D|$, $|N|$, and $|P|$ is the size of the whole population, non-churners, and churners respectively, and *churn ratio* is the ratio of churners in the population, i.e., $|P|/|D|$. It can be observed that all of these datasets are extremely imbalanced.

5.2 Evaluation Metrics

In practice, churn prediction models are used to predict the churn probabilities or attrition scores of existing customers. These scores are then ordered descendingly, so a retention campaign could focus on the most likely churners, i.e., the top K customers. In this case, a model can be evaluated in two manners. One is the number of true churners in top K customers, and the other is the sum of true churners' account balance in top K customers. Accordingly, two evaluation criteria are used in our experiments.

The first one is *recall*, and its definition is given in Equation (4)

$$R@k = \frac{\sum_{x \in Top(k)} c_x}{|P|} \tag{4}$$

Here $Top(k)$ denotes the top K customers. c_x is customer x's label, and it could be 1 or 0, 1 indicates x is a churner and 0 indicates the opposite.

The second one is *balance recall*, which is defined in Equation (5)

$$BR@k = \frac{\sum_{x \in Top(k)} c_x * b_x}{\sum_{x \in P} b_x} \tag{5}$$

Here $Top(k)$ and c_x have the same meaning as above, and b_x is customer x's account balance. For both of these two criteria, a greater value means a better model performance.

5.3 Baselines and Settings

We compare our proposed method with three classical methods of coping with imbalanced or cost-sensitive data. These methods are:

- Balanced random forest [2]. In its ith iteration of learning a decision tree, a subset P_i and $N_i(|P_i| = |N_i| = |P|)$ is evenly sampled from P and N.
- WeightGBM. It is Xgboost with class-dependent weights [10]. Weights of instances in P are set as $|N|/|P|$ in this method.
- EasyEnsemble [9]. In its ith iteration, only a subset N_i is sampled from N, and $P_i = P$.
- CostGBM, our proposed approach.

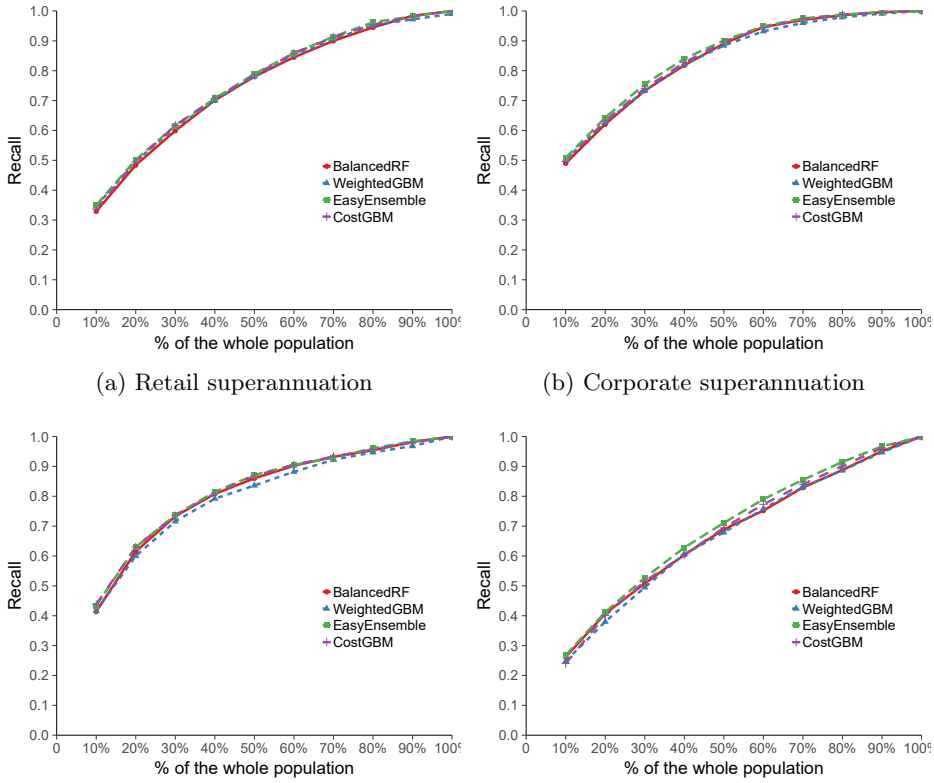
The purpose of comparing our approach with these baselines is to validate the effectiveness of the weighting mechanism designed in this paper, especially in terms of the criterion *balance recall*. To make the comparison fair and convincing, Xgboost is also used in EasyEnsemble instead of Adaboost. The size of Balance random forest, i.e., number of trees is set as 200. In all other 3 methods, the number of iterations is 10 and a Xgboost model with 200 trees is learnt in every iteration. When learning Xgboost model, 'binary:logistic' is chosen as the objective function, and the optimal learning rate is chosen from $0.05-0.3$ through multiple trials. All these methods are implemented in R environment, and the R package *Xgboost* is used to learn Xgboost models.

5.4 Results and Analysis

Datasets are split into training set (80%) and test set (20%). Models are then built using the training sets and evaluated using the test sets. All these methods generate numeric predictions as churn probabilities, and the population as in test sets are ranked in terms of their predictions in a descending order.

To begin with, these methods are evaluated and compared in terms of recall, and the results on the four datasets are depicted in Fig. 2. For any point in Fig. 2, its x value is the top percentage of the whole population, and its y value is the recall. We can see that while EasyEnsemble performs slightly better on these datasets, our proposed method also shows competitive performance in terms of recall, even it gives more focus on high value customers.

When it comes to balance recall, our proposed method outperforms the other three methods significantly on all datasets as shown in Fig. 3. Take the results



(a) Retail superannuation (b) Corporate superannuation

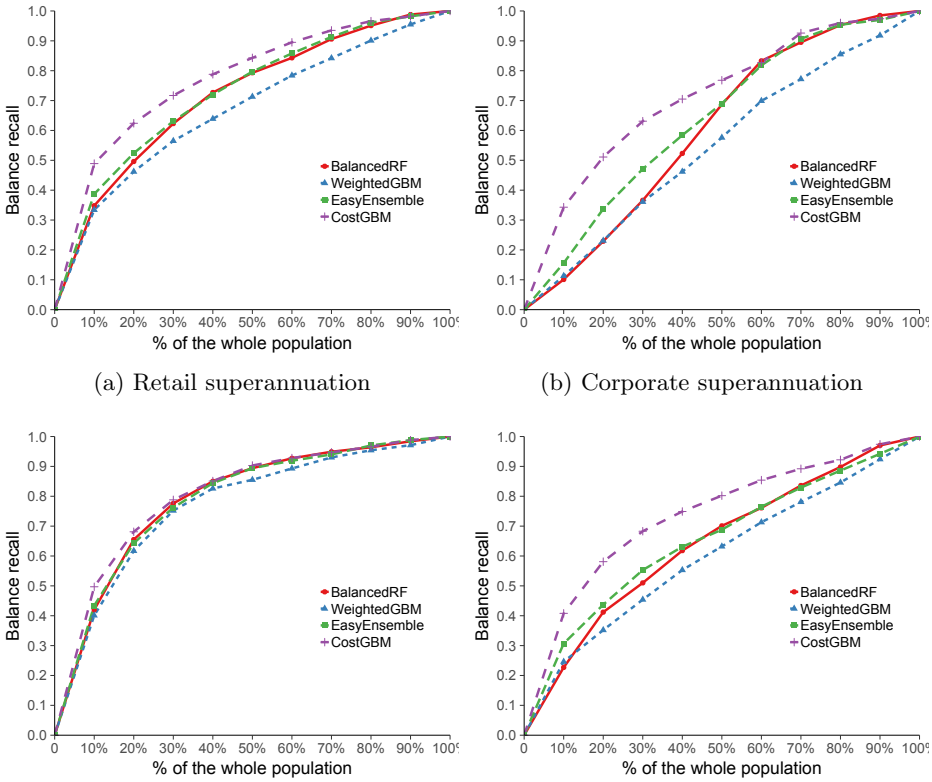
(c) Pension (d) Investment

Fig. 2: Model performance in terms of recall

on corporate superannuation as example, when we look at the top 10% percent of the population, the balance recall of Balanced random forest, WeightedGBM, EasyEnsemble, and CostGBM is around 0.1, 0.1, 0.15, and 0.35 respectively. We can see that the total balance of true churners identified by our method is around 2 times greater than those identified by other methods. Given the volume of corporate superannuation, it means the improvement gained by our method could be millions of dollars.

6 Conclusions

This paper introduces a novel method for churn prediction in fund management services and its implementation in a fund management company in Australia. A sampling framework based on ensemble learning and a new weighting mechanism based on account balance are proposed to deal with imbalanced and cost-sensitive issues with financial data. The practical steps of model implementation



(a) Retail superannuation (b) Corporate superannuation

(c) Pension (d) Investment

Fig. 3: Model performance in terms of balance recall

are also introduced, especially how various data from heterogeneous sources are exploited and integrated to gain a unified view of customers. Evaluation using real word data validates our model's superiority in capturing high value churners compared with traditional methods. Moreover, our method has been applied in real applications and assists the marketing team to narrow down their campaign target. In future work, strategies of incorporating account balance based cost into other advanced models will be investigated, and more features will also be extracted to enhance learning performance.

References

1. Lu, N., Lin, H., Lu, J., Zhang, G.: A customer churn prediction model in telecom industry using boosting. IEEE Transactions on Industrial Informatics **10**(2) (2014) 1659–1665
2. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical report, University of California, Berkeley (2004)

3. Ismail, M.R., Awang, M.K., Rahman, M.N.A., Makhtar, M.: A multi-layer perceptron approach for customer churn prediction. International Journal of Multimedia and Ubiquitous Engineering **10**(7) (2015) 213–222
4. Huang, Y., Zhu, F., Yuan, M., Deng, K., Li, Y., Ni, B., Dai, W., Yang, Q., Zeng, J.: Telco churn prediction with big data. In: Proceedings of the 2015 ACM I nternational Conference on Management of Data. (2015) 607–618
5. Rowe, M.: Mining user lifecycles from online community platforms and their application to churn prediction. In: Proceedings of the 13th IEEE International Conference on Data Mining. (2013) 637–646
6. Runge, J., Gao, P., Garcin, F., Faltings, B.: Churn prediction for high-value players in casual social games. In: Proceedings of the 2014 IEEE Conference on Computational Intelligence and Games. (2014) 1–8
7. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9) (2009) 1263–1284
8. Zhang, Y., Zhou, Z.H.: Cost-sensitive face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(10) (2010) 1758–1769
9. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(2) (2009) 539–550
10. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, ACM (2016) 785–794
11. Rothenbuehler, P., Runge, J., Garcin, F., Faltings, B.: Hidden markov models for churn prediction. In: Proceedings of the SAI Intelligent Systems Conference. (2015) 723–730
12. Dror, G., Pelleg, D., Rokhlenko, O., Szpektor, I.: Churn prediction in new users of yahoo! answers. In: Proceedings of the 21st International Conference Companion on World Wide Web. (2012) 829–834
13. Mahajan, V., Misra, R., Mahajan, R.: Review of data mining techniques for churn prediction in telecom. Journal of Information and Organizational Sciences **39**(2) (2015) 183–197
14. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C **42**(4) (2012) 463–484
15. Cieslak, D.A., Chawla, N.V.: Learning decision trees for unbalanced data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer (2008) 241–256
16. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition **46**(12) (2013) 3460–3471
17. Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition **40**(12) (2007) 3358–3378
18. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (1999) 155–164
19. Zadrozny, B., Langford, J., Abe, N.: Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings of the Third IEEE International Conference on Data Mining, IEEE (2003) 435–442