Predicting Customer Churn for Insurance Data

Chapter · September 2020

DOI: 10.1007/978-3-030-59065-9_21

3 authors, including:

Michael Scriney
Dublin City University
25 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Mark Roantree
Dublin City University
126 PUBLICATIONS **749** CITATIONS
SEE PROFILE

Predicting Customer Churn for Insurance Data*

Michael Scriney, Dongyun Nie, and Mark Roantree

Insight Centre for Data Analytics, School of Computing,
Dublin City University, Ireland
michael.scriney@dcu.ie, dongyun.nie@dcu.ie, mark.roantree@dcu.ie

Abstract. Most organisations employ customer relationship management systems to provide a strategic advantage over their competitors. One aspect of this is applying a *customer lifetime value* to each client which effectively forms a fine-grained ranking of every customer in their database. This is used to focus marketing and sales budgets and, in turn, generate a more optimised and targeted spend. The problem is that it requires a full customer history for every client and this rarely exists. In effect, there is a large gap between the available information in application databases and the types of datasets required to calculate customer lifetime values. This gap prevents any meaningful calculation of customer lifetime values. In this research, we present an approach to generating some of the missing parameters for CLV calculations. This requires a specialised form of data warehouse architecture and a flexible prediction and validation methodology for imputing missing data.

1 Introduction

One of the major goals of Customer Relationship Management is to maximise the *Customer Lifetime Value* (CLV) for the purpose of supporting long term business investment [8]. CLV is a measure that focuses on predicting the net profit that can accrue from the *future* relationship with customers [4]. This metric can be calculated by recording the behaviours of the customer over the longer term and thus, help to build a customised business strategy. It has been a popular research topic, addressed by researchers in different ways, for example, formulaic CLV [11] and Probability Model CLV [15]. One of the core elements in CLV models [5] and the calculation of CLV scores is customer *retention* or its opposite, customer *churn*. In the business sector, customer churn is commonly used not only to support CLV predictions, but to maximise customer profitability by establishing resource allocation decisions for marketing, sales, and customer interaction. As a result of the benefits that churn analysis provides, this topic has become popular for industrial research in recent years. Some of business based research focuses on statistical efforts, often required for CLV calculations [17]. Information technology based research generally experiments with data mining techniques [7] to try to generate the variables needed for CLV scores. In the telecom sector, churn analysis research has been shown to require a specific set of variables [18] for effective results.

* Research funded by Science Foundation Ireland Grant SFI/12/RC/2289_P2.

1.1 Problem Statement

One of the issues with CLV research and the generation of variables such as *churn* is the highly theoretical nature and focus of this area of research. There has been little research on deriving necessary attributes from real world datasets which require activities such as the construction, transformation and enrichment of datasets to make them suitable for existing CLV calculations. Researchers in [2] highlighted the variables that are required as input to CLV calculations. Here, a is the acquisition rate; A is the acquisition cost per customer; d is the yearly discount rate; m is the net income of a transaction; R is the retention cost per customer per year; and r is the yearly retention rate.

On the surface, it appears as if the generation of CLV scores for all customers is a straightforward process but in reality, many of these variables are not easily extracted from enterprise databases. From [2], m and d can be deduced using feature extraction and a detailed clustering process but all others will generally require some form of imputation after a segmentation process. Customer *segmentation* is regarded as a natural process to help companies to classify customers and plan market investment strategies such as direct sales. As such, it has been widely adopted by industry planners or in data warehousing similar efforts include fragmentation of the data and queries [13]. Moreover, it plays a critical role in the development of the company's position by combining product differentiation and marketing segmentation to provide resources, objectives and competences to the company.

1.2 Contribution and Paper Structure

In previous work [10], we presented a methodology for constructing a unified client record, as calculating customer lifetime values (CLV) is not possible without a complete client history. In this paper, we extend this work as we employ a suite of machine learning algorithms to derive the retention rate r for all customers.

The challenge is due to the fact that no single model has been shown to provide the best accuracy. Our approach begins by process of feature extraction from the unified customer dataset and using these variables, derive the retention rate, r, suitable for CLV calculations. We then perform customer churn prediction using ten experimental configurations in order to determine the best method for calculating r. In this way, we provide an understanding of which methods deliver accurate churn predictions for individual customers. A robust validation mechanism uses a number of metrics by which to determine those models that perform best.

Paper Structure. The paper is structured as follows: in §2, we discuss related research in predicting customer churn; §3 describes the steps necessary to construct the dataset used as input for the ten models; in §4, we present our results, evaluation and discussion; and finally, in §5 we present our conclusions.

2 Related Research

In [6], the authors introduce a hybrid approach for calculating customer churn by combining decision trees and neural networks. The method was then evaluated using supermarket data. Similar to our work a significant amount of prepossessing of the data is required. However, the authors did not provide a comparison of the hybrid approach to other models. The authors in [1] analyse insurance data using logistic regression with a generalised additive model (GAM). This research uses data from the insurance sector but here, the researchers benefited from more fine-grained data, which provided a month-by-month view of the data. As a result, the authors did not see the need to provide a comparison between the GAM model and other classification algorithms. In [3], the authors present a means of predicting customer churn for motor insurance. They compare four methods used to calculate churn: decision trees, neural networks, logistic regression and support vector machines. In terms of accuracy across models, the authors neural network approach had a similar accuracy to ours ($\backsim 88\%$). However, the authors' dataset focused specifically on motor insurance. Additionally, our evaluation compares ten different classification methods. In [16], the authors propose a one-class support vector machine which can be used to under sample data. The efficacy of the approach was evaluated using five classification methods on a motor insurance fraud dataset with a credit card churn dataset. The five methods employed were: decision trees, support vector machines, logistic regression, probabilistic neural networks and a group method for data handling. Similar to the authors research, we employ decision trees, support vector machines and neural networks for our evaluation. However, the authors' insurance dataset focused on fraud detection within motor insurance while our approach attempts to calculate customer churn across multiple insurance types. The authors in [19] combine deep and shallow modelling to predict customer churn on insurance data. Similar to our approach, their evaluation compared the performance of their method with several classification algorithms using a similar set of metrics. However, the authors used a large dataset consisting solely of life insurance policies whereas our dataset contains several policy types. This difference is crucial as it results in different sets of dimensions, motivating the need for different methods to be applied. In [12], the authors evaluate the *staying power* of various models of churn prediction for two domains: insurance and internet service providers. The insurance dataset comprises life insurance churn data and four predictive method were used: logit models and decision trees both with and without bagging. The authors found the decision tree method with bagging to show the best performance. In our research we employ a wider range of prediction algorithms to provide a greater comparison of churn prediction methods.

In summary, there are more commonly used when predicting churn for insurance data: decision trees, support vector machines and logistic regression models. However, recent research examined the efficacy of various neural networks. Factors such as granularity, size, available features (and the types of each feature) have a significant impact when determining which model performs best.

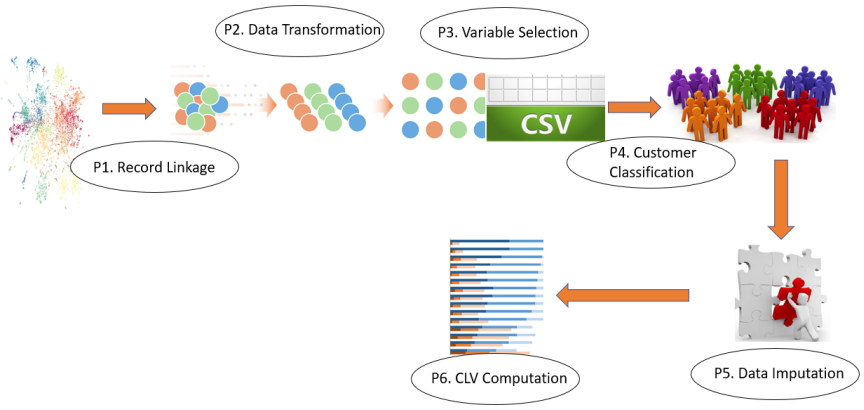

Fig. 1: ETL Pipeline Architecture

3 Data Transformations

In this section, we provide a brief outline of the Extract Transform Load (ETL) architecture but focus on those components which are novel to our architecture and crucial to imputing retention data. The dataset used in this work originated from our collaborator in the insurance sector. In this sector, transactions focused on selling policies and not on building customer profiles and thus, the Extract component in our architecture acquired approx. 500,000 insurance policies.

3.1 System Architecture

An ETL pipeline comprises a series of components extracting data from input sources, transforming the data to match the system's data model, and loading into a data mart (data cube) for reporting and analysis. Our approach, shown in figure 1, is a specialised form of ETL [14], due to the specific requirements of the task (customer lifetime value) and the nature of the data. In particular, this work began with a dataset that was policy-focused and not customer-focused. In effect, it was not suited to analysis by customer. Thus, the first step involved a process known as record linkage where, upon acquisition, data was pivoted to be customer-focused, where a customer record contained 1 or more policies. This work was presented in [10] and, while it provided a more holistic customer record, the data was not suited to the imputation algorithms necessary to impute the missing CLV variables. In addition, the dataset was still unclassified in terms of customer types (good, bad, average).

3.2 Churn Analysis Data Transformation

In this paper, we focus on components P2 and P5 from figure 1. The data used is initially based on two large imports: `detail` and `aggregate`. `Detail` pro-

vides a *policy centric* view year-on-year, recording the type, current and renewal premium for each policy. `Aggregate` is an aggregation of a unified customer record, detailing high level information on customers who hold policies generated during earlier work [10]. The large data imports are combined with other data sources within the warehouse to provide a dataset suitable for predicting *customer churn*. There are three processes involved in the transformation (P2) of a dataset suitable for churn analysis: `Aggregation`, `Augmentation` and `Preparation`. `Aggregation` constructs the initial per-policy view which provides information on policy renewals. `Augmentation` adds features to this dataset such as customer information and pricing. These two processes can be equated to the `E` and `T` processes within a standard `ETL` (Extract, Transform, Load) architecture. The final process `Preparation` provides a final transformation of the dataset so that is it ready for machine learning algorithms.

Aggregation. The goal of the first step is to construct a policy centric view containing those policies that may or may not been renewed. This involves a RollUp operation on the `detail` view to create an aggregated view containing the policy identifier (`policy_id`), the number of years for which the policy is held (`years_held`) and whether or not the policy was renewed (`renewed`). In total, the dataset used for our work contains 443,893 unique policies, of which 300,646 were not renewed with the remaining 143,247 renewed by the customer.

Augmentation. The next step is `augmentation` where views within the warehouse are integrated with the policy-centric aggregation. In total, seven additional views are integrated: policy prices, family policy holders, latest renewal premium, insurance type, location, payment method and gender. **Policy Prices** include the average premium and the standard deviation for premium, which can indicate the amount of variation in year-on-year premium prices. **Family Policy Holders** is the number of family members per customer who also hold policies with the company. **Latest Renewal Premium** is the latest premium for a given policy. **Insurance Type** is the type of insurance, which has four possible values: `Private Motor`, `Commercial Motor`, `Home` and `Travel`. **Location** is the county the customer resides in. **Payment Method** indicates if the premium is paid either in full or monthly. Finally, **Gender** relates to the gender of the policy holder. The result is a dataset with fourteen dimensions including a class label of *Renewed* or *Not Renewed* as seen in table 1 where: `Name` is the name of the feature; `Description` briefly describes the feature and `Type` indicates if *Categorical* or *Continuous*. For our evaluation, the dimensions representing unique identifiers (`pid` and `cid`) were not used.

Preparation. There are four steps in the preparation phase: *cleaning*, *sampling*, *encoding* and *splitting*. In this dataset, just 27 records were removed leaving a dataset with 443,866 rows. Determining whether or not a policy is renewed is, in effect, a classification problem. The class labels for each policy are `Renewed` or `Not Renewed`. As is common with real world data, our dataset

Table 1: Post-Integration Dataset Features

Name	Description	Type
pid	The policy identifier	Categorical
cid	The policy holder identifier	Categorical
years	The number of years the policy was held for	Continuous
avg_total	The average premium since first purchase	Continuous
std_total	The standard deviation of the premium	Continuous
family	The number of family members of the policy holder who also hold policies	Continuous
renew_p	The current renewal premium price	Continuous
total_p	The current premium price	Continuous
pol_type	The policy type (e.g. Home, Car, Travel..)	Categorical
pay_type	The payment type, either Partial (pays monthly) or Full (payment in full on purchase)	Categorical
gender	The gender of the policy holder	Categorical
county	The county of the policy holder	Categorical
province	The province of the policy holder	Categorical
Class	Value: Renewed or Not Renewed	Categorical

has a class imbalance where 300,621 records are labelled `"Renewed"` and the remaining 143,245 records are labelled `"Not Renewed"`. This class imbalance can greatly affect classification results and three methods are generally employed to resolve this: undersampling, oversampling and synthetic sampling. As we have a large number of records for the minority `Renewed` class, *undersampling* was the method selected to address this issue. Using this method, 143,245 records with the class label `Not Renewed` were randomly chosen so that both classes had the cardinality. The downside to this approach is that some of the `Not Renewed` data could have increased the effectiveness our analysis. This is addressed in our conclusions. After undersampling, the dataset comprised 286,490 records, with an equal distribution of the classes `renewed` and `not renewed` (143,245 records each). The encoding step transforms categorical dimensions so they are ready for machine learning algorithms. The dimensions encoded were `insurance_type`, `payment_method` and `county`. The final step splits the data into training and testing sets using the 80/20 configuration.

4 Algorithm Selection and Validation

Due to the characteristics of insurance data and the fact that research into customer lifetime value is quite theoretical in nature, it was decided to use a range of statistical methods and try to determine what works best. In this section, we begin by presenting the set of algorithms used to impute the retention value (churn), then proceed to discuss the evaluation strategy and results and finally, we present a discussion on the results.

Algorithm Selection. The process of determining customer churn in any domain is generally a classification problem with two classes: `Renewed` and `Not Renewed`. The classification methods we employed were: Bernoulli Naive Bayes; Multinomial Naive Bayes; two types of support vector machines; two decision trees; and a series of artificial neural network (ANN) configurations. Support Vector Machines are used regularly in classification. For our experiments we used one Linear Support Vector machine to provide a baseline to our other methods. Two experimental configurations using Naive Bayes were employed, one using a Bernoulli model and another using a multinomial model which has been shown to have increased performance on binary data in some instances [9]. For both models, 100 different alpha values were used on each, ranging from 0.0 to 1 in degrees of 0.01. Two decision trees using the CART (Classification and Regression) algorithm were employed, the first using `entropy` as the splitting measure and the second using Gini impurity. For both approaches, a decision tree was created for each level of depth until the maximum depth was reached and at each depth, test data was used to obtain the accuracy of the tree. Artificial Neural Networks (ANNs) have seen extensive use in predicting customer churn due to their ability to model interactions between features that may otherwise go unnoticed. For our experiments, 20 different configurations of ANNs were constructed with various configurations of hyperparameters.

Evaluation Metrics. We now describe the evaluation metrics used to compare the different prediction models. The measures `TP`, `TN`, `FP` and `FN` are *true positive*, *true negative*, *false positive* and *false negative* respectively. The metrics are: Accuracy, Precision, Recall, Specificity and F_1 score. Standard accuracy (percentage of correct classifications) is insufficient in evaluating a classifier but provides a useful baseline. The precision, recall and specificity metrics provide more information as to the actual performance of a classifier *within* classes. All model configurations will be validated using all 5 metrics.

Results and Discussion. We begin this section with an overview of the 4 different algorithms in isolation, reporting on their relative performances. We then take a comparative view across all algorithms, using different configurations for the more complex models. Unsurprisingly the Linear SVM show a weak performance with an accuracy of 0.754 and an F_1 score of 0.76. However, this model was always intended as a baseline for our evaluation of other models. For both Naive model types, 100 different alpha values were used, ranging from 0.0 to 1 in degrees of 0.01. Interestingly, these changes had no effect on the accuracy score across model configurations. For the algorithm which incorporated entropy, the best performing tree had a depth of 11 with an accuracy of 88.82%. For the Gini-tree, the best performing depth was also 11 and with a very similar accuracy of 88.72%. The results of the top 5 performing configurations can be seen in Table 2, where `id` is the experimental id; `epoch` is the number of epochs; `hlayer` is the number of hidden layers; `hnode` is the number of hidden nodes,; `tr_ac` is the accuracy of the training data,; `tr_l` is the loss of the training data; `te_ac` is the accuracy on the test data; and finally, `te_l` is the loss on the test data. For all 5

configurations, a dropout rate of 0.02 was used. From Table 2, experiment 5 is the best performing with an accuracy of 0.888 on the test dataset. This model consisted of one hidden layer with 31 hidden nodes with a dropout rate of 0.2%. There were other models with increased training accuracy `tr_ac` but are not shown as they have a lower `te_ac` than 88.6% which is generally an indication of over fitting.

Table 2: Configuration of the top five performing ANNs

id	epoch	hlayer	hnode	tr_ac	tr_l	te_ac	te_l
5	50	1	31	0.892	0.251	0.889	0.259
12	50	2	56	0.893	0.25	0.884	0.265
15	37	2	56	0.889	0.26	0.885	0.268
17	37	2	66	0.892	0.252	0.887	0.266
18	37	2	36	0.891	0.257	0.886	0.263

Discussion. Table 3 provides a comparison across all experimental model and parameter configurations. `Method` is the classification algorithm and configuration used; `acc` is the overall accuracy; `err` is the overall error; `pre` is the precision; `rec` is recall; `spe` is specificity; and F_1 is the F_1 score. Overall, most models performed well with 7 of the experiments achieving an accuracy > 0.88, with 6 of those having an F_1 score > 0.89. Interestingly, the difference between the two decision tree methods (Entropy & Gini split) was so small (> 5 decimal places) that they effectively performed the same. The worst performing method was the multinomial Naive Bayes with an accuracy of 0.69 and an F_1 score of 0.678. In terms of accuracy, the best performing model was `ANN-5` with an accuracy of 0.889. This configuration also achieved the highest F_1 score with 0.893. On the other hand, both decision tree methods have higher precision (0.938 vs 0.920) and specificity (0.931 vs 0.914) scores. However, `ANN-5` had a higher recall rate (0.867 vs 0.851). Between these three high performing configurations, `ANN-5` had the highest number of true negatives (24487) while both decision tree methods had a higher number of true positives (26766 and 26767). By examining the `NB-Bernoulli` model results, there is a clear requirement for more in-depth statistics than accuracy alone. This method has an overall accuracy of 0.776 but there is a difference between the measures for recall and specificity (0.717 and 0.879 respectively), indicating that this method is better at predicting negative classifications over positive ones. If we examine the ANN configurations, while `ANN-5` has the highest overall accuracy and F_1 score, other methods show a higher value for specificity (the highest being `ANN-17` with 0.929). However, `ANN-5` shows the highest value for recall out of all neural network configurations indicating that it performs best when predicting positives classes. A high recall value necessitates a low *false negative* rate. Out of all methods employed `ANN-5` has the lowest rate of `fn`, with $4,070$ records being classified incorrectly. The question of which model and configuration to use depends ultimately on

the classification most important to businesses. From our findings, a decision tree classifier is recommended for organisations that wish to obtain the highest number of correct *negative* classes, while `ANN-5` should be used if the goal is to correctly identify the highest number of *positive* classes. As the ultimate goal of our research is to impute `retention` for its use in CLV calculations, this current step sought to obtain the highest number of correct negative classes so the method `DT - Gini` is selected for CLV calculations.

Table 3: Comparison of classification methods

method	**acc**	**pre**	**rec**	**spe**	F_1	**tp**	**tn**	**fp**	**fn**
NB - Bernoulli	0.776	0.911	0.717	0.879	0.802	25999	18477	2551	10271
NB - Multinomal	0.691	0.651	0.706	0.679	0.678	18594	21023	9956	7725
DT - Entropy	**0.887**	**0.938**	**0.851**	**0.931**	**0.892**	**26766**	**24048**	**1784**	**4700**
DT - Gini	**0.887**	**0.938**	**0.851**	**0.931**	**0.892**	**26767**	**24055**	**1783**	**4693**
SVM - Linear	0.754	0.779	0.742	0.769	0.760	22234	20997	6316	7751
ANN - 5	**0.889**	**0.920**	**0.867**	**0.914**	**0.893**	**26449**	**24478**	**2301**	**4070**
ANN - 12	0.884	0.930	0.852	0.922	0.889	26730	23917	2020	4631
ANN - 15	0.885	0.924	0.858	0.917	0.890	26578	24133	2172	4415
ANN - 17	0.887	0.937	0.853	0.929	0.893	26934	23893	1816	4655
ANN - 18	0.886	0.919	0.863	0.912	0.890	26412	24370	2338	4178

5 Conclusions & Future Work

In this paper, we presented an approach to generating one of the key parameters for CLV calculations: retention, a process that requires a specialised form of ETL architecture and a degree of flexibility in selection of models used to impute retention. Our evaluation showed that an artificial neural network provided the highest accuracy and F_1 score followed closely by two decision tree algorithms. While no single model or model configuration achieved the highest score across all evaluation metrics, our approach identified the best model configuration depending on the user's prediction requirements. Our current work remains focused on the identification and selection of best model configuration. Recall that our under sampling method randomly chose a recordset that *may* exclude instance data suited to the predictions of retention. To address this, we are re-running all experiments using different datasets to determine if further improvements can be found in our results. Given the high scores achieved by the neural network models, we are also examining hyperparameter tuning to explore other avenues to improve the prediction accuracy.

References

1. et al., G.: Modelling and predicting customer churn from an insurance company. Scandinavian Actuarial Journal **2014**(1), 58–71 (2014)

2. Berger, P.D., Nasr, N.I.: Customer lifetime value: Marketing models and applications. Journal of interactive marketing **12**(1), 17–30 (1998)
3. Bolancé, C., Guillen, M., Padilla-Barreto, A.E.: Predicting probability of customer churn in insurance. In: Int Conf on Modeling and Simulation in Engineering, Economics and Management. pp. 82–91. Springer (2016)
4. Di Benedetto, C.A., Kim, K.H.: Customer equity and value management of global brands: Bridging theory and practice from financial and marketing perspectives. Journal of Business Research **69**(9), 3721–3724 (2016)
5. Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., Ravishanker, N., Sriram, S.: Modeling customer lifetime value. Journal of service research **9**(2), 139–155 (2006)
6. Hu, X., Yang, Y., Chen, L., Zhu, S.: Research on a customer churn combination prediction model based on decision tree and neural network. In: 5th Int Conf on Cloud Computing and Big Data Analytics (ICCCBDA). pp. 129–132 (2020)
7. Lemmens, A., Croux, C.: Bagging and boosting classification trees to predict churn. Journal of Marketing Research **43**(2), 276–286 (2006)
8. Ling, R., Yen, D.C.: Customer relationship management: An analysis framework and implementation strategies. Journal of Computer Information Systems **41**(3), 82–97 (2001)
9. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with naive bayes-which naive bayes? In: CEAS. vol. 17, pp. 28–69. Mountain View, CA (2006)
10. Nie, D., Roantree, M.: Detecting multi-relationship links in sparse datasets. In: ICEIS 2019, Volume 1. pp. 149–157. SciTePress (2019), https://doi.org/10.5220/0007696901490157
11. Reinartz, W.J., Kumar, V.: On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. Journal of marketing **64**(4), 17–35 (2000)
12. Risselada, H., Verhoef, P.C., Bijmolt, T.H.: Staying power of churn prediction models. Journal of Interactive Marketing **24**(3), 198–208 (2010)
13. Roantree, M., Liu, J.: A heuristic approach to selecting views for materialization. Softw. Pract. Exp. **44**(10), 1157–1179 (2014)
14. Scriney, M., McCarthy, S., McCarren, A., Cappellari, P., Roantree, M.: Automating data mart construction from semi-structured data sources. Comput. J. **62**(3), 394–413 (2019)
15. Sohrabi, B., Khanlari, A.: Customer lifetime value (clv) measurement based on rfm model. Iranian Accounting & Auditing Review **14**(47), 7–20 (2007)
16. Sundarkumar, G.G., Ravi, V., Siddeshwar, V.: One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection. In: 2015 IEEE Int Conf on Computational Intelligence and Computing Research (ICCIC). pp. 1–7. IEEE (2015)
17. Tamaddoni, A., Stakhovych, S., Ewing, M.: The impact of personalised incentives on the profitability of customer retention campaigns. Journal of Marketing Management pp. 1–21 (2017)
18. Ullah, I., Raza, B., Malik, A.K., Imran, M., Islam, S.U., Kim, S.W.: A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. IEEE Access **7**, 60134–60149 (2019)
19. Zhang, R., Li, W., Tan, W., Mo, T.: Deep and shallow model for insurance churn prediction service. In: 2017 IEEE Int Conf on Services Computing (SCC). pp. 346–353. IEEE (2017)