TO CHURN OR NOT TO CHURN?: THE KKBOX KAGGLE CHALLENGE

Project Report for IE-500 Data Mining
HWS 2017

Presented by Team 9:
Simona Doneva
Na Gong
Bengi Koseoglu
Siying Liu
Liam Pang

submitted to the
Data and Web Science Group
Prof. Dr. Heiko Paulheim
University of Mannheim

December 2017

1. Introduction

1.1 Application Area and Goals

KKBOX is a subscription-based music streaming service which operates in the Asia Market. Our project is to assist KKBOX by predicting customer churn as accurately as possible, as a part of Kaggle challenge. The main goal is to compare the evaluation metrics of the various classification algorithms, suggest the most suitable classifier(s) for the task and provide management advice based on the results. To do so, various classification techniques are applied on a pre-processed training dataset and an unadulterated test data set.

1.2. Structure and Size of the Data Set

Kaggle dataset consists of four subsets: members, train, transaction and user log.

Members: it describes profile information of 5,116,194 customers such as city, age, gender, registration method, registration date and expiration date.

Train: it provides churn behavior (is_churn) and customer id of 992,931 customers whose subscription is going to expire in February 2017.

Transaction: it records transaction details for total 21,547,746 customers. For each customer, it records his/her customer id, payment method, length of membership in terms of days, plan list price, actual amount paid and the choice of automatic renew service. All price are in Taiwan dollars.

User Log: it contains listening log of 392,106,500 customers. The log includes customer id, date and number of songs and total seconds played less than 25/ 75/ 98.5/ 100 percentage perspectively.

2. Preprocessing

2.1. Data Cleaning and Preparation

Data Aggregation: Two of the provided datasets are log datasets which consists of customer's daily usage of the service(user logs data set) and customer's monthly/weekly (transactions data set) activities regarding their subscription and payment, therefore these datasets are exceedingly large and have multiple rows per customer. First, in order to reduce the size of these log datasets, most recent customer activities of six months were taken and inner joined with training dataset individually. Secondly, in order to have a single object per user, we aggregated records while creating and deriving new variables, without losing valuable information during the process. For instance, we computed a new variable called 'total churn' from the existing entries about the payment activities of an user in transactions dataset, in order to determine the number of times an individual customer has churned in the past 6 months. We also derived new variables from user logs depending on day, such as the number of unique songs a customer has listened in the month he/she expected to churn. In the end of the data aggregation, we had 126 number of variables, excluding id,target and unnecessary variables.

Merging the datasets: We realized that not all the customers that are in the train dataset are in the member dataset. Therefore, first we decided to inner join members with train dataset. After also joining member and transactions datasets, we left joined user log and transaction datasets to the merged member and transaction dataset. As a result, we had a dataset 876143 rows with 132 columns.

Data Cleaning: We applied different missing handling methodologies, according to the variable. Variables created from the user logs dataset had missing values after join, due to the fact that we focused on only the last six

months of the data and a customer might have not used the service in the past half year. We decided to fill those missing values with 0. Meanwhile, the unmatched customer volume in data files, caused partial null value in transactional variables after the three files were merged. We filled with the most frequent value the categorical attributes and the remaining numeric attributes with the mean value. Additionally, gender variable in the members dataset had more than 50% missing values. Since we considered it important for the churn prediction, we decided to treat the missing gender variables as third category. Furthermore abnormal values such as age having values as high as 2000 and negative values in active days were observed. In order to minimize their interference on prediction and keep original data structure, we set abnormal value with the mean value and negative values to 0.

2.2. Variable Transformation and Sampling

Standardization: The approach for our decision on the best standardization technique was as follows:

1. Dummy encode categorical variables	
2. Train-test-split	
3. Standardize train set	
3.1. Standard Scaler Approach	3.2. Robust Scaler Approach
(1) Handle Outliers with Isolation Forest	(1) No need to explicitly handle outliers [1]
(2) Apply Standard Scaler	(2) Apply Robust Scaler
(3) Run ANN/ Decision Tree in stratified k-fold manner Cross-val score ANN: 91.89 Cross-val score DT: 89.66	(3) Run ANN/ Decision Tree in stratified k-fold manner Cross-val score ANN: 90.82 Cross-val score DT: 88.59
4. Chose scaler based on models performance	
5. Apply the scaler to test set	

Table 2.1: Standard Scaler vs Robust Scaler

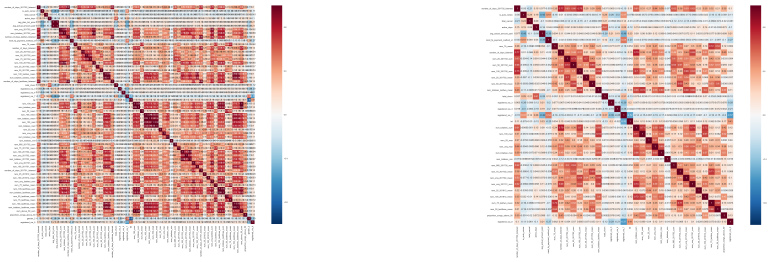
The comparison of the cross validation score clearly showed that our two algorithms performed better when applied to the dataset standardized with Standard Scaler and therefore our further modeling was applied to those datasets.

Undersampling: To handle the fact that our data set has highly imbalanced class distribution, we used a focused under-sampling technique and eliminated negative examples until the number of negative and positive examples were equal.

2.3. Feature Subset Selection

Feature Selection Algorithms: Using large number of potential inputs to the classification algorithm may hinder the efficiency of the algorithm[2]. Therefore feature selection and dimension reduction played an important role in our modeling process. First, we picked top 20 attributes selected by univariate feature selection methods such as Mutual Information and F-Classification, as well as Recursive Feature Elimination, Random Forest, and Principal Component Analysis.
To prepare the final set of features, the results from those algorithms were tracked via an Excel workbook, which can be found in the files accompanying the report. The results were tabulated based on the number of times the individual variable was selected in order to generate an initial feature relevancy ranking. 59 variables were ultimately selected with the minimum required score of 1 out of 5.

Correlation matrix for further feature refinement: The feature selection methods that we used above do not consider collinearity between variables, as a result chosen variables from the feature selection algorithms, may have correlations. Inclusion of correlated variables may lead to the instability of the model results [3], therefore in order to handle multicollinearity and to reduce the number of variables further, the pairwise correlations of the chosen variables from the previous step were examined. Among the variables that had correlations more than 0.8, the one that was selected by more models was chosen whereas the other variable was excluded. As a result, the number of variables were reduced further from 59 to 35. Below one can see the heat maps of chosen variables, before and after collinearity.



(a) Heat Map Before (b) Heat Map After

Figure 2.1: Heat Maps of Selected Variables

3. Data Mining

In order to achieve churn subscriber prediction, several models were trained by applying the following classifier models. We first chose a best model from each classifier, with specific parameter setting (see Table 4.2) and then conducted a performance evaluation among different algorithms in section of result analysis.

3.1. K-Nearest Neighbor

K- Nearest Neighbor algorithm also known as KNN, is one of the most simplest and oldest methods and it is considered as one of the top 10 methods in data mining[4]. In KNN, we focused on three main things; choosing the appropriate algorithm for measuring distance, selecting the best features for KNN and finding the optimal number of k neighbors. For measuring the algorithm, first three models are built with Euclidean distance whereas the fourth model is built with Manhattan distance. In order to find the best features and k values of KNN, models are built using different k values ranging from 1 to 10 as a rule of thumb, and with different combination of variables. The model that was built using RFE variables using manhattan distance and 9k's is chosen as the best model due to having the lowest log loss score.

3.2. Logistic Regression

Logistic Regression (LogR) suits particularly well in predictions regarding customer's behavior about whether they are going to re-purchase a product, remain a customer, respond to a direct email or other marketing stimulus[5]. For this purpose, 8 different models were built using sklearn Logistic Regression function, with different metrics and variables. Models 1 through 7 are built using default settings, where penalty is l2 and C is 1.0, whereas the model 8 is built with selected parameters that minimizes log loss using

Grid Search, where C is 10 and penalty is l2. Among the tried combination of variables, variables chosen by RFE feature selection algorithm that were used in model 7 and model 8 yielded best outcome. Even though, both model 7 and model 8 are built using RFE variables, increasing the C value from 1 to 10 in model 8, gave the model more freedom, therefore the final model resulted better than the model 7. As a result model 8 was chosen as the best model due to having lowest log-loss of 0,26.

3.3. Decision Tree

Decision Tree (DT) is an efficient non-probabilistic supervised learning method for data mining which is widely applied in different industries[6]. This study adopted a sklearn Decision Tree Classifier with CART algorithm to predict churn customer. Based on feature results computed by different feature in section 3.3, total 10 tree models were trained with different variables. During construction, there are two important parameter optimizations. First is to use GridSearch algorithm to find the best combination of top dominate parameters in decision model (see Table.4.2). Second is to remove pure nodes so that the tree generate a better Log loss and AUC [7]. According test, 0.000015 of *min impurity decrease* decrease Log loss by a half without sacrificing prediction accuracy. As a result, the model trained on features selected by RFE algorithm has best performance which has the lowest log loss and highest AUC score without apparent sacrifice in precision.

3.4. Artificial Neural Network

The Keras API [8] was used to utilize a neural network model for our problem. The chosen loss function was binary crossentropy. Tested with a baseline model, it showed that reducing the dimensionality to the 35 attributes chosen after removing correlation had the most positive impact on the performance and therefore this features subset was maintained during the further stages of parameter tuning. Because the networks can be very slow to train, a 0.1 sample from the train dataset was taken for the next steps. Firstly, the optimal number of neurons (50) was determined using scikit-learn Grid Search, which was also used to find the batch size (40) and the optimal number of epochs (30). After that “Adam” [9] was chosen as the best performing optimization function and ”softmax” as the best activation function. Adding a hidden layer led to no performance improvement in this setup. Scaling to the whole dataset however did not show better results, so that the baseline model was kept as optimal.

3.5. Naive Bayes

Naïve Bayes (NB) is a probabilistic supervised classifier which is based on Bayes' Theorem and assume all features are independent from each other. Among three Naive Bayes algorithms, Gaussian Naive Bayes algorithm is chosen due to the fact that given dataset includes variables of continuous and binary type. Five different models are built by using: all the features from the feature selection algorithms after correlation, features that are chosen by 4 different feature selection algorithms, features which appear in at least 3 feature selection algorithms, features which appear in at least 2 feature selection algorithms, and feature which appear at most 2 feature selection algorithms. Among the built models, model 2 is chosen as the best model of Naive Bayes due to having the lowest log loss of 0,63.

3.6. Nearest Centroid

With Nearest Centroid (NC) algorithm, eight different models are developed by using different features with distance metric being Euclidean. These eight models include modeling with all selected variables, variables chosen by 4 different feature selection algorithms, variables chosen by more than 2 different feature selection algorithms and variables chosen by the following feature selection algorithms; F classification, PCA, Mutual Info, RFE and Random Forrest. Nearest Centroid assigns a class to each observation, but not a probability of class, therefore calculation of log loss and plotting ROC Curve is not possible for Nearest Centroid. As a result, the best nearest centroid is chosen by f1 score and recall. Among the models, nearest centroid with 4 ranked features are chosen with the f1 score of 0,50.

3.7. Support Vector Machine

Support Vector Machine (SVM) is another efficient supervised learning algorithm basically designed for binary classification. Four models are constructed by using sklearn C Support Vector Classification with different input variables. Due to the large time complexity, combination of variables only showing good performance in decision tree models are chosen. In order to obtain the best prediction outcome, the cross-validation based GridSearch algorithm was used to select the optimal parameter values for SVM. As a result, the best performance in terms of lowest log loss and highest recall score in class 1, was achieved while training the model with the specified parameters in Table 4.2 and using features selected by Mutual Information algorithms.

4. Evaluation and Conclusion

4.1. Evaluation Metrics

As main success indicator we considered the one provided by the organizers of the competition- the Logarithmic Log Loss metric.
By doing so, we could not only compare the performance of our models against the ones built by other participants, but also had a reliable metric, which heavily penalized the model for making high-confidence incorrect classification.
Further evaluation was based on both average and class 1-specific Precision, Recall, and F1 score. Due to the fact that our experimental dataset is highly imbalanced, the evaluation metric of Accuracy was not used in this study. Instead, a more robust metric- Area under the curve (AUC) was adopted.

4.2. Result Analysis

MODEL	***Parameter Setting***
K-NN	n neighbors=*9*, weights=*'uniform'*, algorithm=*'auto'*, leaf size=*30*, p=*1*, metric=*'minkowski'*, metric params=*None*, n jobs=*None*
LogR	penalty=*'12'*, dual=*False*, tol=*0.0001*, C=*10*,fit intercept=*True*, intercept scaling=*1*, class weight=*None*, random state=*None*, solver=*'liblinear'*, max iter=*100*, multi class=*'ovr'*, verbose=*0*, warm start=*False*, n jobs=*1*
DT	criterion=*'entropy'*, min impurity decrease=*0.000015*, max depth=*15*, max leaf nodes=*None*, min samples leaf=*10*, min samples split=*20*.
ANN	perceptron, hidden units=*100*, input dimensions=*35*, batch size=*16*, epochs=*10* activation function=*'relu'/'sigmoid'*, loss function=*'binary crossentropy'*, optimizer=*'adam'*
NB	prior=*None*
NC	metric=*'euclidean'*, shrink threshold=*None*
SVM	kernel=*'rbf'*, C=*1*, gamma=*0.1*, probability=*True*

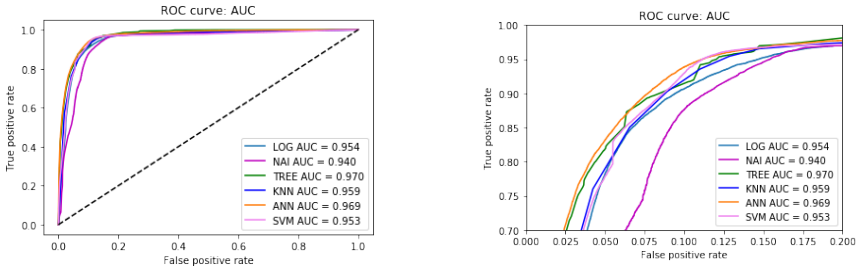
Table 4.1: Optimal parameter setting for each model.

Based on efficiency metrics obtained from best models computed by different classifiers with above specific parameter setting, a statistical performance comparison is visualized in Table 4.2 with a ROC curve in Figure.4.1.

EVALUATION	K-NN	NC	LogR	DT	ANN	NB	SVM
AUC	0.96	-	0.95	0.97	0.97	0.94	0.95
Log loss	0.82	-	0.27	0.20	0.21	0.63	0.26
Precision-avg/total	0.95	0.95	0.95	0.95	0.96	0.95	0.95
Recall-avg/total	0.89	0.89	0.89	0.90	0.91	0.84	0.89
F1 Score-avg/total	0.91	0.91	0.91	0.92	0.92	0.87	0.91
Precision-class 1	0.37	0.36	0.37	0.39	0.41	0.27	0.37
Recall-class 1	0.89	0.86	0.92	0.92	0.93	0.97	0.95
F1 Score-class 1	0.53	0.50	0.53	0.54	0.57	0.43	0.53

Table 4.2: Performance evaluation of different models.

According comparison, following results can be induced:
i. Combined with ROC curve in Figure.4.1(a), all models have good AUC score which are greater than 0.90. Markedly, AUC of both DT and ANN is up to 0.97 which slightly higher than others.



(a) ROC Curve overview

(b) ROC Curve-Zoomed

Figure 4.1: ROC Curves

ii. Benefit from the controlling of pure node, DT stands out in log loss comparison. ANN also has a competitive log loss. While NB presents a most expensive cost in log loss which is even three times than decision tree and ANN. This might be affected by imbalanced test dataset.
iii. From the general weighted classification report, average Precision score of all models present an outstanding performance which is generally above 0.95. Although the general Recall score goes slightly down to about 0.89, the overall F1 score around 0.90 is still acceptable.
iv. When comes to classification report of Class 1 which is 'churn' prediction on customer class, the results show an opposite tendency with iii. Precision score-class1 of all models dramatically decrease to around 0.37. This outcome is foreseeable because the class of interest in our case only occupies 6.5% of total test data. This significant imbalance problem makes it hard to recognize the minority class instance and easily generate large False Positive

(FP) prediction [10]. On the other hand, this also means the Precision of Class 1 prediction is highly possible to be improved by applying in balanced or small dataset.
v. From business perspective, service company would focus more on accurate prediction on real churn subscriber so that they could optimize strategy and resource allocation on CRM [11]. The Recall of churn customer prediction means more for commercial. It is can be seen from the table, Recall of Class 1 of mostly models are higher than general recall score. Particularly, NB has optimal prediction on real churn customer and the score even reaches 0.96. This is might promoted by the efficient correlation-based feature selection. Meanwhile, SVM model also has an outstanding performance. ANN and DT also works well.

4.3. Conclusion

In this study, churn prediction was successfully achieved by conducting KNN, NC, LogR, DT, ANN, NB and SVM classification models on real transaction dataset issued by KKBox company. In terms of a comprehensive performance evaluation, it has been found that both DT and ANN models have outstanding performance in churn prediction, with DT having lowest log loss. In addition to time complexity and transparent structure, Decision Tree is regarded as the best model in this concept of study.
The current research individually implemented and compared different classification models in churn perdition. In future studies, it is possible to explore a hybrid model by integrating strength of different models and further improve precision of churn prediction.

Bibliography

[1] Scikit learn Documentation. (2017).Retrieved November 18, 2017, from http://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

[2] Chorianopoulos, A. (2016). Effective CRM using predictive analytics. Retrieved November 28, 2017.

[3] Larose, D. T., Larose, C. D. (2014). Discovering Knowledge in Data: An Introduction to Data Mining (2nd ed.). doi:10.1002/9781118874059

[4] Wu, X.a.V.K.e., 2010. The top ten algorithms in data mining. CRC Press.

[5] Karp, A. H. (2014, January 6). Using Logistic Regression to Predict Customer Retention. Retrieved November 29, 2017, from http://www.lexjansen.com/nesug/nesug98/solu/p095.pdf

[6] Luo, B., Shao, P.J., Liu, J. (2007). Customer Churn Prediction Based on the Decision Tree in Personal Handphone System Service. Retrieved November 29, 2017, from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4280145

[7] ManzaliY., Chahhou, M., Mohajir, M. E. (2017). Impure Decision Trees for Auc and Log loss Optimization. Retrieved November 29, 2017.

[8] Keras Documentation. (2017). Retrieved November 20, 2017, from https://keras.io/getting-started/sequential-model-guide/

[9] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." (2014). Retrieved November 21None, 2017, from https://arxiv.org/pdf/1412.6980v8.pdf

[10] Mutanen, T. (2006). Customer Churn Analysis – A Case Study. Retrieved November 30, 2017, from http://www.vtt.fi/inf/julkaisut/muut/2006/customer_churn_case_study.pdf

[11] Anjum, Adnan., Usman, Saeeda., Zeb, Adnan., et al. (2017). Optimizing Coverage of Churn Prediction in Telecommunication Industry. Retrieved November 30, 2017, from https://thesai.org/Downloads/Volume8No5/Paper_23-Optimizing_Coverage_of_Churn_Prediction.pdf