Churn Prediction for KKBOX Music Streaming Service

Team 8:
Chun-Yi Yang(N18995303)
Hung-Wei Chen(N13286118)
Sijun Dou(N16590012)

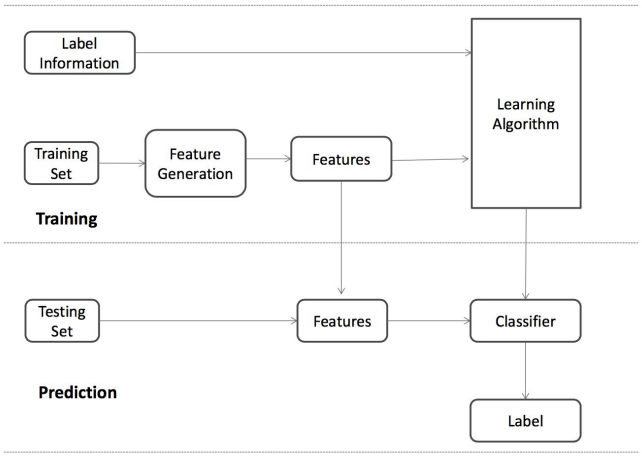
Abstract

In this project, we analyzed the the data from KKbox and predicted whether an user will churn after a short period (i.e. 30 days) from the expiration of the current service subscription. We pre-processed the data and uncovered predictive features from it. Then we applied some of the classification algorithms to train the models, evaluated the performance and analyzed their discrepancies. By tuning these models, we made best prediction by Decision Tree model and got high prediction accuracy at 95.2%.

Introduction

There has been a steady trend in using data mining and machine learning techniques to predict user behavior by customers' personal information and log records. Especially for subscribed-based companies such as telecommunication companies and online music service providers, it will be valuable to know whether the customers will continue to subscribe their services or not and take reaction in advance. Apple music has almost 3 times higher cancellation rate than its biggest competitor Spotify [1]. We are particularly interested in what and how do the customers' listening habit relate to their subscription decision. Therefore, we choose KKBox, an Asian based streaming music provider, as our analysis target. This project intends to design new predictive analytic methods that provide insights by using the user profile data such as membership info, transaction info and listening behaviors.

- Framework



Data Understanding

Dataset

The data is from kaggle project, WSDM - KKBox's Churn Prediction Challenge (https://www.kaggle.com/c/kkbox-churn-prediction-challenge/data). The dataset contains about 5 million members data, 21 million transaction data and more than 0.4 billion user log from February 2017 to April 2017. Total size of data is 32.7 GB. Members data includes attributes such as *city*, *age*, and *gender*; transactions such as *payment_method, is_auto_renew, transaction_data, membership_expire_date, and is_cancel* are also included. The user_logs contains interesting user behaviors such as the sum up of time user listening to music daily and

how long does the user finish each song by percentage in different categories (e.g. 25%, 50%, 75%, 98.5%, 100%).

Training data and testing data consist of users whose subscription expires in February 2017 and March 2017 respectively. It means that we are looking at user churn or renewal roughly in the interval of March 2017 for train set, and the same for test set in April 2017. The train and test sets are split by *transaction_date* attribute.

The data we got is very imbalance, most of users are "not churn" (93.6% v.s. 6.4%). This situation will cause accuracy paradox. Also, more than half of gender attributes are missing value, indicating that most of customers are not rather to inform their gender.

Assumption

The problem assumes that as the subscription is about to expire, the user can choose to renew or cancel the service. They also have the option to auto-renew but can still cancel their membership at any time.

Data Preprocessing

Before proceeding our analysis, we clean our data to remove obvious outliers. The modification of instance is to keep user who are in reasonable range ($10 < age < 100$). Moreover, there are almost no missing value in our dataset except age. Since more than half of age attribute are missing, it is useless in our analysis.

The format of input data is also influential to analysis result. It cannot reflect the true information if the format is incorrect. To handling the imbalanced data, we randomly select 57,146 un-churn data to align the other 57,146 churn data. To better apply the model on our dataset. We normalized the numeric data into scale between 0 and 1 and all the categorical data are implemented by vector technique in binary form.

Feature Selection and Extraction

Feature extraction affects prediction accuracy to a large extent, even a good algorithm cannot get high performance without appropriate features. We think the original features in the dataset, such as *city, age, is_auto_renew and is_cancel*, are relevant to whether churn or not. On the other hand, based on the research of Kristof and Dirk[2], they found that some subscription behaviors are strong drive to churn prediction such as the_length_of_current_subscription, elapsed_time_since_last_renewal, elapsed_time_since_last_suspension and days_of_renewal

_before_expiration. More than this, we also derived the average and standard deviations of these new features just mentioned. There features selected and generated are list below:

Variables Name	Type	Description
City	Categorical	22 cities
Age	Numerical	10 -100
Register_via	Categorical	19 register channels
Payment_method	Categorical	41 payment methods
Payment_plan_days	Numerical	0 -50
Plan_list_price	Numerical	0 -2000
Is_auto_renew	Categorical	binary
Is_cancel	Categorical	binary
Month_of_subscription	Categorical	12 months
Elapsed_days_since_last renewal	Numerical	The days between renewal and current subscription
Elapsed_days_since_last renewal_avg	Numerical	The average of days between renewal and subscriptions
Elapsed_days_since_last renewal_std	Numerical	The standard deviation of days between renewal and subscriptions
Elapsed days since last suspension	Numerical	The days between last suspension and current subscription
Elapsed days since last suspension_avg	Numerical	The average of days between suspensions and subscriptions
Elapsed days since last suspension_std	Numerical	The standard deviation of days between suspensions and subscriptions
Days_of_renewal before_expiration	Numerical	The days of renewal before expiration
Days_of_renewal before_expiration_avg	Numerical	The average of days of renewal before expiration
Days_of_renewal before_expiration_std	Numerical	The standard deviation of days of renewal before expiration
Avg_num_25	Numerical	The average of 25% listening interval songs
Avg_num_50	Numerical	The average of 50% listening interval songs

Avg_num_75	Numerical	The average of 75% listening interval songs
Avg_num_985	Numerical	The average of 985% listening interval songs
Avg_num_100	Numerical	The average of 100% listening interval songs
Avg_num_unique	Numerical	The average of unique songs
Avg_total_second	Numerical	The average of listening seconds

Classification Algorithms

To address our binary classification problem, we explored state-of-the-art algorithms, especially decision tree, tree ensembles (e.g. Random Forest) and Support Vector Machine (SVM). Each of the algorithms has its own characteristics.

Decision Tree

Decision Tree is one of the simplest classification algorithm. It involves constructing and pruning the tree, and then uses the tree to make predictions. The root can be selected based on partition criteria (e.g. Information Gain), and then recursively select sub-trees based on the same rule, until there is no attributes for further partitioning.

However, decision tree partitions the tree greedily at each node, which would lead to overfitting. One solution to this issue is using bagging, and Random Forest algorithm [4] uses this technique.

Random Forest

Random forest is one kind of tree ensemble methods. Instead of boosting which uses additive training, the trees in Random Forest are trained using bagging. The method of training the trees is also the key difference of tree bagging and boosting algorithms.

The forest grows many decision trees, each tree gives a classification of the input data. Finally the forest chooses the classification having the most votes. There are two keys: bagging and decision tree (acts as base classifier) construction. Bagging (aka. Bootstrapping) is used for row (data) and column (feature) sampling to train the trees. This prevents overfitting. The steps of growing a tree in the forest are:

- Sample N data objects from N input data objects with replacement, and this will be the training set of growing a tree.

- Sample m feature columns from M input features (m << M) without replacement, and the best split on these m features is used to split the nodes. Information gain can be used as one method of splitting.
- Each tree is grown to the largest possible extent.

There are two factors of forest error rate: 1) correlation between any two trees, which has positive effect on error rate; 2) the strength of individual trees, which has negative effect of error rates.

The significant advantage of Random Forest is that it does not overfit, because bootstrapping provides with internal unbiased estimate of true error. Besides, it runs efficiently on large data bases, handles thouthands of features without selection, and also gives the importance of features. But it needs enough trees to guarantee the stabilization and accuracy.

SVM

SVM is one of the most popular classification algorithm, because its capability of generalization. It achieves relatively low generalization error with various input data distribution. That is, SVM gives high accuracy without strict requirement on input data distribution.

Performance Evaluation

We tried three methods: Decision Tree, Random Forest and SVM. The testing accuracies are below:

1) Decision Tree

Testing Accuracy: 95.16%

2) Random Forest

Testing Accuracy: 90.59%

3) SVM

Testing Accuracy: 92.83%

Conclusion and Further Work

In this churn prediction competition, we surveyed a lot of methods, like Decision Tree, Random Forest, SVM etc and tried a lot of different features to increase the prediction accuracy. Those features are not easy access, we checked the data many times and tried to use feature

engineering method to extract useful features to train. And it proves that everything is worth it, those extracted features put our prediction to an impressive accuracy.

In addition to feature extraction, we also spend a lot of time to implement our system. Since our dataset is quite huge(about 50GB), we can't use simple software like RapidMiner to help us to easily process the data and get the results. We need to use Large Scale data processing framework like MapReduce or Spark to help us to deal with massive data. We choose Spark because it's comparatively easy to use and the code is succinct which is easy to maintain. And for hardware, we run our program on NYU's HPC - Dumbo. We also modified a lot of Spark's setting to let our program run successfully because our dataset is too large, we must use special configuration to make resources be balanced.

Last, in this project we really learn a lot. Whole process let us build everything from scratch so we can understand each steps' details. We also faced some practical questions which let us understand how to handle those practical issues which really benefit us a lot.

Reference

[1] https://www.spotify.com/us/

[2] Coussement, Kristof, and Dirk Van den Poel. "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques." Expert systems with applications 34.1 (2008): 313-327.

[3] J. Zhang, J. Fu, C. Zhang, X. Ke and Z. Hu, "Not Too Late to Identify Potential Churners: Early Churn Prediction in Telecommunication Industry," *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*, Shanghai, 2016, pp. 194-199.

[4] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr

[5] https://spark.apache.org/docs/latest/quick-start.html

[6] https://www.kaggle.com/c/kkbox-churn-prediction-challenge

[7] N. Lu, H. Lin, J. Lu and G. Zhang, "A Customer Churn Prediction Model in Telecom Industry Using Boosting," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1659-1665, May 2014.

[8] https://mapr.com/blog/churn-prediction-sparkml/

[9] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, K.Ch. Chatzisavvas, A comparison of machine learning techniques for customer churn prediction, In Simulation Modelling Practice and Theory, Volume 55, 2015, Pages 1-9.

[10] Kristof Coussement, Dirk Van den Poel, Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques, In Expert Systems with Applications, Volume 34, Issue 1, 2008, Pages 313-327.

[11] Xiaobing Yu, Shunsheng Guo, Jun Guo, Xiaorong Huang, An extended support vector machine forecasting framework for customer churn in e-commerce, In Expert Systems with Applications, Volume 38, Issue 3, 2011, Pages 1425-1430.

[12] Guo-en XIA, Wei-dong JIN, Model of Customer Churn Prediction on Support Vector Machine, In Systems Engineering - Theory & Practice, Volume 28, Issue 1, 2008, Pages 71-77.