KKBox's User Churn Prediction

Yujia Wang, Liyi Cao, Tong Ye, Weixuan Jiang

yujia8@bu.edu, caoliyi@bu.edu , tongyewh@bu.edu, jwx0728@bu.edu

1. Project Task

KKBOX is an Asia's leading music streaming service and it holds tens of millions of Asia-Pop musics and since it provides services to millions of people, they need a model on accurately predicting churn of their paid users.

So our input is the data provided by Kaggle, and the task is to build an algorithm that predicts whether users will be lost after the subscription expires. And to analyze the reason for the user leaving in order to be proactive in keeping users by using different methods.

2. Related Work

Currently, the company uses survival analysis techniques to determine the residual membership life time for each subscriber. Survival analysis is a statistical method that considers the results and considers the survival time. It can make full use of the incomplete information provided by the censored data to describe the distribution characteristics of survival time and analyze the main factors affecting the survival time.

3. Approach

1.Logistic Regression: Logistic Regression is a basic classification method taught in class. Basically, it uses a function to model a binary variable.

$$g(z) = \frac{1}{1+e^{-z}} \qquad h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

And when $h_\theta(x)$> 0.5, we predict it as 1;else we predict it as 0.

2.Neural Network: Neural networks are a set of artificial neurons like the human brain, which are designed to recognize patterns. They interpret sensory data through Neural network, labeling or clustering raw input.

3.LightGBM: LightGBM contains two key points: light is lightweight, GBM gradient hoist**.** It is a gradient boosting framework that uses a decision tree based on learning algorithms. It can be said to be a distributed, efficient machine learning algorithm.

4. Dataset and Metric

1.Dataset

train.csv & train_v2.csv contains the IDs and whether these users have churned or not.
train Shape= 992931 rows × 2 columns.
train_2 Shape = 970960 rows × 2 columns.

transactions.csv & transactions_v2.csv gives us details like *payment method* or whether the subscription was cancelled.
Shape = 1431009 rows × 9 columns.

user_logs.csv contains the listening behaviour of a user in terms of number of songs played.
Shape = about 200000000 rows × 9 columns.

members_v3.csv includes the user's age, city, and such for users that have these membership information.
Shape = 6769473 rows × 6 columns.

sample_submission_v2.csv serves as the *test* data set for the users for which we are tasked to predict their behaviour. Shape = 907471 rows × 2 columns.

From the last data analysis report, we can get that all the data features in user_log.cvs are not useful features. We will only use the number of times users appear. Also, the user's age, city, bd in members.cvs are not a important features. Based on our analysis of the data, we train by analyzing user behavior who are in train.cvs, transactions.cvs and members.cvs. And use train_v2.cvs as training label. The files transcations_v2.csv is the test set, and use the prediction result compare with sample_submission to calculate the error.

2.Metric:

To evaluation metric, we use Log Loss

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$$

At this time, yi is the real category of the input xi, and pi is the probability that the predicted input xi belongs to category 1. The log loss for a perfect classifier is 0.

Eventually we will upload our predicted probabilistic results(pi) on kaggle. The system will automatically score for us.

5. Preliminary Results

1.*Memory Reduction*

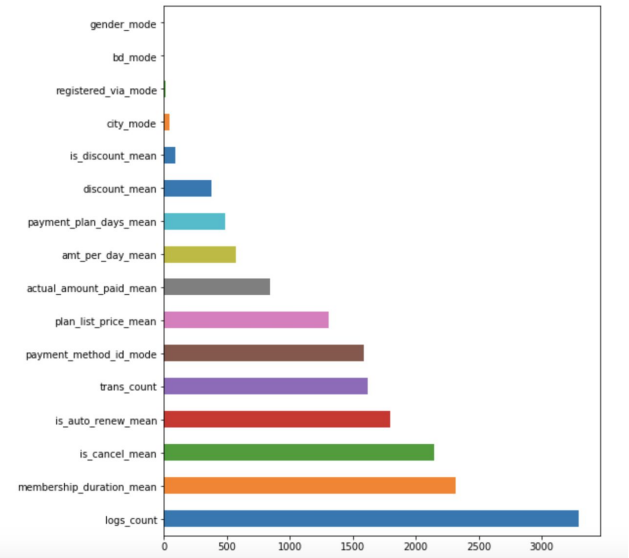
The memory consumed by each files are too high. We reduced the memory of data by converting variable type. Memory can be reduced for columns having values of type integer or float. We have to go through each column and find maximum and minimum value of data length. According to the data length of each line, we converted data type from int64 to int8, int16, int32 and int64. For example, converting the range of length of data from -128 to 127 to the type of int8. In the same way, we converted all float64 types to float32. After this process we have already reduce around 50%.

2.*Feature*

Last time we talked about feature selection and we used 'sum of plan days', 'sum_paid', 'amount_per_day' for the new features. But for this time, we found some other useful feat. We no longer use the 'sum of plan days', 'sum_paid' and 'amount_per_day', but we use the mean of these three features. The reason is typically we are using the data in transactions_v1.csv to train, and use the data in transactions_v2.csv to test. And since there will be more rows for one user in transactions_v1 than in transaction_v2, we need to scale them into the same scale to make sure the testing step is good.

So now we are using some different features, they are:
'payment_method_id_mode',
'payment_plan_days_mean',
'plan_list_price_mean',
'actual_amount_paid_mean',
'is_auto_renew_mean',
'is_cancel_mean',
'discount_mean',
'is_discount_mean',
'membership_duration_mean',
'amt_per_day_mean',
'city_mode',
'bd_mode',
'gender_mode',
'Registered_via_mode'.

Then we use LightGBM to train and predict the data. The detail will illustrated below. And in addition, we explored the feature importances for every feature.



So 'gender', 'bd', 'registered_via' are not important features, just as what we discovered last time!

3.Hyperparameter Optimization

For higher accuracy, we used one strategy called Hyperparameter Optimization. This method is advanced version of normal methods like grid search. And Bayesian Optimization is one common strategy which builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set.
For simple using, we used a python package called *skopt* to tuning the hyperparameters. And its principle is Bayesian Optimization. We chose a scale of 'learning rate' and 'n_estimators' and simply put them into this API, and run it.
Finally we found that the iterated values of 'learning rate' and 'n_estimators' are

```
x_iters: [[0.003778533854675O553, 930], [0.004573836852868271, 931], [0.003870691090357917 2, 723], [0.002892603
8196334172, 576], [0.002817968883740339 7, 765], [0.00443650618632648, 766], [0.003178354388302 4895, 926], [0.00301218
84812518056, 842], [0.003104724619521644, 981], [0.002421052341237 9354, 942]]
```

And the optimized one is

```
x: [0.004573836852868271, 931]
```

So we'll probably choose these hyperparameters first. However, maybe we'll choose some other kinds of parameters and get the optimized combination.

4.LightGBM

Using ensemble learning method is popular in classification problems. For this project, we choose LightGBM to increase the performance of our model.

Light GBM is a gradient boosting framework that uses tree based learning algorithm. It is specified as light because it takes lower memory to run on the large data.

Different from other tree models, LGBM uses level-wise tree growth which will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm and save the memory. Meantime, it increase the complexity of the model which is the main factor in achieving higher accuracy. However, it can sometimes lead to overfitting which can be avoided by setting the max_depth parameter.

Light GBM use histogram based algorithm that buckets continuous feature values into discrete bins to fasten the training procedure.

5.Compare LightGBM and Logistic Regression

One disadvantage of Logistic Regression(LR) is that LR is hard to deal with data skew situation. And in our project, the label of is_churn occupy around 92%. So, the result of using LR to be our classifier is not ideal.

Also, the training speed of Logistic Regression is limited because the activation function is sigmoid function. However, LightGBM use histogram algorithm, which shorten learning time cost.

And, the accuracy of LightGBM is higher than Logistic Regression.

Pic1. The score of log-loss by using LR

Pic2. The score of log-loss by using LightGBM

Actually, the score on Kaggle seems not bad, it's almost top 15% on the leaderboard. We'll try some other parameters and implement some other machine learning method in the next time, to improve our score.

6. Timeline and Roles

Task	Deadline	Lead
Data collection and cleaning	11/04/18	Liyi Cao
Implement logistic regression	11/14/18	Weixuan Jiang
Implement LightGBM	11/24/18	Tong Ye
Implement Neural Network	12/04/18	Yujia Wang
Prepare report and presentation	12/11/18	all

References

1) K. Gregor, I. Danihelka, A. Graves, D. Jimenez Rezende, D. Wierstra. DRAW: A Recurrent Neural Network For Image Generation, arXiv.org, 2015.
2) G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.
3) S. Hochreiter and J. Schmidhuber, Long short term memory. Neural computation, 9(8):1735–1780,1997.