KKBox's User Stickiness Analysis

Liyi Cao, Weixuan Jiang, Yujia Wang, Tong Ye

Boston University

Electrical and Computer Engineer Department

Introduction

KKBOX is an Asia's leading music streaming service and it holds tens of millions of Asia-Pop music's and since it provides services to millions of people, they need a model on accurately predicting churn of their paid user.
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 method.
In our project, through data analysis and feature selection, we finally chose to use logistic regression, random forest, neural networks and LightGBM methods to build different models and compare the advantages and disadvantages of these models.

Objective

This project tests whether users are lost after the member expires through different machine learning models. We also obtain parameters in the model through hyper-parameter optimization. By comparing the prediction results with the log loss of each model, it is judged which model is more suitable for the data set of the project.

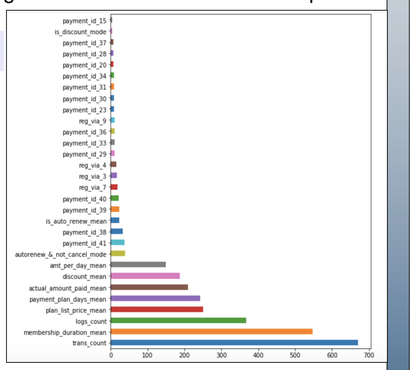
Methods

Data Processing

The data processing section contains three main steps

Select features

As shown in the figure, we have obtained some features that are useless through density maps and histograms.

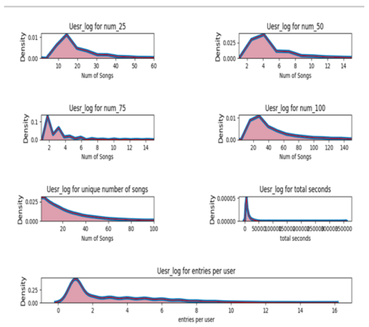


Reduce memory

The memory consumed by some 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. 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. After this process we have already reduced around 50%.

Data loss or exception

we filled NANs with zeros and filled inf with zeros. Second we found out one missing 'msno',we filled some data. After data cleaning, we scaled the data from 0 to 1.



Evaluation metric

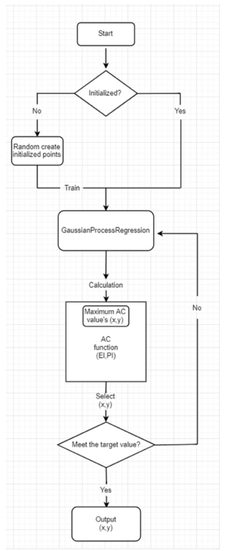
$$logloss = -\sum_{i=1}^{n}\left(\frac{y_i}{n}\log(p_i) + \frac{1-y_i}{n}\log(1-p_i)\right)$$

$$logloss = -(0.14log(0.14) + 0.86log(0.86)) \approx 0.405$$

Since the distribution on training data is: 15% of the users are churned, so if we are using random guessing, the logloss should be approximate 0.405. So lesser the logloss we got, better the result we have!

Bayesian hyperparameter optimization

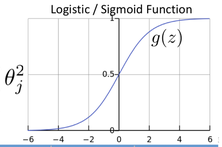
Since the objective function is unknown, the Bayesian strategy is to treat it as a random function and place a prior over it. The prior captures our beliefs about the behaviour of the function. After gathering the function evaluations, which are treated as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as infill sampling criteria) that determines what the next query point should be.



logistic regression

Logistic regression is classic binay classification and in this project I use grid search to exhaustive searching through the specified subset of hyper-parameters..

$$J_{\text{regularized}}(\theta) = J(\theta) + \lambda\sum_{j=1}^{d}\theta_j^2$$



Penalty \ C	0.01	0.1	1	10	100
L1	0.274892	0.274657	0.274634	0.274028	0.274469
L2	0.274829	0.274670	0.273530	0.273675	0.274472

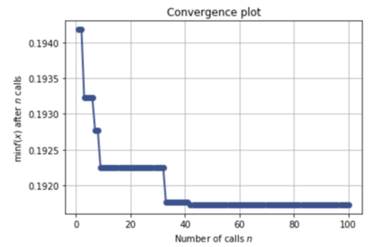
Random Forest

Based on the decision tree, the idea of random forest is to integrate multiple decision trees. Each decision tree is a classifier. Finally, the category with the highest number of votes is specified as the final output.

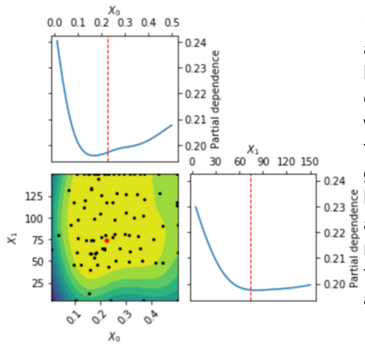
After we build a RF classifier, we applied Bayesian Optimization to select the value of number of decision trees(n_estimators) and max_depth of trees. Considering the speed of operation, we choose n_estimators = 240, max_depth = 10

LightGBM`

Light GBM is a gradient boosting framework that uses tree based learning algorithm.



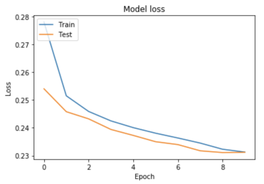
Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise While other algorithm grows level-wise.

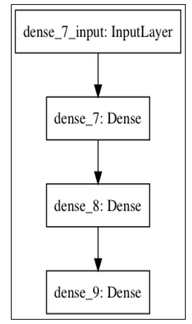


It while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.

Neural Network

Neural Network is also a good way for a binary classification problem. For this example we used a model as the right image shown.





Results

The best result with logistic regression model

sample_submission_v2.csv 0.14948 0.15038
25 minutes ago by Tong Ye

After tuning the inverse of regularization strength and choose the L2 regularization, the final result on Kaggle is shown below. Apparently it's not perfect, and the tuning process using grid search is too slow.

The best result with Random Forest model

According to the process of tuning, the accuracy of RF model increases along with the max_depth. However, the running time also goes up. As the result shows, the performance is not good as well as other methods.

Submission and Description — Private Score

sample_submission_v2.csv — 0.16068
a day ago by lllllllll

The best result with LightGBM model

submission_test.csv 0.12004 0.12108
11 days ago by Brother jia
add submission details

The best result with NN model

submission_test.csv 0.18216 0.18431
6 days ago by Brother jia

Conclusions

Through the data results of the four models and the comparison of log loss, we found that the LightGBM model is the best. The effect of the logistic regression model in the three models is the worst.
For the tuning process, the basic gid search works well in find the best result, but it takes too much time because it exhausts every possible combination. So we choose another method, bayesian optimization, in the afterward works. And it shows almost the same optimal result and takes much less time than the grid search.

Reference

- **G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.**