KKBox's User Stickiness Analysis

Yujia Wang, Liyi Cao, Tong Ye, Weixuan Jiang

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

1. Project Task

Our project input is the data provided by Kaggle from KKBox, and the task is to build an algorithm that predicts whether users will be lost after the subscription expires and analyze the reason for the user leaving.

2. Related Work

Currently, the company uses survival analysis techniques to determine the residual membership life time for each subscriber.

3. Approach

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.(We implemented LR this time)

SVM,**Neural Network**: (To be implemented next time)

4. Dataset and Metric

train_v2.csv contains the IDs and whether these users have churned or not. 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.

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.

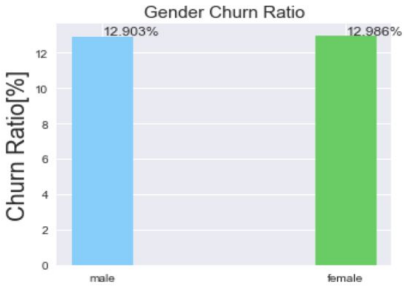
Eventually we will upload our predicted probabilistic results(pi) on kaggle. The system will automatically score for us.

5. Preliminary Results

1. Feature Selection

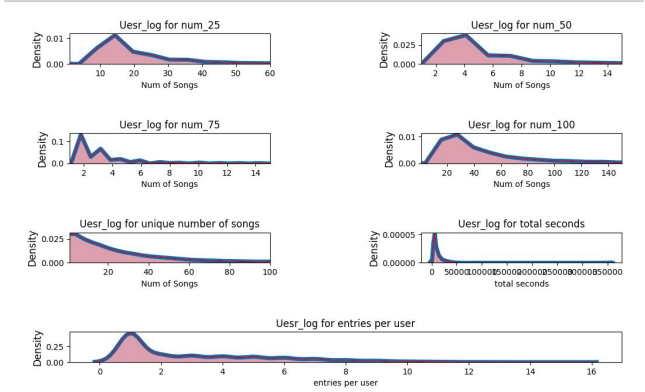
We want to decide whether the features may influence the prediction. The selected features are listed as below.

Gender:



When we analyze gender, we counted each gender population and each gender is_churn population. As the result showed above, the churn ratio of each gender is very similar. So, we can believe that the gender won't affect the the prediction. Thus, we can remove this features.

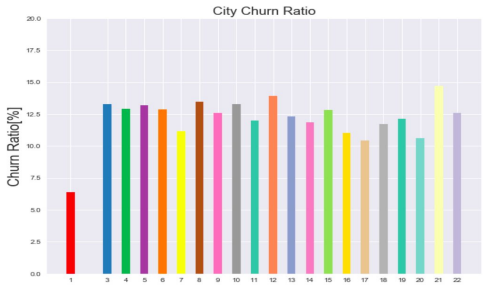
User_logs.cvs all features:



The figure shows several features in User_log: num_25, num_50, num_75, num_100, num_unq, total_secs and set a features as entries per user. By comparing the density map of the whole data (blue thicker line) and churn's user's density map (red area),we observe that these pictures can be almost completely coincident. It can be proved that all of the above features are useless features. The reason of why we choose the density map instead of the histogram is that the density map can reflect the distribution of most users' selections, while the histogram can only perform quantitative/average analysis that may get a large deviation.So the whole document we will not use in our project.

City:

By analyzing the ratio of churn in each city, we find that the probability of churn in cities is not much different except for City1. City1's data is a fairly small part of all data. Therefore, the cities feature does not largely affect our training and prediction results. We decided to abandon this feature.



2, Data selection and predicting methods

We discarded the features that have similar distributions in churn dataset and not churn dataset in training our model. In this case we use all of the features in transactions.csv combined with 'city' features as our training features.

In training step, we picked the users who have their expiration dates fall in March, 2017 as our training data and use the given labels of whether the customers churn or not in March,2017 in train_v2.csv as the training label. For testing step, we select users who's plan dates end in April as the testing data. By plugging the data into the logistic model trained in training step, we produce our own prediction of whether the customers churn or not in April. In the end, we will upload our predictions on the kaggle website to get the log-loss score.

3, Preliminary result by logistic regression

In addition to all the features as described in part 2, we add some more features which we think are important as well, such as 'sum_plan_days' = sum of the 'payment_plan_days', 'sum_paid' = sum of the 'actual_amount_paid', 'amount_per_day' = 'sum_paid' / 'sum_plan_days' . And before we use the sklearn logistic regression model, we did some data cleaning. First, we filled NANs with zeros in transactions.csv. Second, we found out some of the 'amount_per_day' are inf because we set some 'sum_plan_days' with zeros, so we found them and also set them to 0. Finally, since the csv file we should submit needs 907471 rows, we need to first merge the sample_submission.csv and 'other features' on 'msno(user id)'. But it turned out that the merged data only has 907470 rows. So we found out that missing 'msno' and insert it in the final possibility result with '0.5', in the right index. After data cleaning, we scaled the data from 0 to 1, and used the logistic regression model to fit our train data. After this step, we also used log loss to calculate the training loss, which is 0.22.

```
from sklearn import metrics
metrics.log_loss(train_label,y)
```

```
0.2274896632865764
```

Finally we used the test_data to predict. After we did this, we submitted it on the Kaggle website, and our log_loss score is 0.16.

Execution time	Score
7 seconds	0.16010

And the first in the leaderboard is about 0.07, so we still have hard works to do.

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 Support Vector Machine	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.