Capstone Project 1

KKBox Customer Churn Prediction

How will I retain my customers?

- Most companies utilize subscription business model, including KKBox
- KKBox shared their customer data on Kaggle to learn more about predicting churn rates.
- **Inquiry:** Can we predict if a user will make a new service subscription transaction within 30 days after the current membership expiration date based on their behaviors and interactions with the product?

KKBox is a Taiwan-based music streaming software

Data Provided

Train Dataset
msno
is_churn

992,931 rows

Train set is extremely biased with only a 6.4% churn rate → Downsample majority group (new churn rate 45.8%)

Member Data
msno
city
bd
gender
registered_via
registration_init_time

6,769,473 rows

21,547,746 rows

Transactions Data
msno
payment_method_id
payment_plan_days
plan_list_price
actual_amount_paid
is_auto_renew
transaction_date
membership_expire_date
is_cancel

400,000,000+ rows

User Log Data
msno
date
num_25
num_50
num_75
num_985
num_100
num_unq
total_secs

Data After Downsampling

Member Data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 123815 entries, 0 to 123814
Data columns (total 6 columns):
msno                    123815 non-null object
city                    123815 non-null int64
bd                      123815 non-null int64
gender                  63206 non-null object
registered_via          123815 non-null int64
registration_init_time  123815 non-null int64
```

Issues:

- Missing data for gender
- Lots of outliers for age (extreme negative and positive values)

Transactions Data

```
RangeIndex: 1909677 entries, 0 to 1909676
Data columns (total 9 columns):
msno                    object
payment_method_id       int64
payment_plan_days       int64
plan_list_price         int64
actual_amount_paid      int64
is_auto_renew           int64
transaction_date        int64
membership_expire_date  int64
is_cancel               int64
```

Description:

- Spans over 3 years (2015/01/01 to 2017/02/28)

Issues:

- Columns with 0 values in the early days

User Log Data

```
RangeIndex: 10927698 entries, 0 to 10927697
Data columns (total 9 columns):
msno          object
date          float64
num_25        float64
num_50        float64
num_75        float64
num_985       float64
num_100       float64
num_unq       float64
total_secs    float64
```

Description:

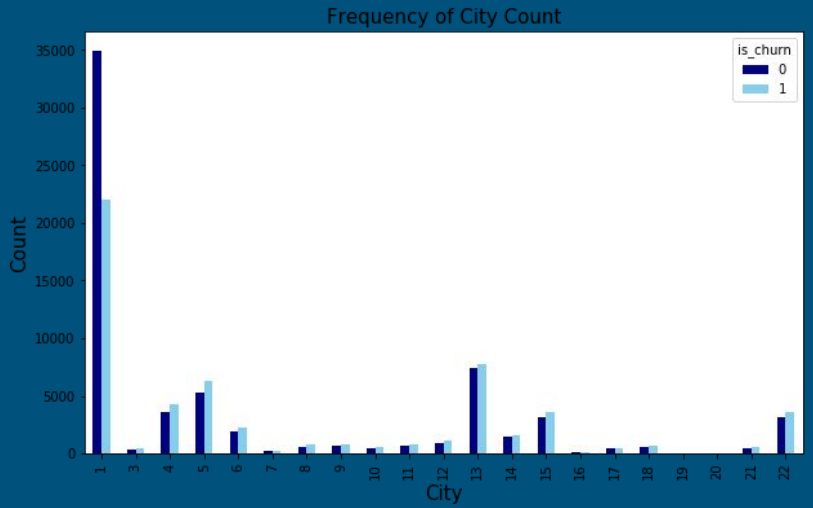
- Spans over 3 years (2015/01/01 to 2017/02/28)

Issues:

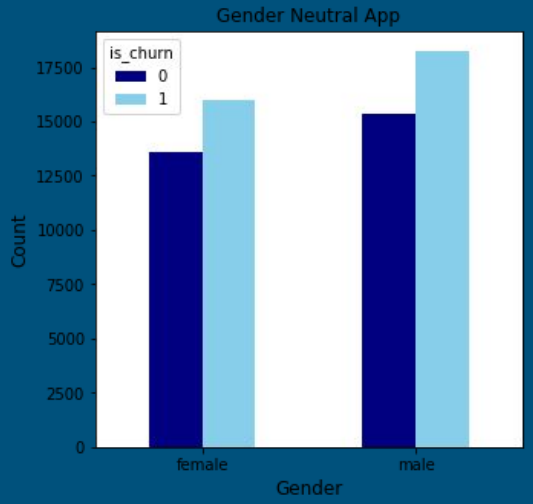
- Total seconds has extremely negative value as minimum that occured between 2015/04/22 and 2016/04/18

Exploratory Data Analysis: Churn vs Not Churn

Both groups are similar in certain demographics.



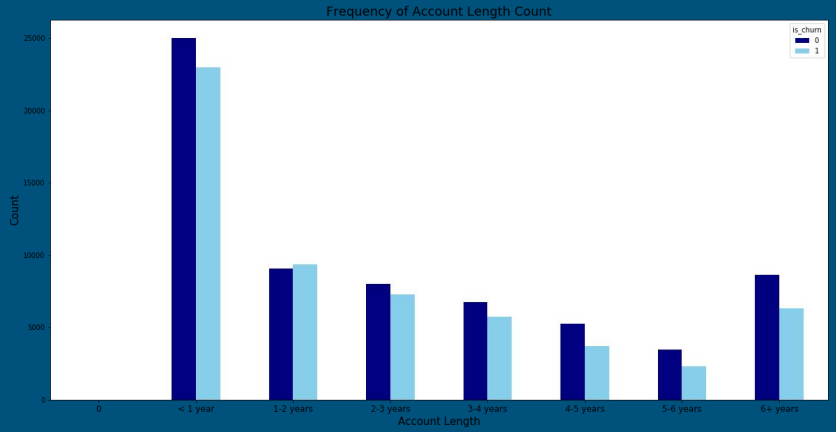
Most live in city 1



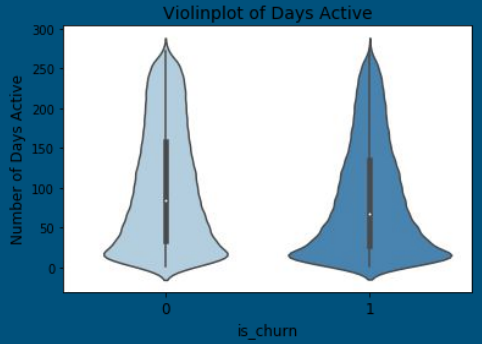
Even gender ratio

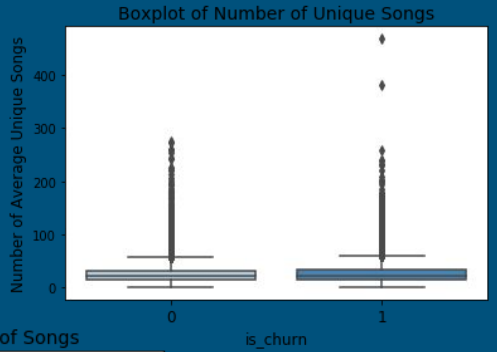
Exploratory Data Analysis: Churn vs Not Churn

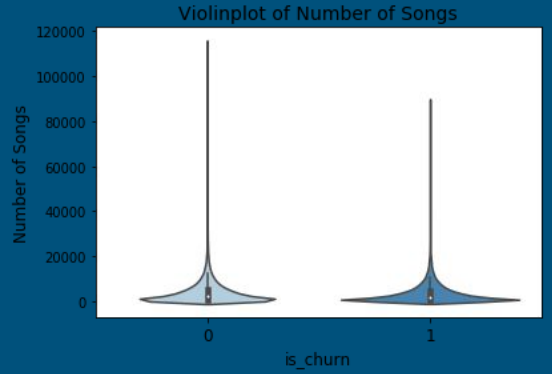
Their behaviors with the app was also similar.



Similar length of account distribution



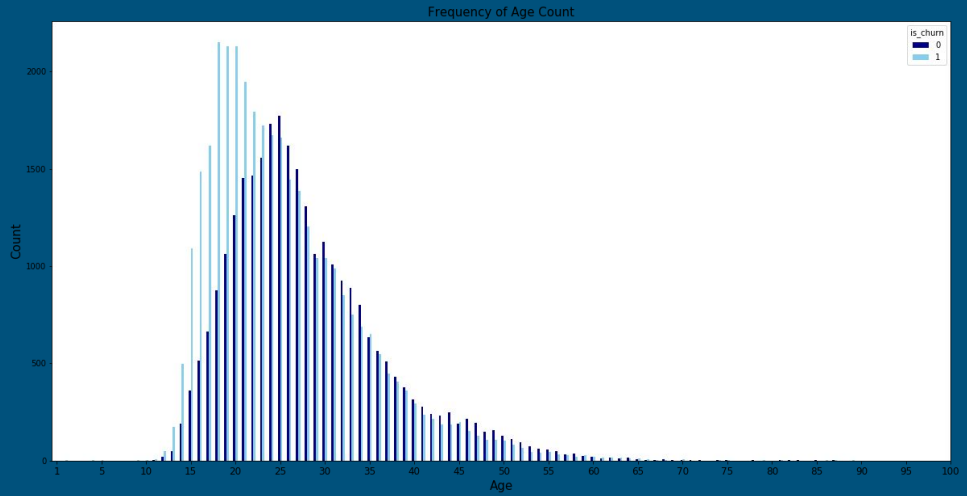




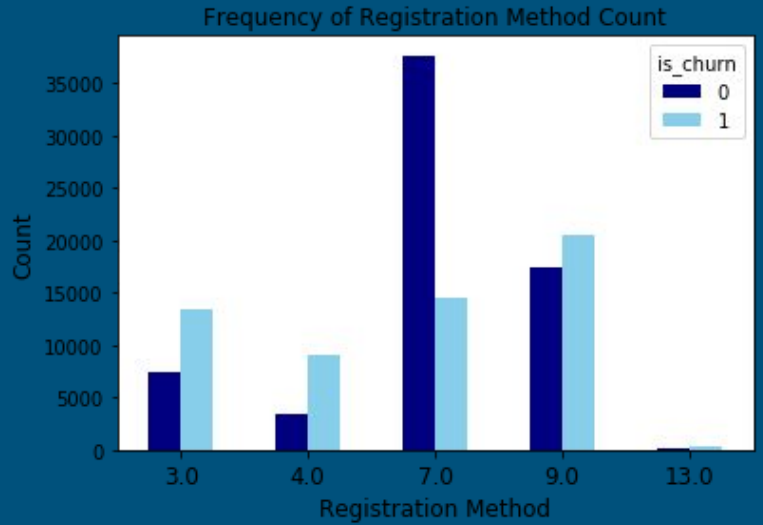
There's not much difference in the distribution of number of days and songs.

Exploratory Data Analysis: Churn vs Not Churn

A few demographics features did vary between the 2 groups.



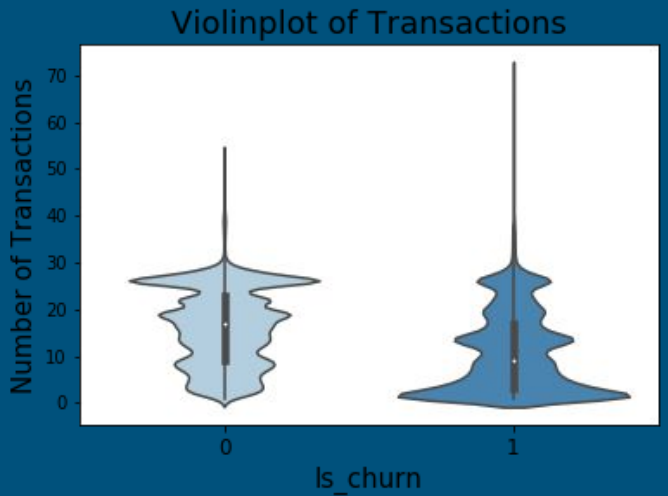
Most of churned users are in their early 20s while users who did not churn are mostly in their late 20s.



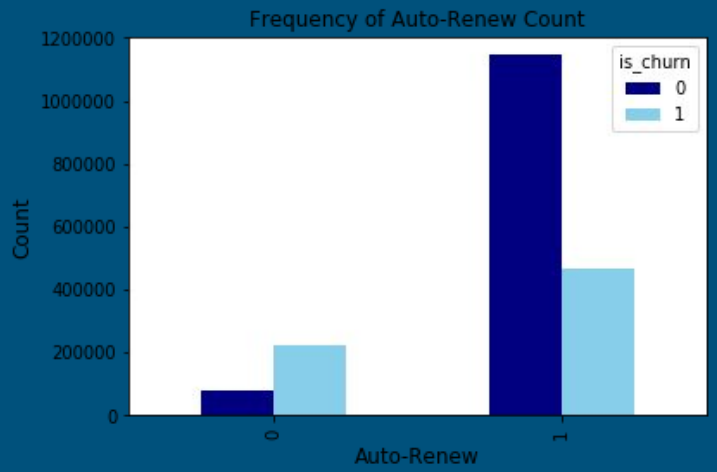
Users who did not churn mostly signed up via registration method 7 while most churned users registered via method 9.

Exploratory Data Analysis: Churn vs Not Churn

They also differed in the monetary transactions made.



Those who did not churn seemed to have made more transactions than those that churned.



The users who used the auto-renew feature tend to be less likely to churn.

Inferential Statistics

Although the distributions may seem to look like they differ visually, these hypotheses should be tested to see if the difference is statistically significant

Method used: Permutation (Bootstrap) Hypothesis Testing

Null Hypothesis	Mean Difference	95% Confidence Interval	P-value	Reject $H0$
There's no difference in the age between user who churned and did not churn.	2.574	[2.437, 2.711]	0	True
There's no difference in auto-renewal rate between user who churned and did not churn.	0.441	[0.437, 0.445]	0	True
There's no difference in the number of transactions between user who churned and did not churn.	5.520	[5.431, 5.609]	0	True
There's no difference in the number of unique songs listened per session between user who churned and did not churn.	-1.093	[-1.299, -0.888]	0	True

Building the Classification Model

To take advantage of the temporal aspect of the data, we will split the dataset into 2 separate time periods for training and testing data:

- Training Data: February (2017/02/01 - 2017/02/28)
- Testing Data: March (2017/03/01 - 2017/03/31)

Feature Engineering

Transactions Data
msno
payment_method_id
payment_plan_days
plan_list_price
actual_amount_paid
is_auto_renew
transaction_date
membership_expire_date
is_cancel

Transactions Data
msno
avg_actual_amount_paid
avg_is_auto_renew
avg_is_cancel
avg_payment_plan_7
avg_payment_plan_30
avg_discount_received
trans_count

Member Data
msno
city
bd
gender
registered_via
registration_init_time

Member Data
msno
age
gender_male
registered_via
days_since_reg
city_3
city_4
city_5
city_6
city_7
city_8
city_9
city_10
city_11
city_12
city_13
city_14
city_15
city_16
city_17
city_18
city_19
city_20
city_21
city_22
registered_via_4
registered_via_7
registered_via_9
registered_via_13

Feature Engineering

User Log Data
msno
date
num_25
num_50
num_75
num_985
num_100
num_unq
total_secs

User Logs Data
msno
avg_num_25
avg_num_50
avg_num_75
avg_num_985
avg_num_100
avg_num_unq
avg_rate_num_25
avg_rate_num_50
avg_rate_num_75
avg_rate_num_985
avg_rate_num_100
total_secs
log_count
days_since_login

In total, there are 48 features.

Data Preprocessing

Not all users in the training data and testing data that had churn information had records of transactions or user activity.

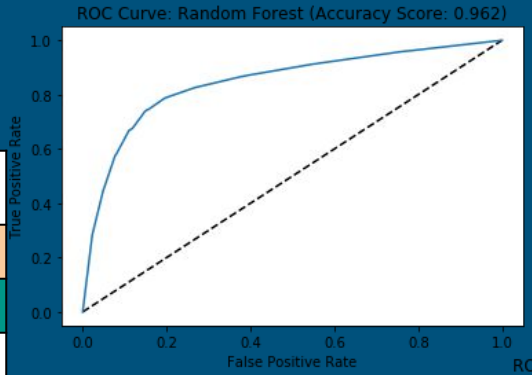
- If the user did not have any transactions data, the transactions count feature was filled with 0.
- If the user did not have any user log data, the days since log in was filled with 31.
- All the other features were filled with the sample mean for the column.

In the end, all the values were normalized.

Selecting Classification Model

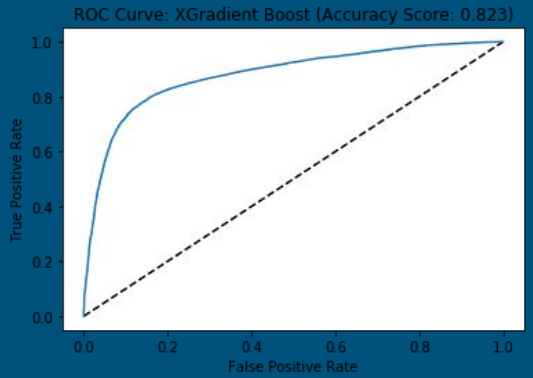
Fit 5 different classification models on the data using the default parameters and utilized log loss as the model metric.

Classification Model	Average Log Loss
Logistic Regression	0.463
Random Forest	1.464
Adaptive Boosting	0.681
Gradient Boosting	0.428
Extreme Gradient Boosting	0.427



Random Forest
Accuracy Score: 0.962

XGBoosting
Accuracy Score: 0.823



Fine Tuning Hyperparameters

Utilized GridSearch cross validation to select best parameters for both models:

- Extreme Gradient Boosting
  - colsample_bytree = 0.8
  - learning_rate = 0.05
  - max_depth = 8
- Logistic Regression
  - penalty = ‘l2’
  - dual = False
  - C=1
  - max_iter = 100

Interpreting Logistic Regression Coefficients

Feature	Coefficient
avg_is_cancel	0.905624
avg_payment_plan_7	0.362903
avg_discount_received	0.165612
avg_is_auto_renew	-0.947160
trans_count	-0.848782
log_count	-0.192631
registered_via_7	-0.145023

The features with the highest effect mostly seem to relate to the transactions dataset.

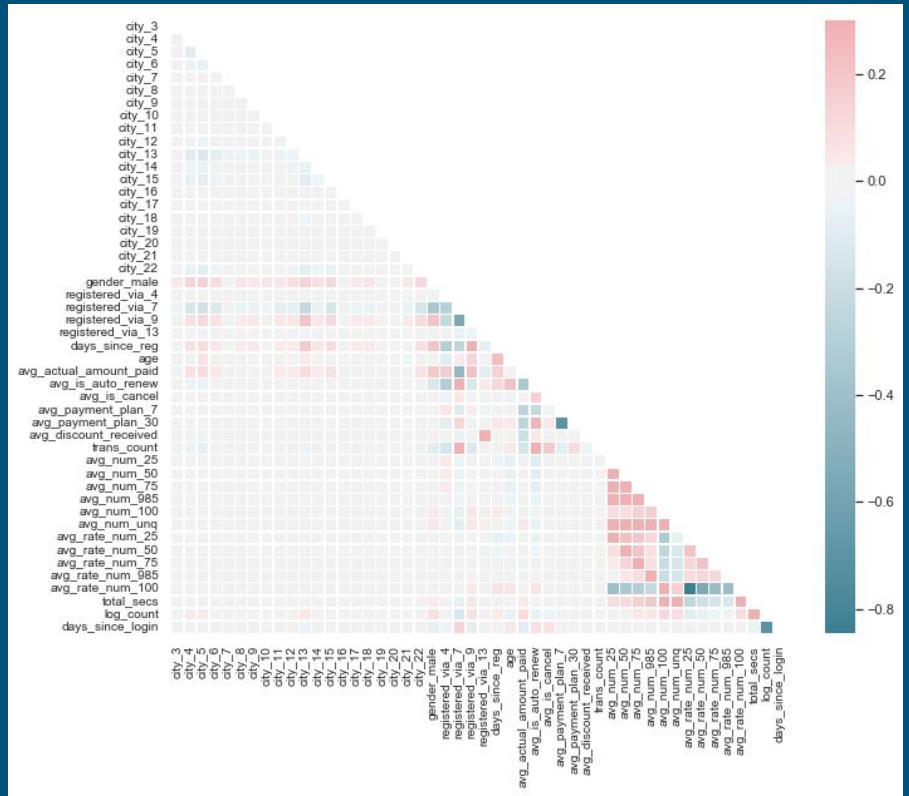
- Those that opted to use the auto-renewal feature are less likely to churn (coefficient of -0.947)
- The more number of transactions made, the less likely the user would churn (coefficient of -0.849).
- Those who had a payment plan of 7 days were more likely to churn. (coefficient of 0.363)
- The registration method also seemed to matter since those that registered via method 7 were less likely to churn. (coefficient of -0.145)

Feature Selection

Feature selection is the method of selecting the most important features to be used in the model. There are multiple ways to perform feature selection:

1. Drop one of the variables that is highly correlated to another
2. Use the recursive feature elimination method to see dropping which features would affect the model the least
3. Perform feature importance to see which features are the most important for the model

Feature Selection: Multicollinearity



As shown by this pairwise scatter plot, there are 3 pairs of variables that are highly negatively correlated.

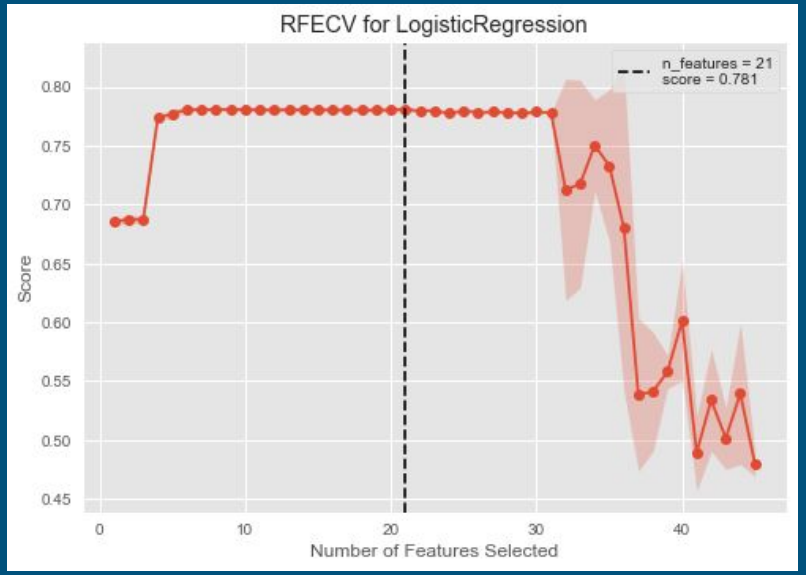
- avg_payment_plan_30 and avg_payment_plan_7
- avg_rate_num_25 and avg_rate_num_100
- log_count and days_since_login.

Dropped the following features since produced the better score:

- 'Avg_payment_plan_30'
- 'Avg_rate_num_25'
- 'days_since_login'

The log loss score after dropping these variables was 0.463304.

Feature Selection: Recursive Feature Elimination



RFE is able to work out the combination of attributes that contribute to the prediction on the target variable (or class).

As the graph suggests, the ideal number of features to keep is 21.

Feature Selection: Recursive Feature Elimination

Column name	Keep
city_7	True
city_8	True
city_9	True
city_10	True
city_16	True
city_18	True
city_19	True
city_20	True
registered_via_7	True
registered_via_9	True
registered_via_13	True
avg_is_auto_renew	True
avg_is_cancel	True
avg_payment_plan_7	True
avg_payment_plan_30	True
trans_count	True
avg_rate_num_25	True
avg_rate_num_50	True
avg_rate_num_75	True
avg_rate_num_985	True
avg_rate_num_100	True

Here are the 21 features to be kept.

A majority of the features kept are regarding the demographics of the user. The other half of the features is composed of the nature of the user's transactions (type of payment plan, number of transactions, auto-renewal, cancellation) and the average rate of how much of the song the user actually listens to. The number of active days did not seem to make it onto the list.

The log loss score after dropping these variables worsened to 0.468988.

Feature Selection: Feature Importance

The scoring method used to rank each feature is Mutual Information.

Number of Top Features Kept	Log Loss Score
18	0.4674
20	0.4642
24	0.4634
28	0.4634
30	0.4634

Feature Selection: Feature Importance

Feature	MI Score
avg_is_auto_renew	0.190177
trans_count	0.145051
avg_actual_amount_paid	0.123797
avg_is_cancel	0.088513
avg_payment_plan_7	0.075063
avg_discount_received	0.064310
registered_via_7	0.061058
age	0.023075
registered_via_4	0.020251
avg_payment_plan_30	0.015818
days_since_reg	0.013782
registered_via_9	0.011532
gender_male	0.010578
city_14	0.005798
city_5	0.005577
city_4	0.005511
log_count	0.004673
city_15	0.004220
city_13	0.004218
city_12	0.004160
avg_num_985	0.004034
city_17	0.003831
avg_rate_num_25	0.003822
city_6	0.003816

Here are the 24 features to be kept.

The top 6 features all seem to relate to the details of the transactions the users have made. The most important feature is the average auto renewal rate.

The features kept due to feature importance is very similar to the features kept due to recursive feature elimination. They are comprised of features related to transactions details and user demographic information. The user activity does not seem to be that important as only 3 of the features are from the user activity dataset.

The log loss score is slightly better through this method with a log loss score of 0.46393.

Principal Component Analysis

Another way to improve the model is through adding more variance into the model. Principal Component Analysis (PCA) linearly transforms the features into a lower dimension while keeping the most important components of the feature. The larger the variance the larger the amount of information the variable contains. Since the Recursive Feature Elimination method and the Feature Importance both removed many features from the dataset, the variance in the dataset was reduced. I have added the top 2 principal component into each of the features of the dataset to see if they would result in a better log loss score.

Feature Selection Method	Log Loss Before PCA	Log Loss After PCA
Recursive Feature Elimination	0.4690	0.4677
Feature Importance	0.4639	0.4637

Final Results

The final model uses the top 24 most important features as well as the 2 principal components to train on. The following are the results on the test data and train data:

Algorithm Used	**Train Data Results**	**Test Data Results**
XG Boosting	Accuracy score: 0.8367 Log Loss: 0.3953	Accuracy score: 0.8027 Log Loss: 0.4559
Logistic Regression	Accuracy score: 0.8129 Log Loss: 0.4637	Accuracy score: 0.8016 Log Loss: 0.4718

Limitations and Further Opportunities

- Model can be improved using server with higher CPU
  - Solving issue of unbalanced dataset by assigning higher weight to users that churned so that all the data can be used to train model
  - Fine tune more hyperparameters
- Model can be improved in a better understanding of the variables provided
  - Extreme negative values in columns that should strictly be positive (age and total seconds listened)

A model with a low log loss allows accurate prediction of the likelihood of churning. Clustering similar users together in terms of demographics and activity can pave the opportunity of a recommender system that provides the most effective promotion for the user.