Marco Riva

Final Report:
KKBox Churn Prediction

Problem Statement

KKBox is Asia's leading music streaming service with millions of users and songs offered via subscription model. KKBox is interested in understanding customer behavior to improve customer experience and satisfaction and drive growth and value by reducing its churn rate. For this purpose, KKBox has provided a rich dataset containing user transactions, logs and personal information, which can be used to train a machine learning model and predict if a user will churn after their subscription expires.

The goal of this project is to transform the raw data into actionable insight that can lead to a better use of marketing resources for the next KKBox campaign. In particular, a model able to identify potential users leaving the service is particularly important for subscription models given that user acquisition costs are generally significantly higher than retention costs. Hence, churn should be strictly monitored and minimized in order to maximize revenue.

The objective of this work is to build and deliver a supervised machine learning model for the binary classification of an imbalanced target variable (~5% churn rate) by predicting the user churn probability. The metrics chosen for model selection is the log loss, as suggested by the Kaggle competition requirements. However, the recall of the positive class (churn) and precision of the negative class (no-churn) are the more interesting metrics during production. In detail, we aim to reduce the log loss value of 0.19 obtained by a dummy classifier which always predicts the majority class. Furthermore, we wish to identify features which are highly predictive of customer churn to provide actionable insights.

Data Wrangling

The raw dataset consists of three tables of a relational database in csv format organized as:

- Member Information, ~5M records including:
    - msno: user ID
    - city
    - bd: age
    - gender
    - registered_via: registration method

    - registration_init_time: format %Y%m%d
    - expiration_date: format %Y%m%d, taken as a snapshot at which the member.csv is extracted. Not representing the actual churn behavior
- User Transactions, ~22M records including:
    - msno: user id
    - payment_method_id: payment method
    - payment_plan_days: length of membership plan in days
    - plan_list_price: in New Taiwan Dollar (NTD)
    - actual_amount_paid: in New Taiwan Dollar (NTD)
    - is_auto_renew
    - transaction_date: format %Y%m%d
    - membership_expire_date: format %Y%m%d
    - is_cancel: whether or not the user canceled the membership in this transaction
- User Logs, ~400M records including:
    - msno: user id
    - date: format %Y%m%d
    - num_25: # of songs played less than 25% of the song length
    - num_50: # of songs played between 25% to 50% of the song length
    - num_75: # of songs played between 50% to 75% of of the song length
    - num_985: # of songs played between 75% to 98.5% of the song length
    - num_100: # of songs played over 98.5% of the song length
    - num_unq: # of unique songs played
    - total_secs: total seconds played

Furthermore, the data is provided by the following tables which are divided by month when churn was recorded:

- transactions.csv: User transaction until the end of February 2017
- transactions_v2.csv: User transaction in March 2017
- user_logs.csv: User logs until the end of February 2017
- user_logs_v2.csv: User logs in March 2017
- train.csv: User ids and target variable for January 2017
- train_v2.csv: User IDs and target variable for February 2017
- submission.csv: User IDs and target variable for March 2017 (test set)

The goal is to predict churn for the month of April 2017 on the test set which includes data until the end of March by using January data (churn in February) and February data (churn in March) as training and validation sets. The target value has comparable imbalance among the 3 month period and is about ~5%.

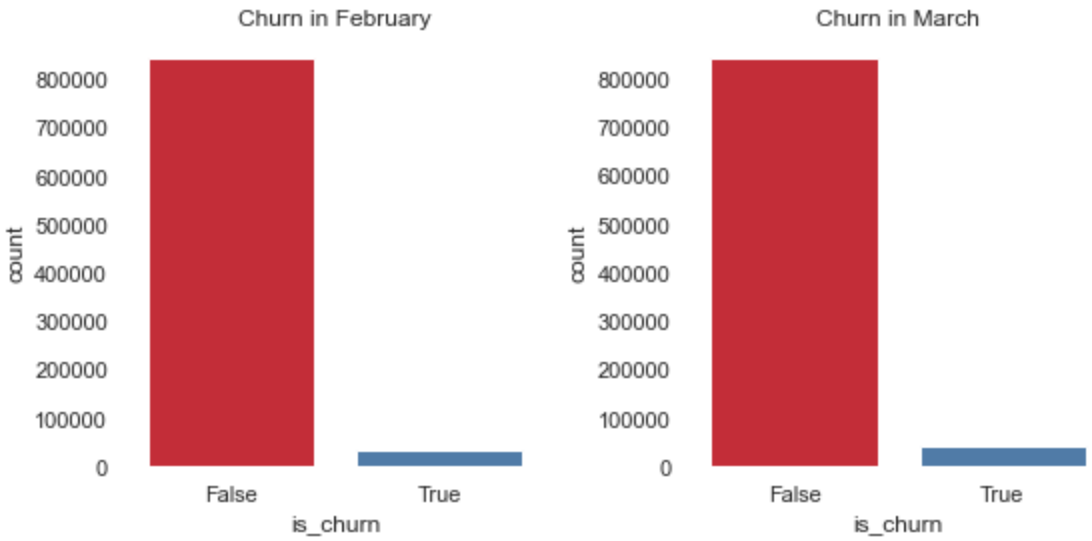

Figure 1: Churn count in February and March.

Data cleaning resulted in removing features with more than 40% missing values and correcting errors in the reported age of the population, which contained values ranging from -7000 to 2015. In particular we kept ages between 0-100 and converted to null values the remaining records. We also removed transactions before the year of 2015 as other records in the 1970-2015 period were also reported.

Exploratory Data Analysis

A first data exploration was performed on the cleaned transactional, logs and member data. In particular, we noticed the vast majority of membership are renewed on a monthly basis, although there are exceptions regarding both longer or shorter subscription times, e.g. to the weekly basis. Subscription plans with duration less than a week are considered as trials, as seen in Figure 6.

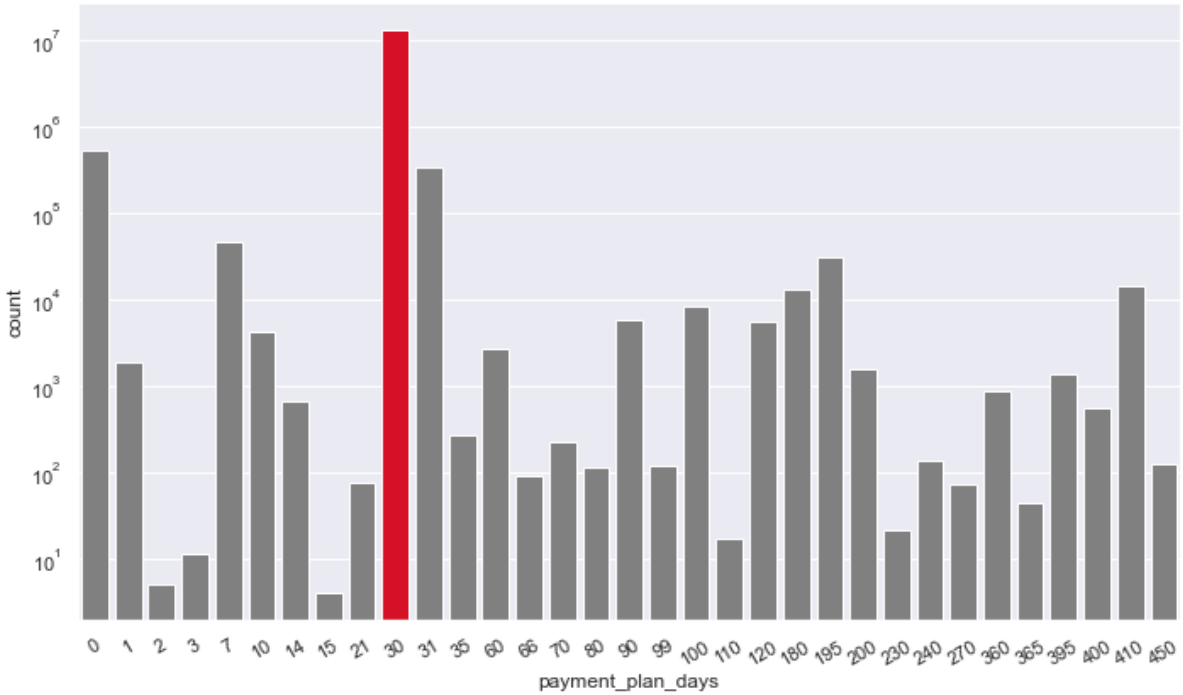

Figure 2: Count of the payment_plan_days feature showing most memberships last 30 days.

A variety of payment methods are available, but most users use either method 41 or 40, as seen in Figure 7.

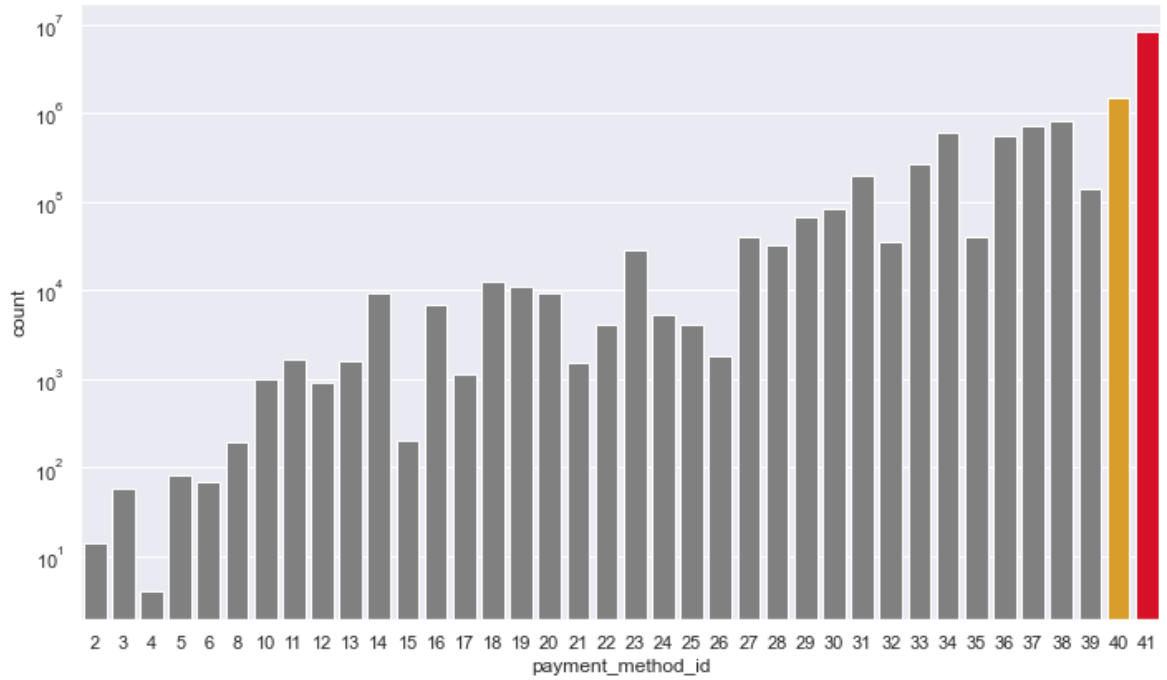

Figure 3 :Count of the payment_method_id feature showing most payments are performed via method 40 and 41..

Transactions with canceling action are rare and most customers have enabled subscription auto-renew.

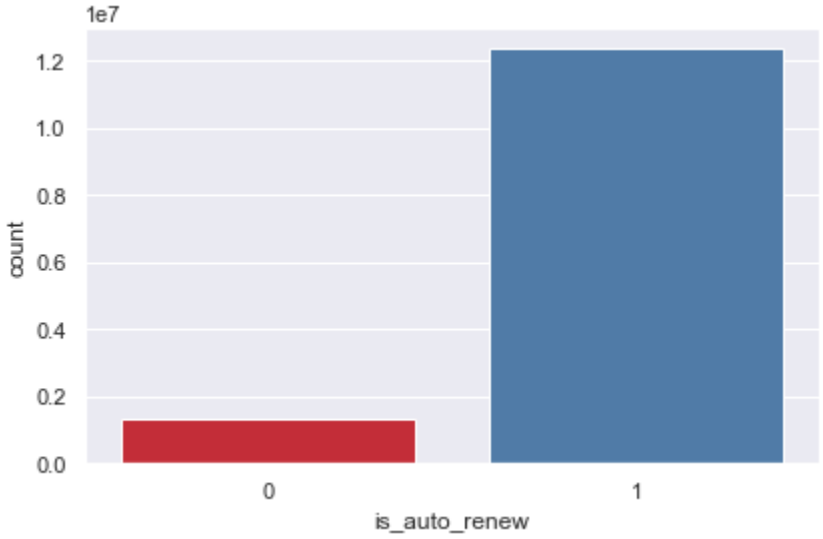

Figure 4: is_auto_renew count.

Finally, the distribution of users log, where the stacked histogram indicates the number of songs played for less than the various percentiles of the songs length, is shown in Figure 9.

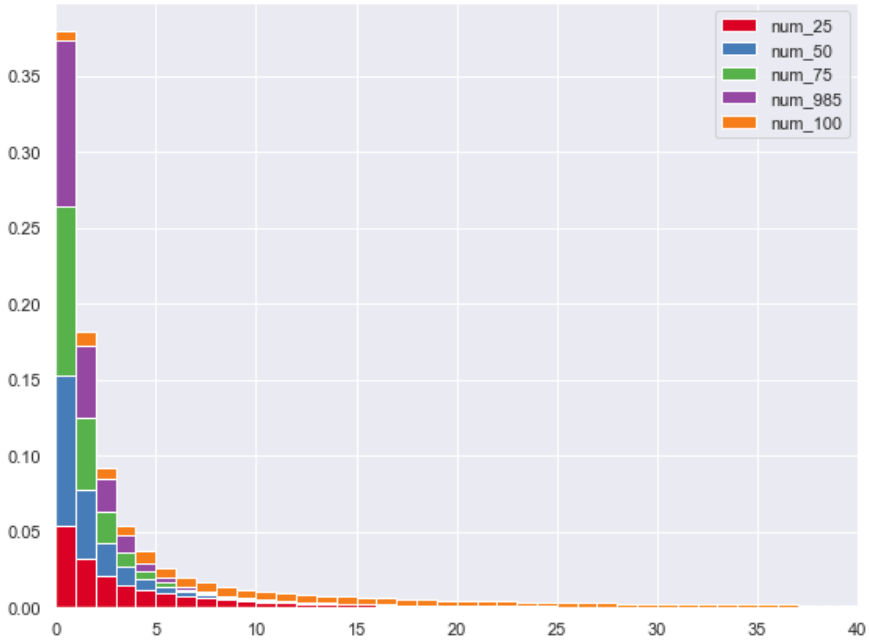

Figure 5: Distribution of number of songs played for a certain percentile of the total song length.

In order to investigate the cause of churn, transactional and data logs having multiple records per user were aggregated through sum, average, and count operations to generate summarizing features and a tidy set was obtained (further details are provided in the Feature Engineering Section). Additional exploratory data analysis was then performed on this compact dataset with particular interest in the relationship between the features and the target values.we report here the most interesting findings.

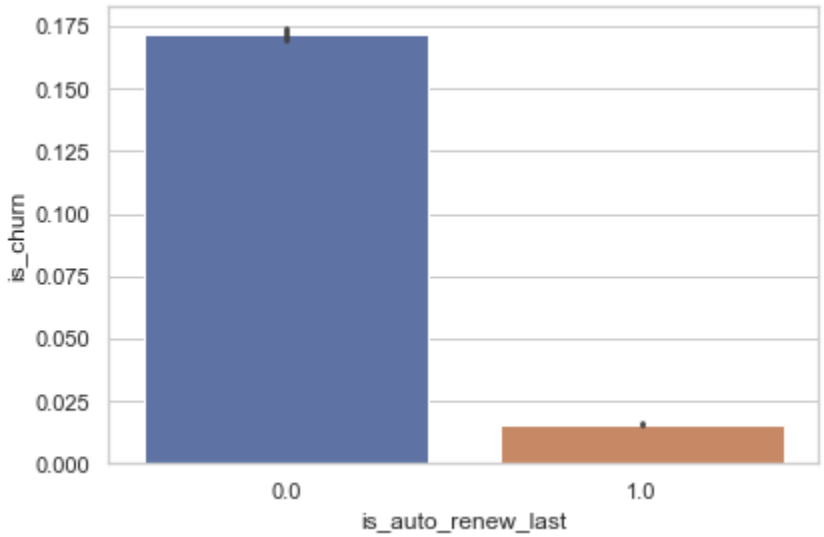

Figure 6: Churn rate vs is_auto_renew_last.

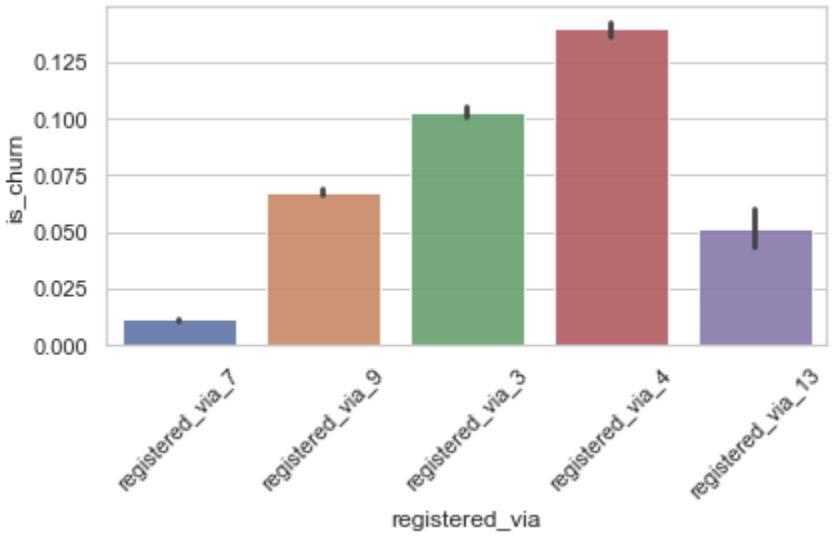

Figure 7: Churn rate vsregistered_via.

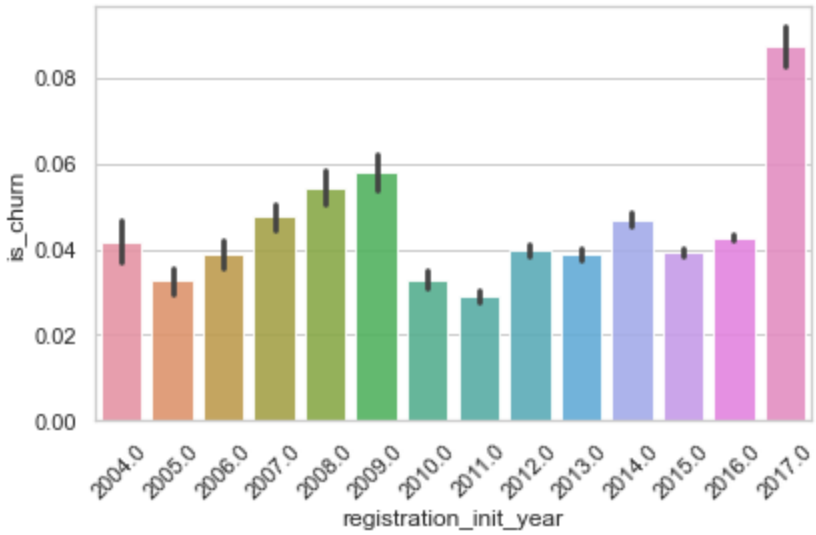

Figure 8:Churn rate vs registration_init_year.

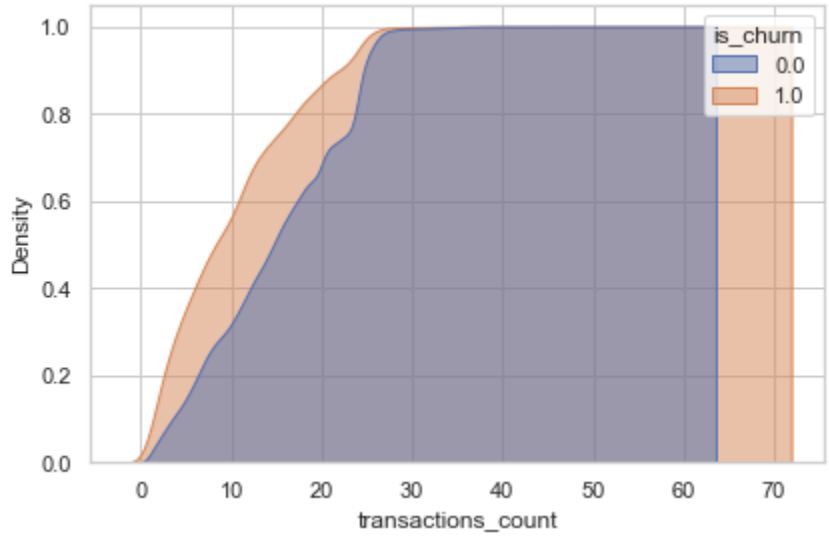

Figure 9: Transaction count cumulative distribution hued by is_churn.

Feature Engineering and Modeling

Since we were provided a rich dataset with multiple entries per user, the first step to create a trainable model was aggregation by count, sum, and averages of transactional and log data for each member. The dataset number of rows was therefore reduced from several millions to about 900K. We also added some derived features, e.g. a boolean flag for customers that do not have the auto-renew option enabled and have canceled their subscription or the relative time between the last log-in or transaction and the membership expiration date. These derived tidy datasets were then merged with the user member dataset to obtain the final train, validation

and test set. Since the objective was to predict churn for the month of April, the dataset for churn in February and March were used to train and validate the model, under the verified assumptions of consistent distributions among the datasets.

A data preprocessing pipeline was implemented to perform missing values imputation and encode categorical features with a simple one-hot-encoder - this is omitted when we trained the CatBoost classifier since this boosting method is able to automatically handle categorical columns via mean target encoding. After the encoding the dataset contains 133 features. The data was used to train CatBoost, XGBoost, and LightGBM. Even though boosting models are less interpretable than a simpler logistic regression, they still offer feature importances which shed some light on what choices the classifiers made and give insights into what features are most important. The metric of interest is the log loss (as requested by kaggle for the competition) but we will also track the recall of the positive class. The baseline performance of the dummy classifier predicting the majority class is 0.19 for the log loss.

Tuning and overfitting were addressed by searching the best parameters through a randomized search with cross validation. All models provided similar scores with train test log loss of ~0.11 and validation varying between 0.11 to 0.14. The most performing model is a tuned XGBoost which obtained a train log loss of 0.112 and a test log loss of 0.114 on the kaggle test set showing overfitting was accurately handled. However, due to the great imbalance of the dataset, this best model which would rank within the top 40 positions of the Kaggle Public and Private private leaderboard comes with the inconvenience of racing high recall of the positive class, e.g. 80%, at the price of a very low precision, e.g. 20%, as shown in the precision recall figure of Figure 10.

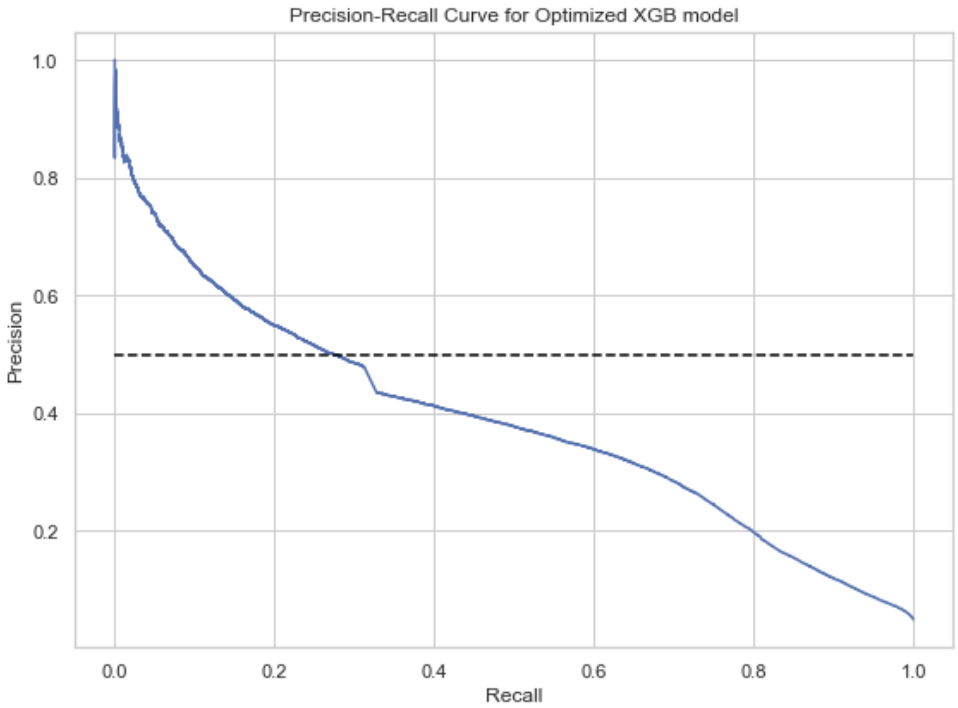

Figure 10: Precision-Recall curve for tuned XGBoost.

Feature importances for the top 20 features of the final model are shown in Figure 11. It is found that “is_auto_renew” is the most dominant feature in determining whether a customer will churn, in line with the exploratory data analysis results. The registration method follows together with the cost of the subscription averaged by day. The other features have lower importances.

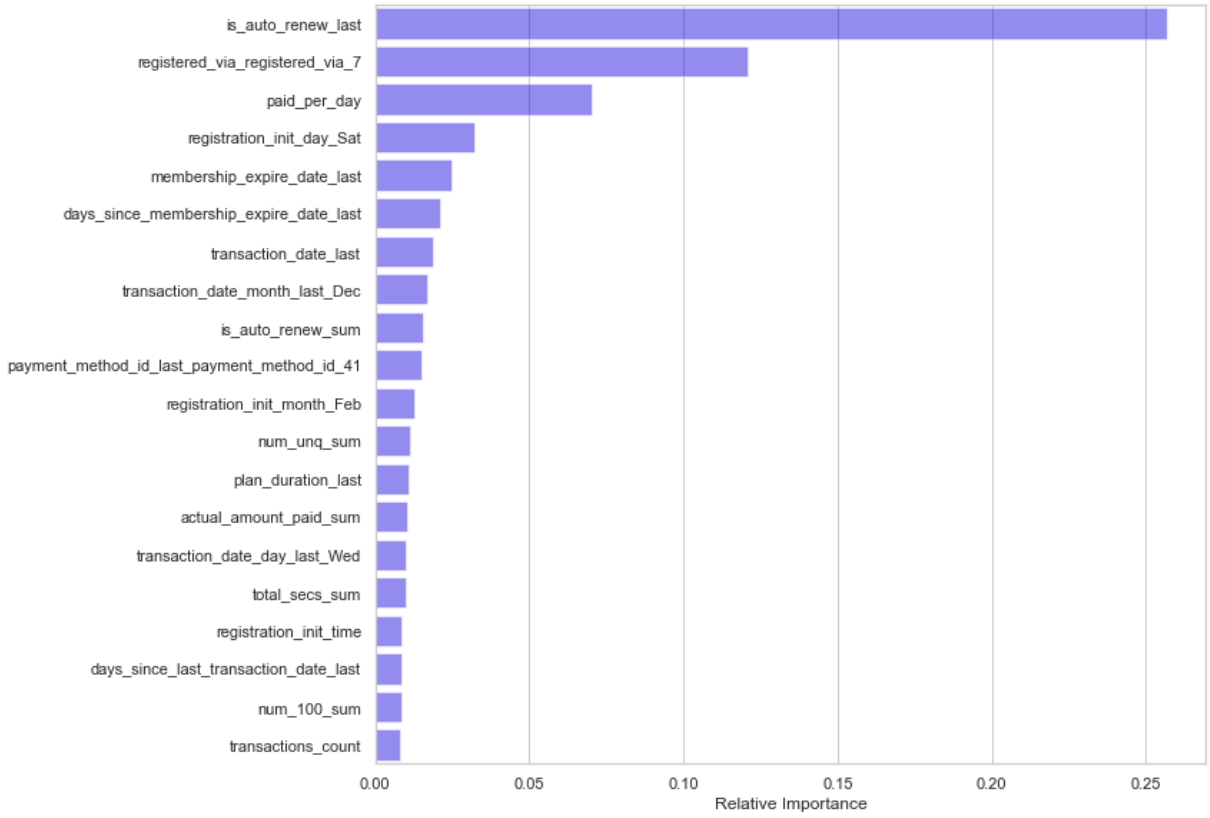

Figure 11 : Relative feature importances for best XGBoost model.

Key Findings and Future Research

First, the most predictive feature is “is_auto_renew_last” which implies that a user had the auto renew option active in the last transaction. Second, the registration method seems to be a powerful predictor of future customer churn or retention. Third, the remaining features have less relative importance and include several aspects from the price paid to the temporal gap between transactions, logs and membership expiration or the number of songs listened.

Future research should aim to develop a classifier with a better balance between recall and precision. Techniques to be considered SMOTE combined with random undersampling as well

as modifications of the Loss function to give more weight to the minority class. Currently the model can be used to identify churners but accepting very low precision in favor of high recall