Bradley Mensah
Springboard Data Science, Nov 2nd Cohort

# Capstone 2 Project Report

## Applying Machine Learning to Predict Churn for Music Streaming Service KKBox

## 1. Context

KKBox is a music streaming service popular in South East Asia with over 10 million users. It functions on a subscription-based business model, with the majority of subscriptions lasting 30 days. An account is marked as *churn* if there are no new transactions within 30 days after a subscription has expired. KKBox would like to be able to predict which subscribers are likely to renew within a month of their membership ending and which ones will churn.

I accessed data for the project via Kaggle. Information was provided in four .csv files: one containing user demographic information, another with listening history, and a third with transaction history. The fourth table held the target data; a binary column called "is_churn" which marked whether or not a customer had any new transactions within 30 days. This file held information on subscribers who churned in March 2017. Therefore, the goal of this project was to use the information of subscribers who had churned in March to predict which ones were likely to churn in the upcoming month, April 2017.

Understanding churn is key to KKbox's business model. As with all subscription-based services, retaining customers is as key to revenue as attracting new customers, if not more so. By being able to predict churn, KKBox would be able to apply various methods such as targeted advertisements, special discounts, or adjustments to their apps features in order to retain subscribers who are discovered to be at risk of churn.

Data source citation:
KKBOX Group. (2017, September). WSDM - KKBox's Churn Prediction Challenge, Version 2. Retrieved March 3, 2021 from https://www.kaggle.com/c/kkbox-churn-prediction-challenge/overview/evaluation.

# 2. Data Wrangling

In each file, there was a unique identifier for each subscriber called "msno." After reading in each of the .csv files into dataframes, I merged them on this unique identifier so that each row in my final dataframe would represent an individual customer. The transactions data held information on all of the customers' transactions, so I decided that in order to avoid having multiple rows for a single customer, I would choose the most recent transaction. After doing this, I found that some customers made multiple transactions on their most recent transaction date, so I dropped duplicates for those few customers. The members dataset had a unique msno id in each row, which made for a rather simple merge. The user listening history data had about 18 million rows, with each row representing a log of user activity on a given date. I grouped the data by msno id and aggregated by sum in order to get a picture of each subscriber's listening history over the full duration of their membership.

With my four files merged into one dataframe with each row representing a unique customer, I began exploring the data further. I discovered that only one feature, gender, had NaN values. I did not want to lose the valuable information in the over 300,000 rows missing a gender value, so instead of dropping those rows I replaced NaN values by randomly assigning "male" or "female."

Three categorical features, city, payment_method_id, and registered_via had some categories with less than 1% of the total number of rows. I dropped those categories to avoid creating a model that overfit to these rare categories in my modeling phase.

While examining the age column, I noticed that the 25th quartile, median, and 75th quartile were all zero. This most likely meant that zero was placed instead of NaN for subscribers who did not provide their age. I first found the interquartile range of the data with the zeros removed. Then, I replaced those zeros with values randomly chosen from the interquartile range.

For the cost of the subscription service, KKBox provided both the price of a subscription, and the actual price the customer paid. I also discovered several subscribers who paid less than the listed price. I assumed that these customers had been given a discount, and therefore kept them in the dataframe. This would prove useful later on in the modeling stage, as I checked if providing discounts had any effect on a user's likelihood of churning.

After removing a few outliers from the user listening data, I had a clean dataframe and was ready to begin exploratory data analysis.

# 3. Exploratory Data Analysis

I began exploring the cleaned data by looking at the percentage of subscribers who churn in each category for the categorical features. I discovered that almost 100% of the subscribers who used payment method 32 churned. While there were interesting results for the other categories, there none as dramatic as this one. I could immediately determine that a recommendation should be to retire payment method 32 for any new subscribers, if this had not been done already. Unfortunately, as the payment method ids were already encoded as numbers, I could not speculate further as to why such a large disparity existed. Two other notable results from this exploration of categorical data were that first, city number 1 had the most loyal subscribers, with only 4% of subscribers there churning, while around 8% of subscribers in other cities churned. Second, subscribers who used registration method 7 were least likely to churn.

When looking at user listening data, I hypothesized that part of the reason why a user might churn would be because they did not find the app to be useful. Surprisingly, this was not true. Users who churned and users who renewed listened to music on the app at rates which were relatively equal.

Looking at prices also provided a surprising revelation. Users who were given a discount were actually more likely to churn than users who were not. However, as I only had 634 subscribers with a discount in my dataset, I did not immediately conclude that discounts were ineffective.

A plot of churn by registration date revealed that the oldest and the newest subscribers churned at much higher percentages than other users. I reasoned that the high churn for first time users might be common, as it is reasonable to assume that any app might have users who try it out for just a month before deciding that it is not for them. On the other hand, the high churn percentage for subscribers who registered when KKBox was first founded in 2004 evaded explanation. Further research should be done comparing recent app updates and usage trends for these original subscribers to determine if there are new features which may have turned off some of those subscribers who first used the app.

The most significant discovery was that churn was highly correlated with price. Users who churned were on plans that lasted three times as much as users who renewed, and therefore were also paying three times as much. From this, I concluded that switching subscribers to plans with a shorter duration would likely have a big impact on churn and allow KKBox to retain more subscribers.

# 4. Pre-processing and Training

There were two challenges to resolve in the Pre-processing and Training stage: how to transform my data for modeling, and which model to choose. The first transformation was to extract meaningful information from the three date columns before dropping them. I created a feature called “mem_duration” by subtracting the date each user registered for KKBox from their membership expiration date. I used the transactions dataframe to discover how many transactions each user had made. I added one final feature called “secs_per_day” which estimated how many seconds a user spent on the app each day by dividing the total seconds by the total days.

After engineering these additional features, I dropped the date features, scaled the numerical features, and encoded categorical features using one-hot encoding.

I used a dummy classifier to get a baseline against which to compare the metrics of various classification models. This dummy classifier simply predicted that every subscriber would renew. This resulted in an accuracy of 94.45%, as the vast majority of subscribers did renew. However, this high accuracy was misleading, as my focus was on those subscribers who churned. I needed a metric that would highlight the models’ performance on false negatives: subscribers who churned, but were missed by the model. Therefore, I chose recall as my metric.

The linear classifiers performed the worst. Logistic Regression and Support Vector Classifier each scored 0.56 on recall. Tree-based classifiers performed better, with Random Forest scoring 0.64 on recall, and XGBoost scoring 0.65. I decided to use the XGBoost model going forward.

# 5. Modeling

With half a million rows and over forty columns after encoding, running XGBoost locally was too expensive for my machine to handle in a reasonable timeframe. I used the Google Cloud Platform (GCP) to run my model in parallel on eight different Tensor Processing Units (TPU). The advantage in processing speed was remarkable: what took more than three hours to run on my local machine only took about nine minutes on GCP.

I selected learning rate (eta), max_depth, min_child_weight, colsample_bytree, and gamma as the hyperparameters to tune. I used both random search and bayesian optimization to search the hyperparameter space for the best options. Of the two methods, random search did better on the training data with a recall score of 0.67 to bayesian optimization's 0.66. Recall on the test set was extremely close, with bayesian optimization beating out random search by only 0.0006, scores being 0.6423 and 0.6417, respectively. I decided to go with the model tuned by bayesian optimization.

A feature importance chart confirmed what I'd seen in EDA: subscription price and duration were the most important features in determining churn. I decided to model several scenarios to test if making any changes would result in a reduction in churn.

The first scenario I modeled was providing all subscribers at risk of churn with a 50% off discount. This had a slight effect on churn, with the percentage of subscribers predicted to churn dropping . Next, I tested if increasing user engagement, switching long-term plans to monthly plans, and having subscribers at risk of churn sign up for auto-renew would have an effect on churn. Of these scenarios, switching subscribers to monthly plans had the greatest effect on churn, decreasing it by 7.7%. Combining these scenarios led to a predicted 18.28% decrease in churn.

While this decrease in churn is valuable, it did not result in an impressive increase in revenue. Without taking into account the decrease in price that these scenarios require, revenue was projected to increase by 3.25%. With the changes in price taken into account, this drops to only 0.56%.

# 6. Conclusions and Recommendations

We began with this Problem Statement:

**What opportunities exist for KKBox to report a positive percent change in revenue by the end of the current quarter through subscriber retention, attracting new subscribers, or increasing prices?**

While the number of new subscribers KKBox adds per month has increased since 2004, there has been a significant decrease in new subscribers since 2015. Therefore, we cannot rely on attracting new subscribers alone to lead to an increase in revenue.

After modeling various scenarios, we found that by addressing the following, KKBox could decrease churn by 18.28%:

- convincing subscribers at risk of churn to sign up for auto-renew
- restructuring any subscriptions longer than 30 days to be monthly
- provide subscribers at risk of churn with a 50% off discount for one month
- increase user listening metrics

This decrease in churn is estimated to increase revenue by only 0.56%. Therefore, while retaining 18.28% of subscribers who were at risk of churn will have a positive effect, especially in the long-term, additional efforts could be made to increase revenue further.

The final solution explored, increasing prices, appears to be the most promising. Doubling prices would lead to only a 4.25% increase in churn. However, this result is misleading. Doubling the price of the service would likely discourage many subscribers from using the app and increase churn by much more than 4.25%. The data provided is not quite suited to truly answer this question. In order to truly determine the effect of an increase of price on churn, we would need new data. Data on the cost of plans by competing services and on user income could help provide a more accurate prediction.

To achieve the goal of ending the quarter with a positive percent change in revenue, KKBox should implement the scenarios for decreasing churn. Further analysis should be done on ways to address the declining number of new subscribers, through advertisements or special promotional programs. Finally, an increase in plan prices could be another effective option for increasing revenue. Data collection efforts on the prices charged by competing services would help determine if raising prices would be a valid solution.