Cedric Herman
Mentor: Srdjan Santic

Springboard
January 2018 cohort

KKBOX Music Provider Churn Prediction

Contents

\

1) Scope

a) Motivation:

Any subscription-based business is committed to keep their customer happy in exchange for their loyalty. Despite their efforts, some customers will NOT renew their subscription. In this latter case, if a customer has not renewed within a time window after its subscription expiration date, this customer is said to have churned. Although each service provider offers several subscription options (monthly, yearly, basic, premium...), the time window to consider a customer has churned varies quite a lot. It seems that each company sets their own.

This is a common issue for businesses where machine learning has proven to be effective. It is much costlier to attract new users rather than keeping your current subscribers. Therefore, customer behavioral analysis through data analytics is an investment that pays off. Becoming familiar with this type of dataset is highly valuable.

b) Description

This project is based on a Kaggle competition where KKBOX, an Asian leading music streaming service, is interested in predicting their churn rate on a monthly basis. The time window is 30 days in order to consider a customer has churned after its current membership ends.

c) Data

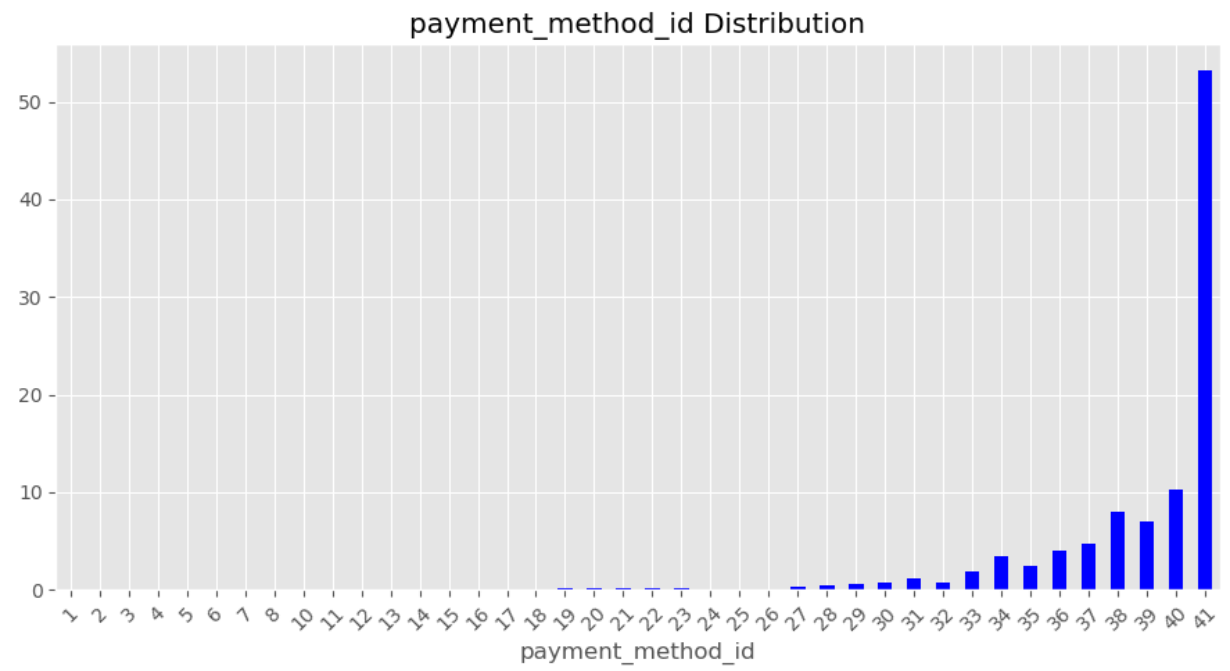
Most of the time, data from Kaggle competition comes pre-cleaned and arranged in an analysis ready format. This is not the case. There are multiple files that contains different information and in the data description it clearly states there are some unrealistic entries. The largest file size is nearly 30 GB for a total of about 33GB. More details can be found here.

2) Data Wrangling

KKBOX (streaming music service provider) made available 3 data files to get to know their customer base behavior. Transaction CSV file contains transaction history from January 1st, 2015 to February 28th, 2017. Information about plans subscription and prices are included. User log CSV file captures user's daily activity such number of songs played from January 2015 to March 2017. And finally, member CSV file features demographic information on users who created an account on KKBOX.

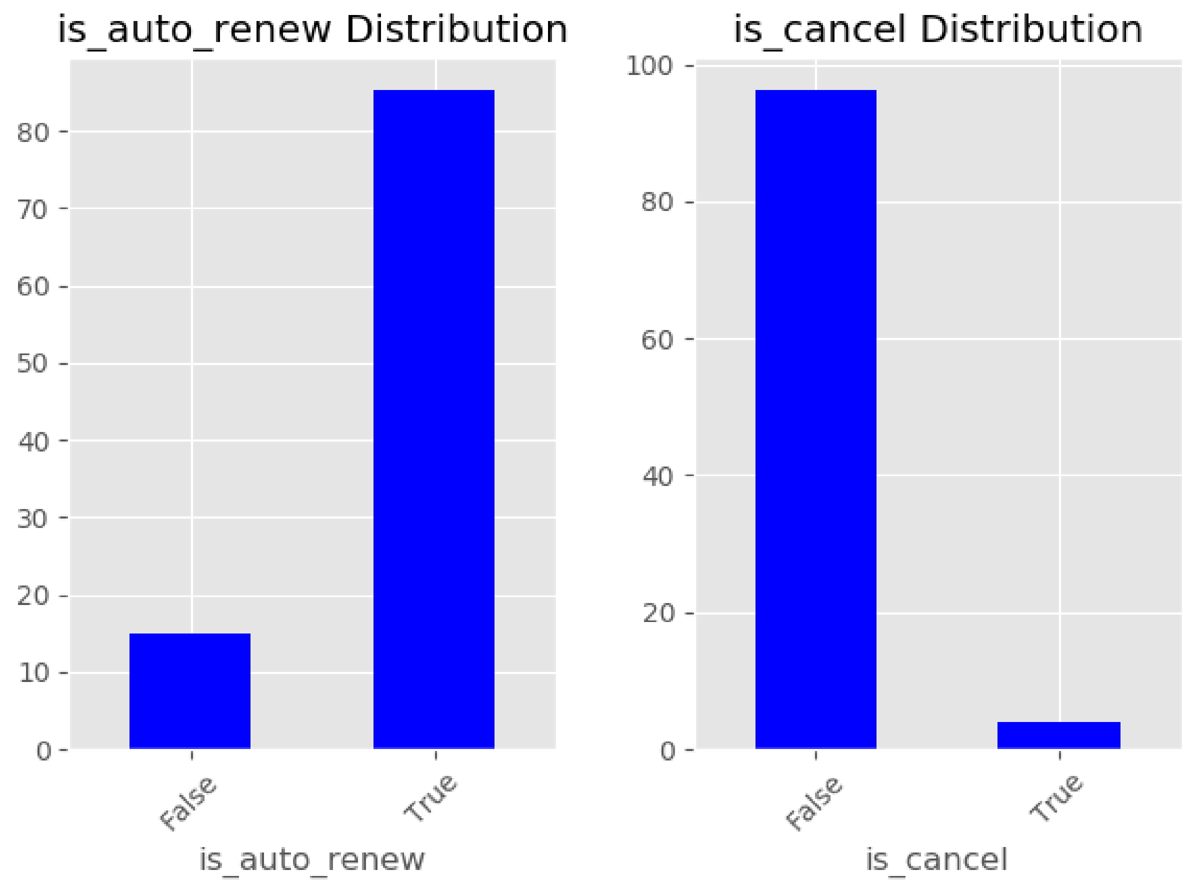
a) Transactions

Based on roughly 21.5 Million transactions, more than 50% are completed using payment ID 41. Unfortunately, payment methods are represented by numbers only, so we don't know what it means. Intuitively, this may be representative of credit card automatic payment.

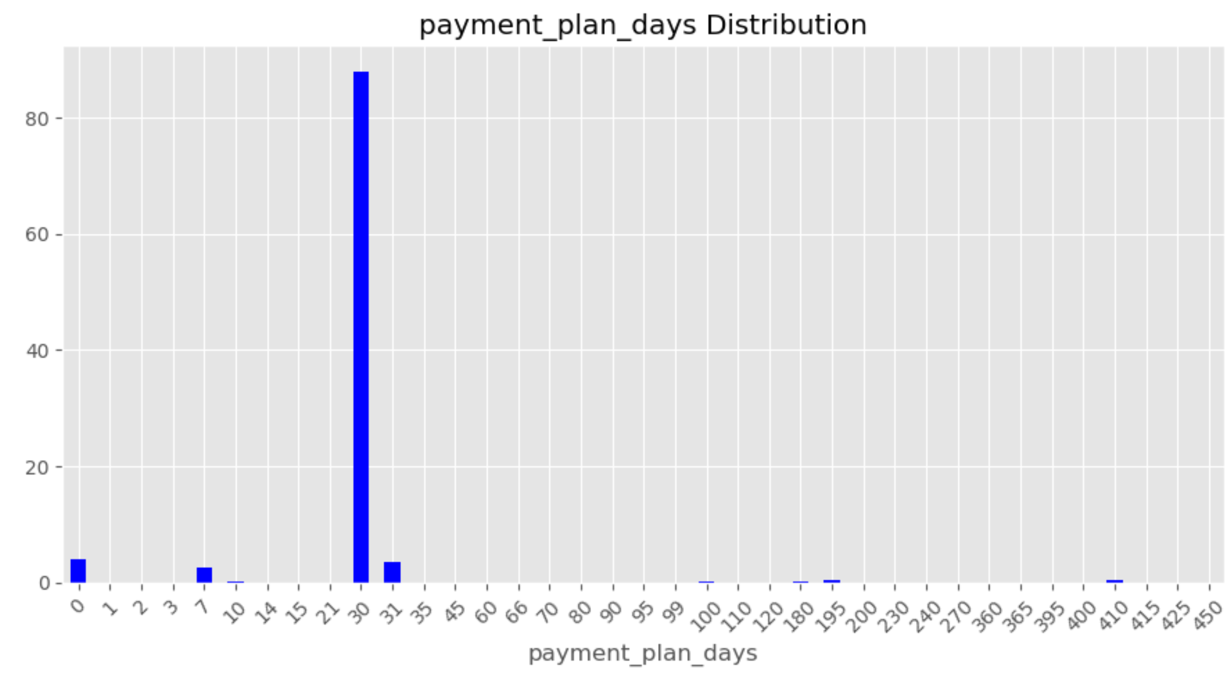


In this 2 years and 2 months period, membership automatic renewal is authorized 85% of the time. Looking at KKBOX website and using google translate (it's in Mandarin), there is a discount when automatic renewal is active which can explain its popularity. Users are allowed to cancel at any time, this is recorded as a Boolean. An overwhelming majority of transactions do not show a cancellation (~95%). This is consistent with the fact that churn rate is very low (e.g. 6% in

February 2017). Also, it happens that a user chooses to cancel its current plan to join another one.



Plan's duration in days shows a wide variety compared to KKBOX website where 30 days, 90 days, 180 days and 365 days are listed. However, KKBOX does offer bonus days which adds multiple combination and their special offers may have changed over this 2 years period. The most popular plan by far is 30-31 days followed by 7 days. It extends up to 450 days. Note this data has 0-day plans which seems erroneous, it represents 4% of all transactions. Taking a closer look at 0-day plans, all of them are also missing list prices which have a value of 0 (We will worry about list prices later).



To replace 0-days payment plan, we need to consider multiple scenarios:

- User renews on time
- User renews late
- User is actively cancelling his/her plan

When a user renews on time, it means there is an overlap between the current transaction date and its prior expiration date. We will take advantage of this fact and thus use the time difference in days between successive expiration dates.

When a user waited past his/her membership expiration date then this is an independent transaction. Hence, current transaction date and expiration date time difference should tell us the plan duration.

Finally, when a user decides to cancel his\her membership, one can use the most recent plan he/she subscribed to before leaving.

At this stage, one may noticed some estimated plan duration were negative! It turns out some user's transaction history did have expiration date which went back in time. This behavior cannot be explained, it is most likely an error during data collection. Those events were discarded.

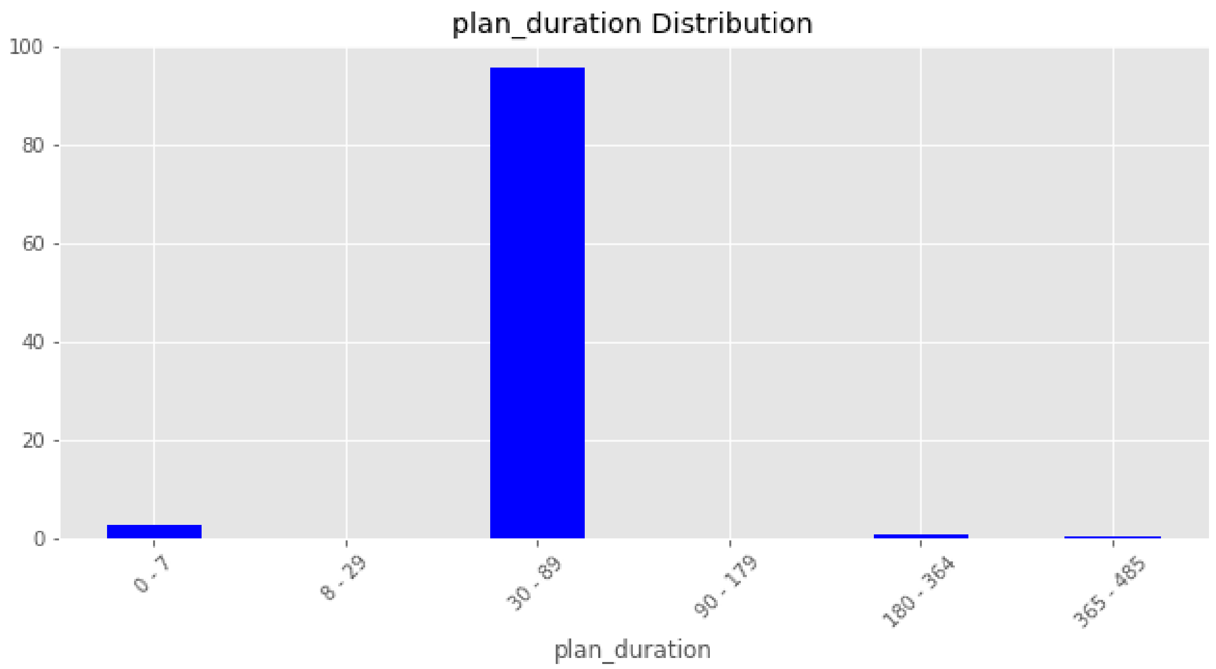
Furthermore, when there is a missing value for payment plan days on a user's first transaction, we do not have any prior history to estimate its plan duration. Users having a unique transaction leaves us clueless so we are removing them. For multiple transactions history, when both first and second transactions are automatic renewal then it must have the same payment plan days. Except the latter scenario, there isn't any other case where we can be sure of the right payment plans days. Therefore, we wil discard all other user's first transactions (total of ~95k). Note that those transactions, including unique transaction, are still useful to determiner whether a user has churned.

Our estimated plan duration can be adjusted for monthly subscription. Because 30 days plans actually means monthly membership, our estimation varies between 28 to 32 days depending on the number of days in each consecutives month. Plan duration have been corrected to take this fact into account. Even the raw data has 31 days plan where it really is 30 days.

At this point, we actually created more payment plan than we had in our data even after monthly membership adjustement. To reduce the number of plan, we can group them by intervals:

| 0 - 7 | 8 - 29 | 30 - 89 | 90 - 179 | 180 - 364 | 365 - 485 |

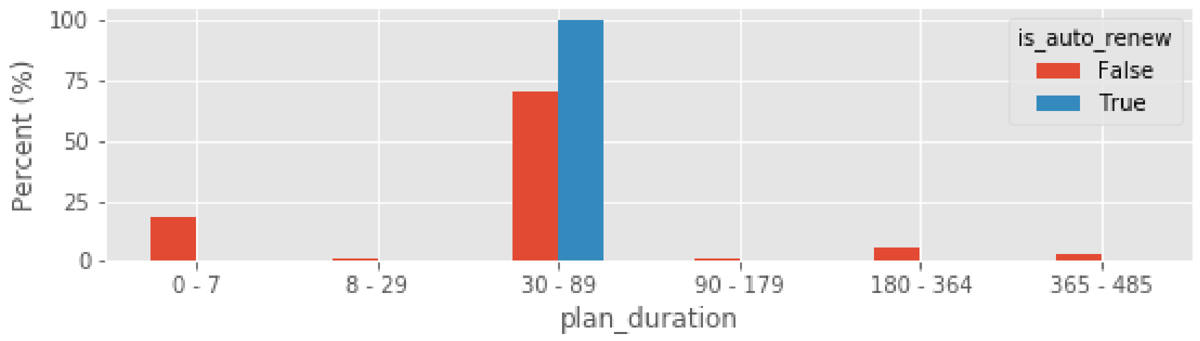
Those intervals are based on the information available on KKBOX website. For instance, there is a 30 days and 90 days plan hence we make an interval between 30 and 89 days. The assumption is you can only have extra days in your subscription.

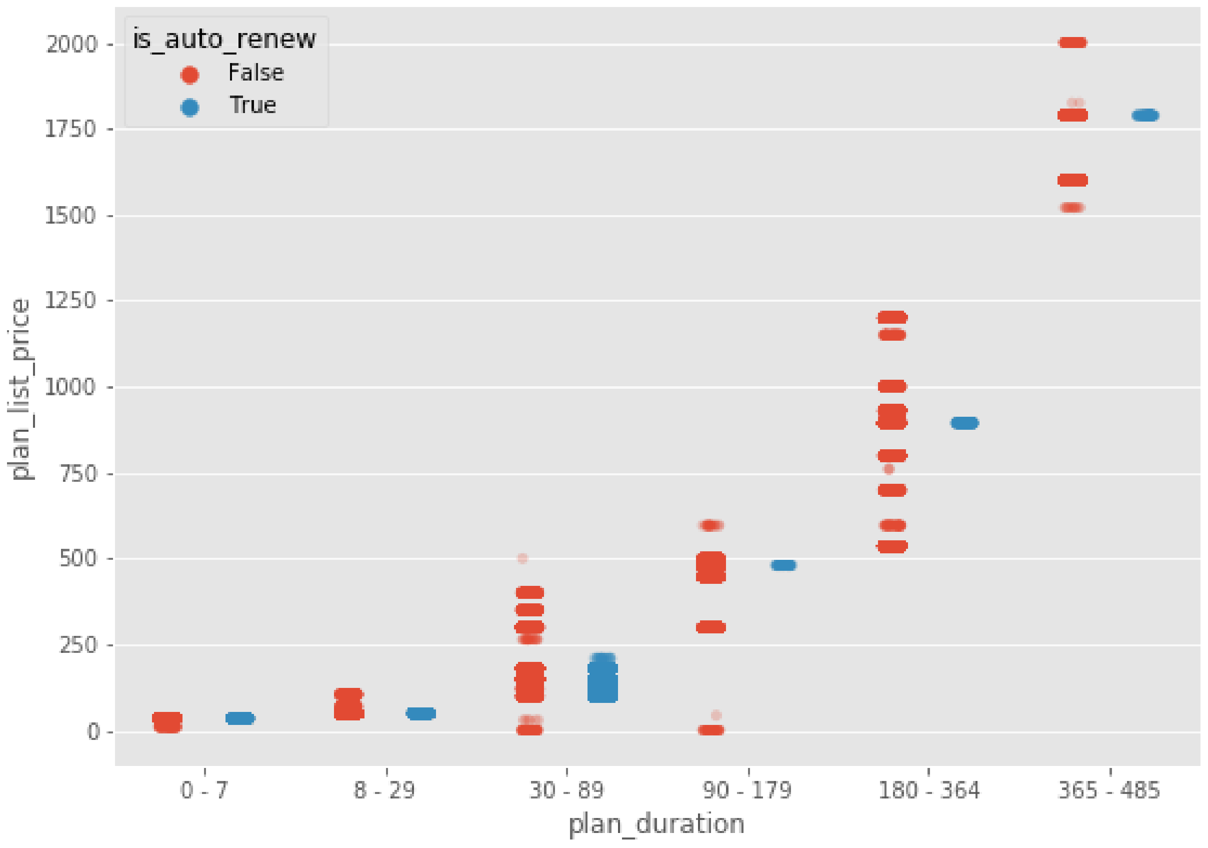


Plan list price remains below 150 NTD (New Taiwan Dollar) for 90% of all transactions. Most frequent prices are 149, 99 and 129 in descending order. There is a non-negligible number of 0 NTD list price (around 8%) which indicates missing values. Indeed, plan list price should always have an amount for each available plan regardless of how much customers actually paid. We saw earlier that missing payment plan days also had missing list price. As a result of filling payment plan days, we have new plan duration. So we should find a mapping between plan duration and list price based on our data as opposed to payment plan days. Otherwise, we won't be able to map every plan duration to a list price. Each of the six plan duration intervals has been linked to the most significant list price as shown below:

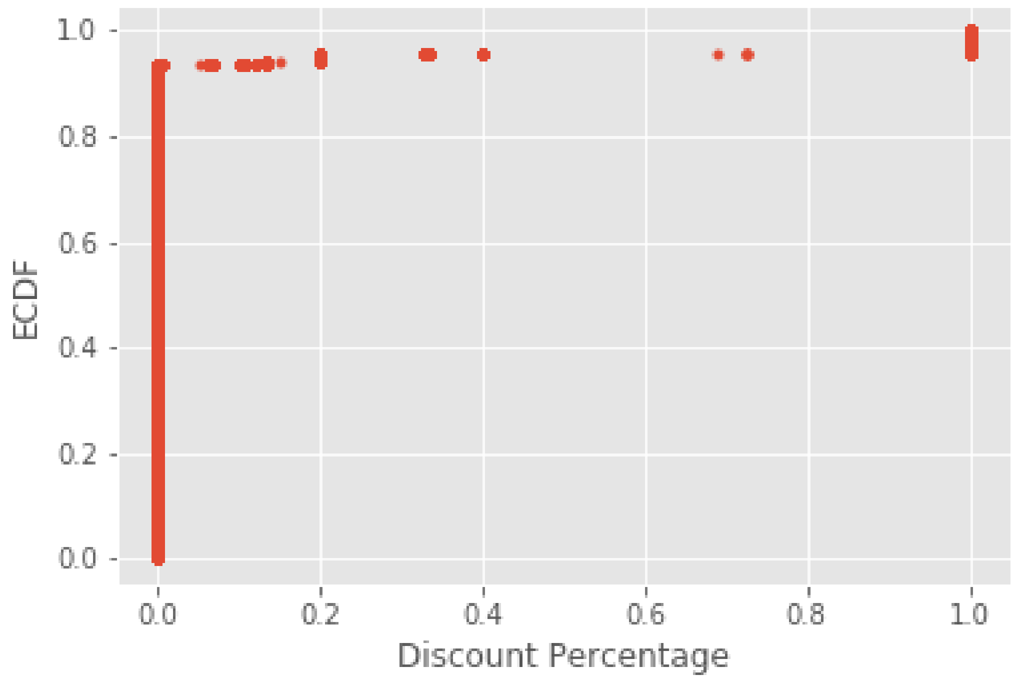
Plan Duration in days	Plan list price in NTD
0 - 7	35
8 - 29	50
30 - 89	149
90 - 179	480
180-269	894
270 - 364	1200
365 - 485	1788

All missing values have been replaced at this stage. Let's take a look at plan list price as a function of plan duration. As expected, longer plans tends to be more expensive. Interestingly, there are overlaps in list price except for long term plans (> 365 days) which are clearly more costly.



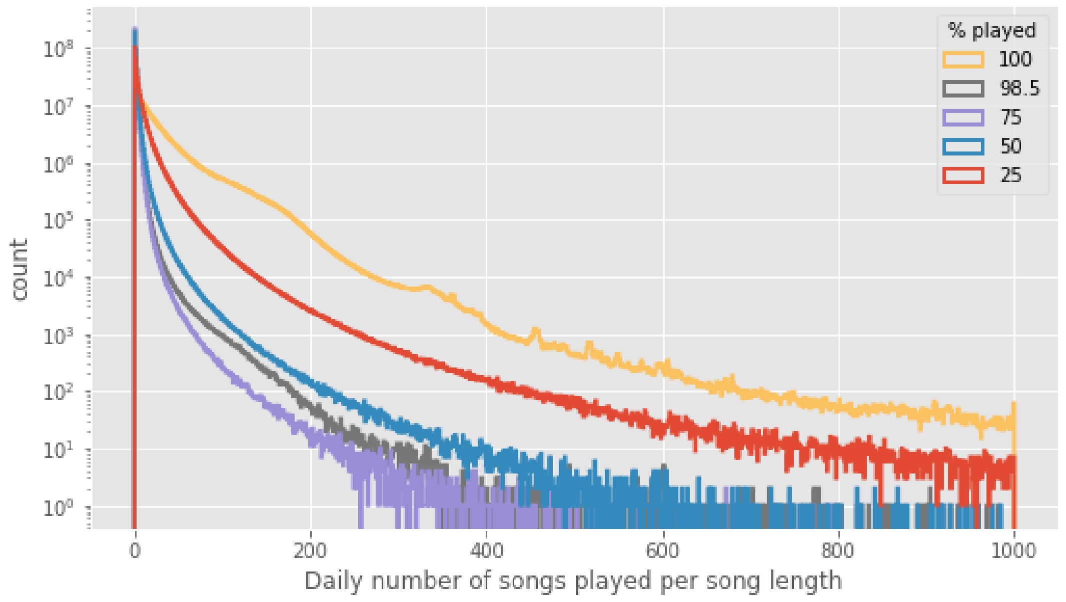


Finally, we will look into the difference between list price and actual amount paid. Is there any discount? Any trend? As a matter of fact, getting a discount is pretty rare. Less than 8% of transactions have discount! Surprisingly, there are negative discount! This means the actual amount paid is greater than list price. To correct for this anomaly, list price were matched to their actual amount counterpart.

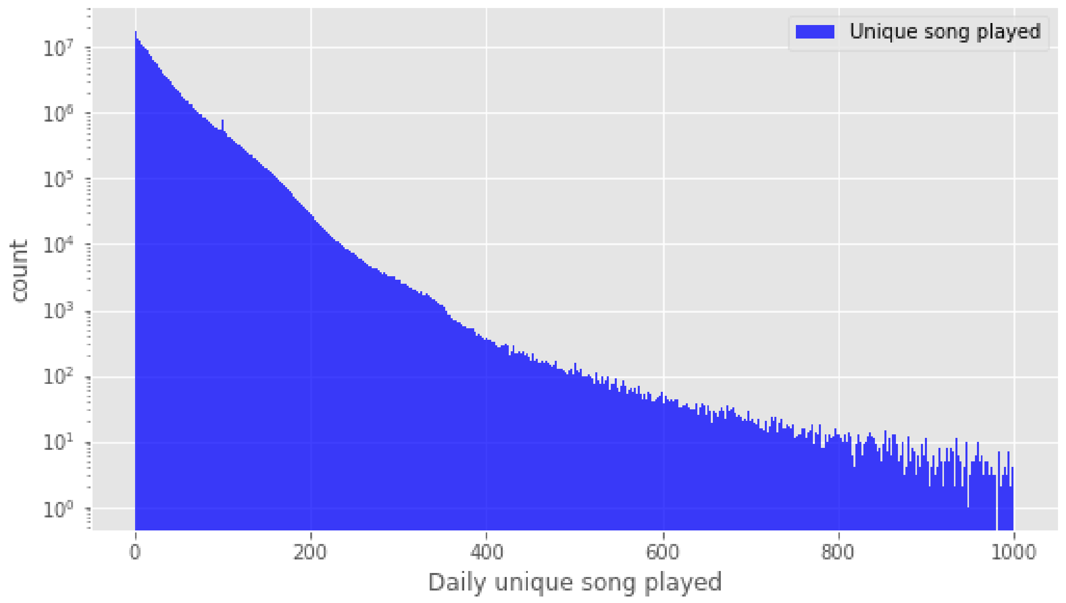


b) Users Activity log

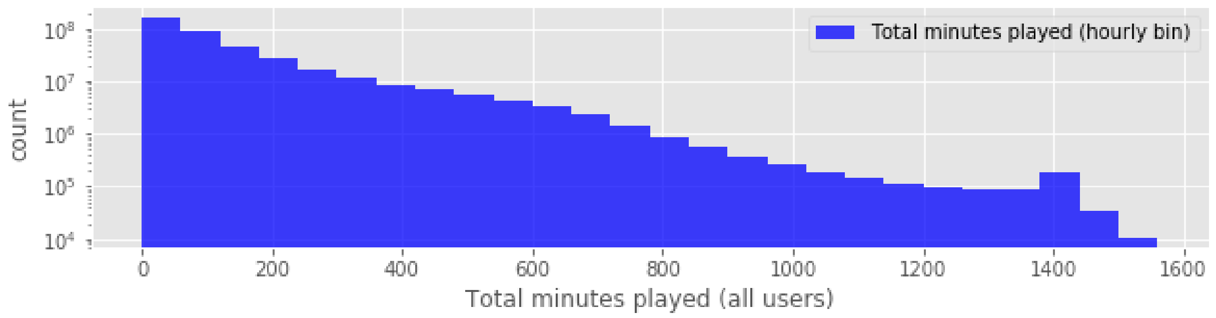
Users log records multiple metric related to user's listening habits. For instance the number of songs played per percentage of song length quickly decays as shown on the figure below. Percentage of song length are intervals. At 50% it represents number of songs played between 25% to 50% of song length. As the number of songs increase (>8 songs), there are more people listening to the whole song compared to 25% second followed by 50%, 98.5% and 75%.

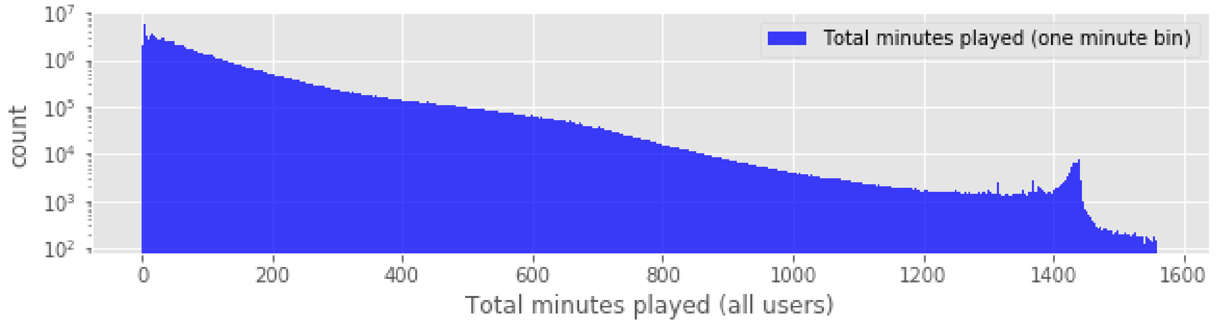


If we take a look at the number of unique songs played, it also decreases rapidly indicating most people don't have very long playlist.



The total number of seconds played is consitent and declines as well. Surprisingly, there is a small peak (plots are log scale) at 24h and more data beynd 24h which should not happen as it is a daily activity log. One scenario to explain this 24h peak would be 24/7 shops which relies on this streaming service for their background music. The most frequent time is 3-4 minutes of daily music. There are some peak forming every 15min which probably means some people have busy schedule and just listen to a playlist that accommodate their free time.





The last interesting plot is to look at the total listening time per day. Overlapping the number of active users each day, we can clearly see that total listening time and number of active users correlate very well. Some values were extremely negative or positive which were filtered out by making sure total listening time is between 0 and 24h.

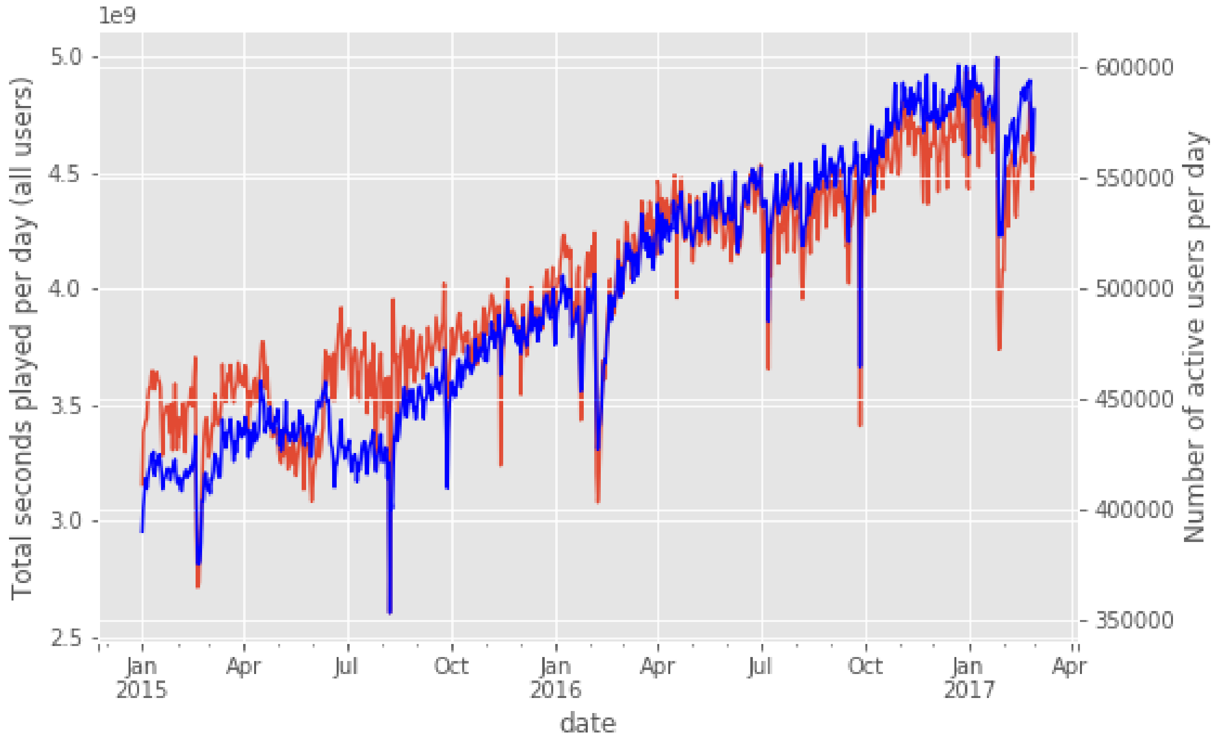
Let's test if the pearson correlation coefficient of 96% is not a result of chance. Our null hypothesis is that this correlation is actually zero. The alternative hypothesis is that the daily number of users is highly correlated with the daily listening time.

Null hypothesis: number of active users and total listening time are independent.

Alternate hypothesis: there are correlated

We used hacker statistics to compute replicates of this data and compute the probability to obtain a positive correaltion at least as extreme as we observed by chance. This is the definition of the p-value.

Our p-value was so small we couldn't find any instance in our 1 million replicates where we had such a correlation by chance. Therefore we can safely reject the null hypothesis. There is significant evidence that daily number of users and daily listening time are strongly correlated.



There are multiple drops occurring in user's count and more so in listening time. Late January drops corresponds to Chinese New Year which last 5 days not including new year's eve.

Another periodic drop is on Father's day (always August 8th). Note that in 2015, father's day was on a Saturday while father's day in 2016 was on a Monday. Thus we can see a lesser drop on Sunday, August 7th in 2016.

Decline on September 28th, 2015 corresponds to mid-autumn festival. It does repeat in 2016 with lesser effect (September 17th).

The closest event to the large drop on September 27th, 2016 was Teacher's day (birthday of Confucius) that took place on Sep. 28th, 2016.

There is an extended drop in May 2015. It could be due to technical malfunction (e.g music app not working for iphone for instance).

3) Feature Engineering

In this section, we will discuss what predictors we can use based on our data. Regarding user's activity, song counts were aggregated on a monthly basis. In addition, we will keep data for the preceding 6 months in respect to our prediction month. Indeed, we don't want to go too far back in time as user's experience may have change over time. Many companies have software as often as twice a year. Table 1 summarizes what features are considered from user's activity in a given month.

Table 1: Features from User's activity log on a given month

Feature name	**Description**
num_25	# of songs played < 25% of song length
num_50	25% ≤ # of songs played < 50% of the song length
num_75	50% ≤ # of songs played < 75% of the song length
num_985	75% ≤ # of songs played < 98.5% of the song length
num_100	98.5% ≤ # of songs played < 100% of the song length
cum_25	# of songs played up to 25% of the song length
cum_50	# of songs played up to 50% of the song length
cum_75	# of songs played up to 75% of the song length
cum_985	# of songs played up to 98.5% of the song length
cum_100	# of songs played
num_unq	# of unique songs played
total_secs	total listening time in seconds

Song counts and total listening time were normalized by the number of days per month. Based on transactions, we will derive multiple features listed in Table 2.

Table 2: Features based on transaction history

Feature name	**Description**
time_of_month	Fraction of number of days left in expiration month. It tells us about what day of the month the current membership expiration date was set (regardless of active cancellation)

churn_zone_days	Number of days in or out of churning zone at the end of prediction month. This is the time difference between current expiration date and end of the prediction month based on the most recent transactions prior to our prediction month.
cancel_ratio	Ratio of active cancellation to number of transactions. This is an indicator of how many active cancellation occurred in each user entire history.
auto_renew_ratio	Ratio of auto renew to number of transactions. This is an indicator of the number of automatic renewal transactions over each user entire history.
uninterrupted_days	Number of uninterrupted days of membership. It will tell us how long each user has been renewing on time up to our prediction month.
plan_duration	Last Plan duration which is an interval determined during Exploratory Data Analysis.
pay_id	True if most recent payment ID is 41, False otherwise

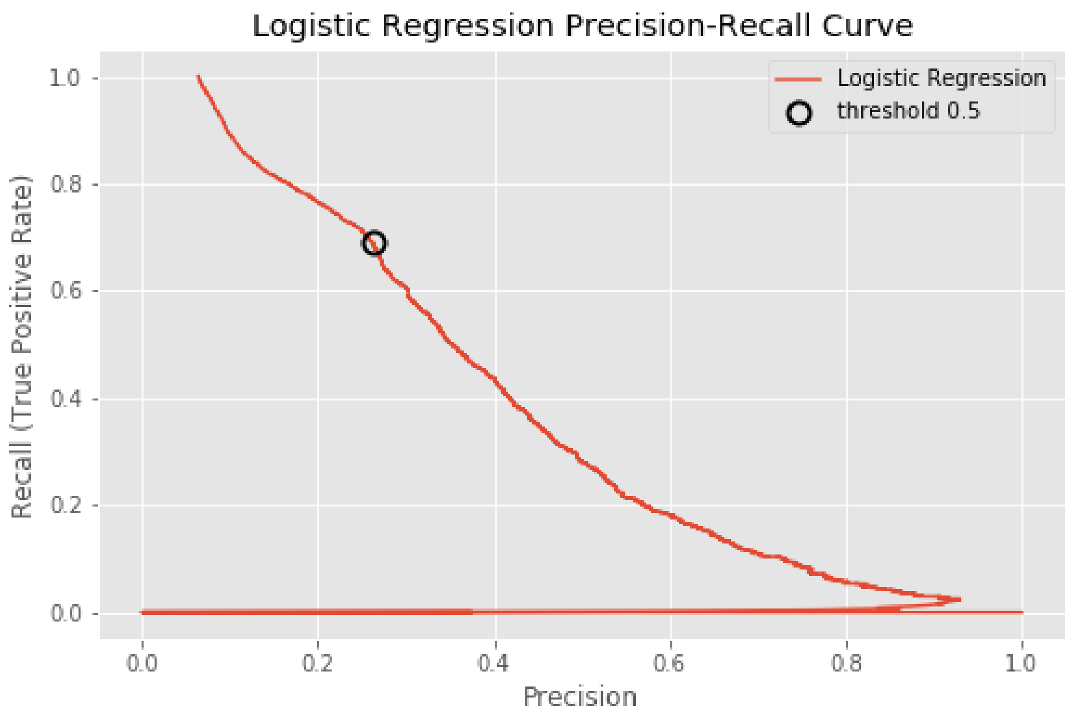
4) Model Building

Our model contains 19 fundamental features. However, after one-hot encoding of plan duration and feature extention to prior 6 months, we will have 67 columns in our data matrice. Because of the class imbalance, we will assign larger weights to our minority class so that our loss function will be penalized more heavily for not identifying people who churned correctly. Another approach would be to use SMOTE (Synthetic Minority Over-sampling TEchnique) to upsample the minority class by adding synthetic samples. SMOTE uses kNN so proper scaling is required. Following SMOTE, we can also apply ENN (Edited Nearest Neighbors) to clean up outliers and boundaries resulting in a better model.

For practical reason being it will takes days to tune our hyperparameter using SMOTE and ENN, we will focus on class weights. In addition, Logistic Regression will be used and its sag/saga solver will be faster for our data size (893,637x67) in comparison to liblinear (closed form). Because we use gradient descent, scaling will speed up our loss function minimization. Having a mix of contineous and binary features, we will use Min-Max scaler so that binary features will remain 0 or 1 and all others will be scaled between 0 and 1.

During hyperparameter tunning, we selected f1 score so that we will focus on identifying customers who are most likely to churn and minimize the number of false positive/false negative. This metrics focuses on the positive class (i.e. churn in our case which is also the minority class). An alternative metric would be average precision which represents the area under the precision-recall curve.

Our optimized model based on f1 score yields the following precision-recall curve:



Therefore, one has to be to choose a threshold that provides the best tradeoff between precision and recall. The default threshold of 0.5 does favor recall at the expense of precision. This is actually desirable since false negative are minimized such that we will not label churn as no churn. Many users who were not going to churn may be labelled as churn but users who churned are mostly identified. It is much more costly to lose a user rather than trying to retain a user who is loyal.

5) Conclusion

Based on our most predictive features, users who have actively cancelled their membership multiple times and played less unique songs during the prior month are more likely to churn.