

SPRINGBOARD DATA SCIENCE COURSE

CAPSTONE PROJECT 2

**Building a music recommendation system using
machine learning techniques**

Abhishek Sukhadia

12th April, 2020

Contents

1. Introduction.....	1
2. Data Acquisition and Cleaning.....	1
3. Feature Creation and Outlier Analysis.....	4
3.1 Feature Creation	4
3.2 Outlier Analysis	5
4. Exploratory Data Analysis	7
4.1. Song Source.....	8
4.2. Gender	10
4.3. Song Language and City	10
4.4. Count Features.....	11
4.5. Date and Time Features.....	12
4.6. Song Type.....	13
4.7. Registered Via	14
4.8. Numerical Features.....	14
4.9. Artist and Song Frequency	16
4.10. Summary and Key Findings	17

1. Introduction

In the earlier days, we used to buy products based on recommendations from our friends and family and the products they trust. Before the digital age, that has been the default method of purchase. But now with advent of digital age, this is changing as the circles have expanded to more than just friends and family. Companies like Amazon, Flipkart, Spotify and Netflix have developed recommendation systems that would recommend product to their customers. These systems filters data using different algorithms by capturing past behaviour of a customer and then recommends the most relevant items to the users. This project's focus is to build a similar recommendation system for music industry.

With mobile devices and the internet coming in, offline music CDs, music players have gone extinct. Online streaming has become the new norm for today's generation. This has led to music going online and customers getting access to millions of songs. The number of songs available exceeds the listening capacity of a single individual. People at times feel it is difficult to choose his favourite song from millions of songs. Thus customer convenience, efficient management of songs/playlists and personalized music discovery have become the need of the hour. This has given rise to companies bringing in music recommendation systems to improve customer experience.

Lots of platforms like Spotify, Saavn, Apple Music, etc. are gathering lots of data based on users' listening habits, song info and user info. These huge amounts of data paved the way for machine learning algorithms in building a good recommendation system. Lots of research is also been done on music recommendation system.

2. Data Acquisition and Cleaning

I acquired the data from Kaggle which is freely available. The dataset is part of a competition (WSDM 2018) where the challenge is to build a better music recommendation system using a donated dataset from KKBOX. The data set consists of information of the first observable listening event for each unique user-song pair within a specific time duration. Metadata of each unique user and song pair is also provided. The train and the test data are selected from users' listening history in a given time period and split based on time period. The dataset has over 50lakhs rows and more than 30 features.

Each row is a single encounter of a customer's listening event with details given below.

Customer/Members Characteristics:

- Membership No, City, Gender, Registered via, Age, Registration and expiration date for membership

Song Details:

- Song id, Song length, Genre Ids, Artist Name, Composer, Lyricist, Language

Extra Song Info:

- Song Name, ISRC code

Song Listening Characteristics:

- Source System Tab, Source Screen Name, Source Type

Target:

- Whether there are recurring listening event(s) triggered within a month after the user's very first observable listening event

The above data was downloaded as a CSV and then imported in python. More details of this can be found in this [IPython Notebook](#). The training data had 7377418 rows while testing data had 2295971 rows. Post importing, data was pre-processed using the below mentioned steps.

- 1) Merging and Transformation: The different features like song and user characteristics are present in different tables. Thus there was a need for merging into one table using the columns song_id and membership no (msno) as key ids. Songs, members and songs_extra tables were merged with the training and testing data. Registration and Expiration date was imported as object class. So, we transformed these columns to "datetime" type so working with it becomes easy.
- 2) Missing Value Analysis: [Table 1](#) is the summary of the missing values. Lot of important features have missing values.

	feature_name	null_values	%missing	category_count
0	lyricist	3178798	43.09	33889
1	gender	2961479	40.14	3
2	composer	1675706	22.71	76065
3	isrc	577858	7.83	269761
4	source_screen_name	414804	5.62	21
5	genre_ids	118455	1.61	573
6	source_system_tab	24849	0.34	9
7	source_type	21539	0.29	13
8	name	1457	0.02	234145
9	song_length	114	0.00	60267
10	artist_name	114	0.00	40583
11	language	150	0.00	11

Table 1: Features with missing values

Testing data set had similar missing values but more in volume. (Table 2)

	feature_name	null_values	%missing	category_count
0	lyricist	1224744	47.90	24911
1	gender	1052224	41.15	3
2	composer	619304	24.22	52307
3	isrc	196643	7.69	170988
4	source_screen_name	162883	6.37	23
5	genre_ids	42110	1.65	502
6	source_system_tab	8442	0.33	9
7	source_type	7297	0.29	13
8	name	778	0.03	154717
9	song_length	25	0.00	45658
10	artist_name	25	0.00	27564
11	language	42	0.00	11

Table 2: Missing Values in Testing Data

Missing values were handled differently for each feature.

- **lyricist, gender and composer** have >20% missing values. These columns are important features so dropping these columns is not an option. We cannot also replace the missing values with mode. So, we created separate categories as “missing” for these features to tackle this.
- **isrc, source_screen_name, source_type, genre_ids, source_system_tab, name and artist name** have 0-10% missing values. Most of the columns being important and categorical in nature, missing values were considered as a separate category and rows were not dropped.
- **Language** has some values as -1. This value is assumed to be for missing values or where language is not known. So, missing values of language were replaced by -1.
- Missing values of **song_length** were replaced by median song length

Another interesting thing observed was the the missing values between few variables were correlated. Eg. 90% correlation between missing values of artist name and language were observed. We can infer that artist name was determining the language of the song. Source type and Source_system tab, composer and lyricist were other correlations that were found (Figure 1). Though there was correlation, this was not used to fill the missing values.

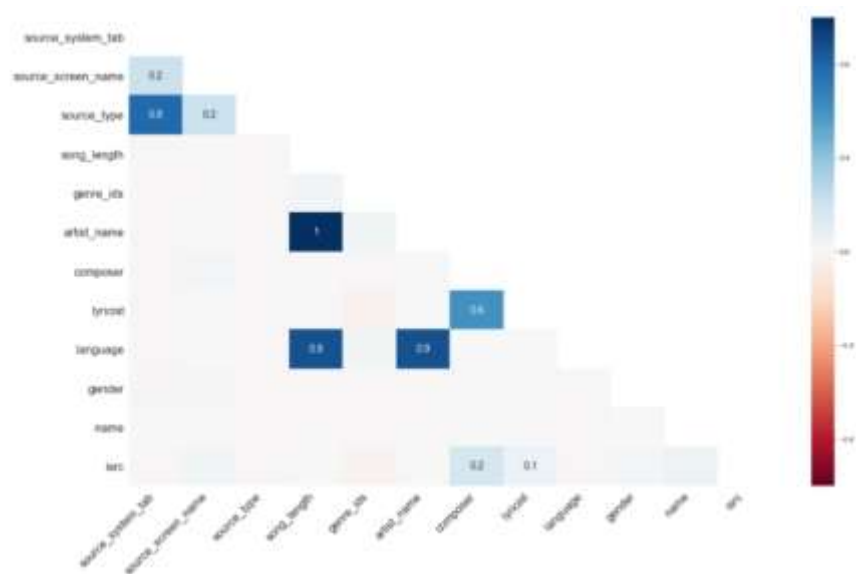


Figure 1: Correlation between missing values of different features

3. Feature Creation and Outlier Analysis

3.1 Feature Creation

We had a limited set of features and most of them are categorical variables like song name, artist name etc. These makes the problem complex and hard for algorithms to learn. So, to provide more information, we manually extracted many features from existing variables. They are listed as below.

Features Based on Songs Features:

Sr. No.	Feature Notation	Feature Description
1	lyricist_count	Number of Lyricist for a song
2	artist_count	Number of Artists for a song
3	composer_count	Number of Composers for a song
4	genre_id	Number of genres for a song
5	count_song_played	Number of times the song was played
6	song_freq	Cumulative count of the song played over time
7	Count_artist_played	Number of times an artist was played
8	artist_freq	Cumulative count of the artist played over time
9	composer_artist_lyricist	Whether composer, artist, lyricist were same or not
10	composer_artist	Whether composer and artist were same or not

Features based on Time:

Sr. No.	Feature Notation	Feature Description
1	duration	Membership duration i.e. difference between registration and expiration date
2	registration_year	Year of membership registration
3	registration_month	Month of membership registration
4	registration_day	Day of membership registration
5	expiration_year	Year of membership expiration
6	expiration_month	Month of membership expiration
7	expiration_day	Day of membership expiration

Other features:

Sr. No.	Feature Notation	Feature Description
1	age_of_song	Created from isrc year. Difference between 2017 and the isrc year. Recency of the song being registered with isrc
2	song_type	Whether the song is long (>6 mins) or short (<3 mins) or medium (3 to 6 mins)

Many more features can be created but this being a huge data set and having limitation on computational power, the exercise was stopped after creating above mentioned features. More Details of the feature creation is available in this [IPython Notebook](#).

3.2 Outlier Analysis

The numerical variables in the data set were age, song_length and all the count features that were extracted above. [Figure 2](#) and [Figure 3](#) shows the distribution of all numeric variables in the dataset.

Age shows very abnormal distribution having values more than 1000 and negative values. A peak is also observed at zero value. It is inferred that zero is missing values and other outliers are data entry problems. We then cleaned the age using the following rules.

- Making Negative values as positive values as they are data input error
- (> 1000) values are data input error so subtracting 1000 from such values
- For age between 1 to 10 and >110 considering them as missing values (0). It is assumed that oldest persons living more than 110 are rare and children with less than 10 years are not matured enough to use a music app.
- Apart from that there are lot of values with age as zero (0). These are mostly missing age values and kept as it is.

Composer count also showed some abnormal values more than 100. A song can rarely have so many composers. Upon checking that, it was found to be erroneous data and cleaned. Similar checks was done for all the count variables for the highest values and they were

cleaned. Duration also had some negative values, which were replaced by mean duration. These negative value errors were due to incorrect registration or expiration date.

Figure 4 shows the distribution of variables post outlier analysis. This clean data is now ready for exploratory analysis. **Note:** All the above cleaning replicated for test data set also.

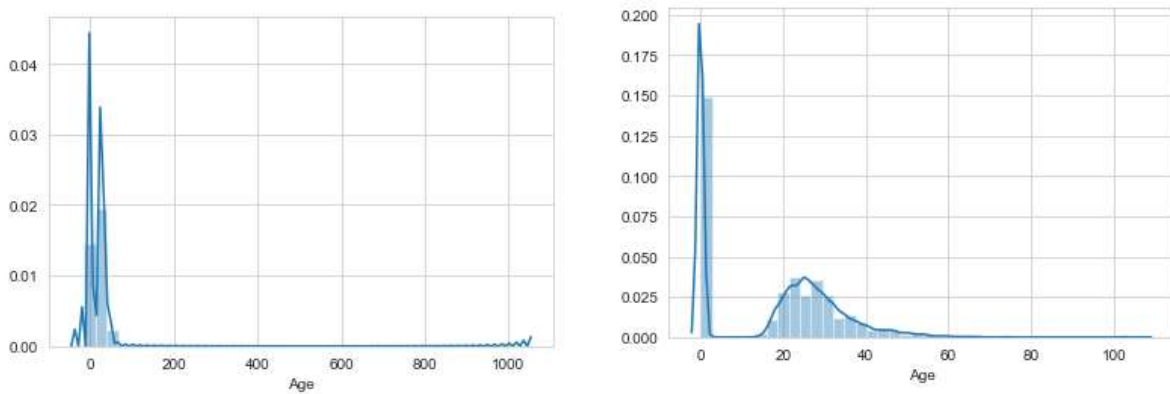


Figure 2: Original and Cleaned Distribution of Age

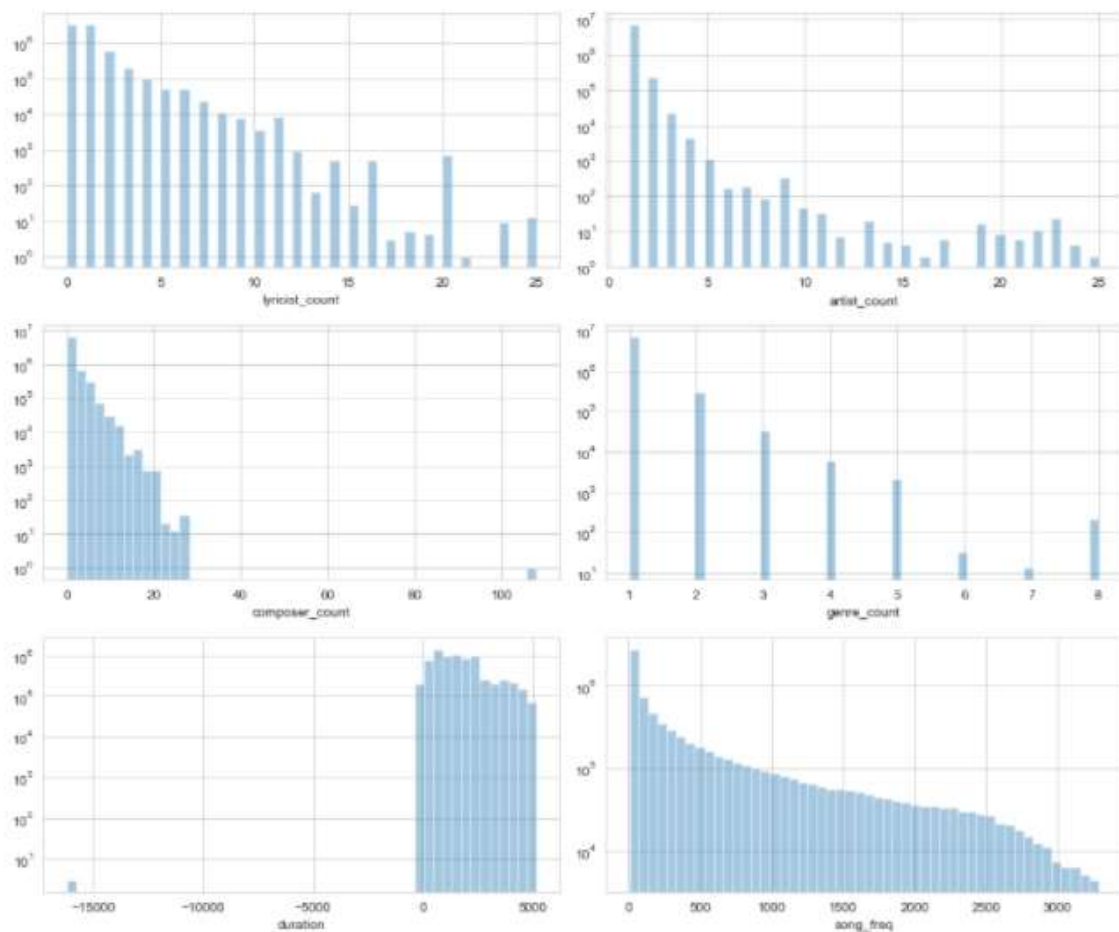


Figure 3: Pre-Outlier Analysis Distribution of Variables

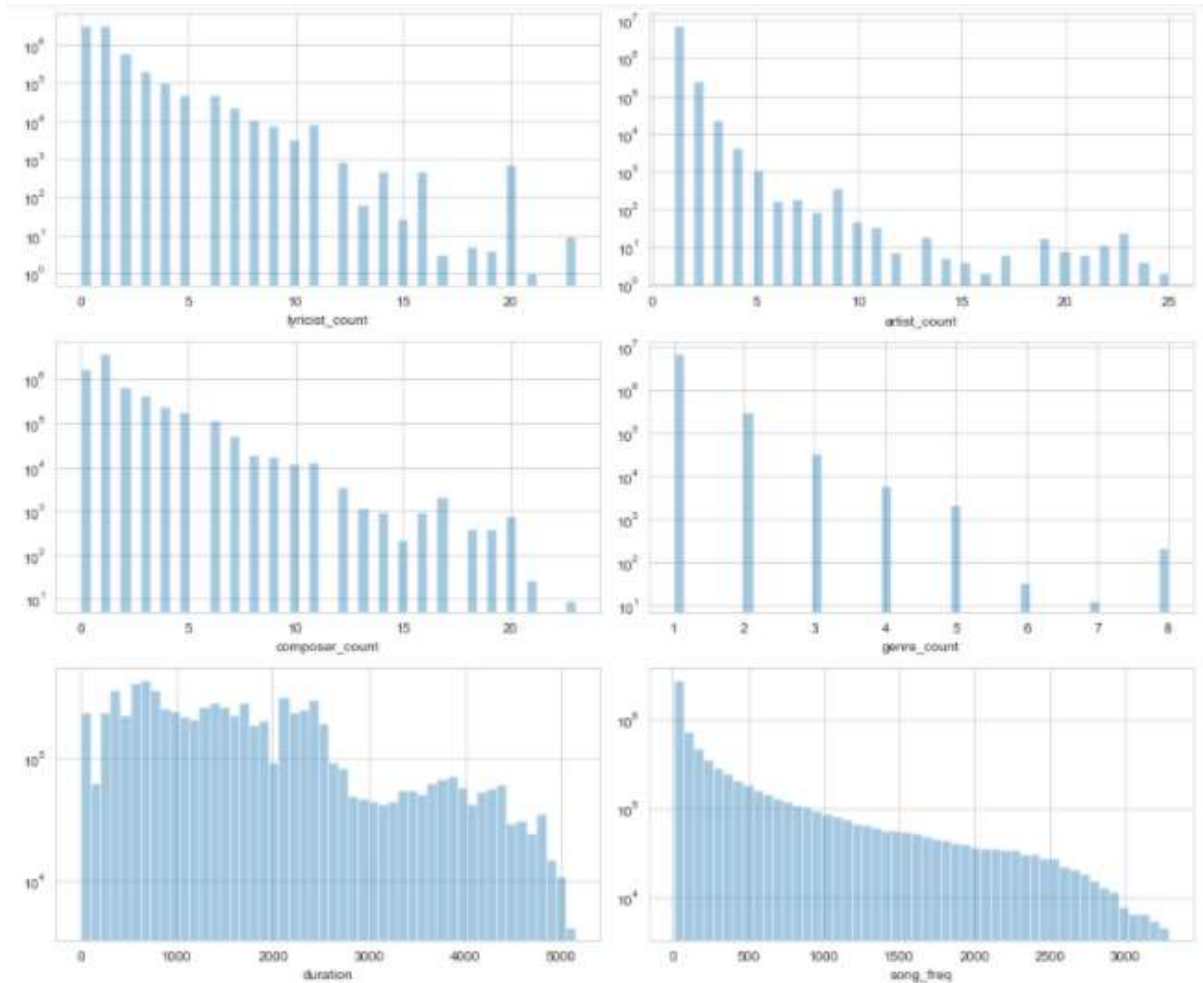


Figure 4: Post-Outlier Analysis Distribution of Variables

4. Exploratory Data Analysis

In the cleaned data, there are now 7377418 records, 36 features/columns and one target variable. We will go through most of the features in the dataset to explore the relation with the target and the listening habits of the users.

The target variable for this project contains two values:

- “0” : the song was not listened again within 30 days of the first observable event
- “1”: the song was not listened again within 30 days of the first observable event

The distribution of target variable in [Figure 5](#) shows that our dataset is fairly balanced. Thus we will be not needed to use techniques to solve imbalances.

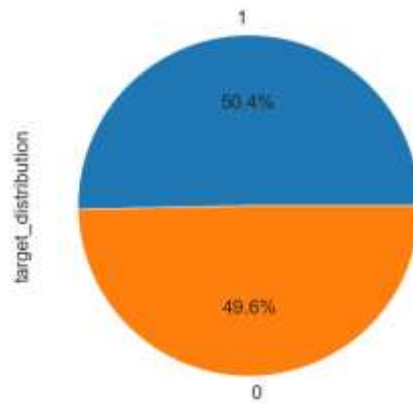
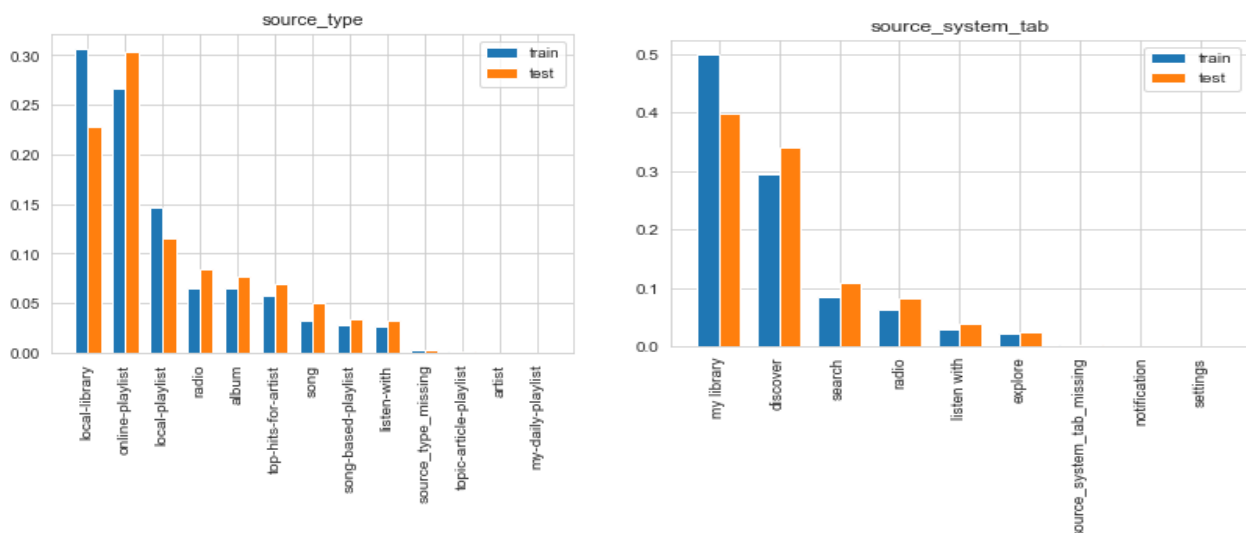


Figure 5: Distribution of Target Variable

4.1. Song Source

A user can access songs on the app/web is through various sources. They are source_type, source screen name and source system tab. Based on intuition, a user journeys through these sources to listen to his song of choice. The category names of these features also hints towards that. We observe in Figure 6 that for this category of features, there is change in distribution between training and testing data. Shift is from listening songs in local library to online playlist and discovering. This tells us that there is a change in listening habits over time. This tells us that our model should be robust enough to handle this changing behaviour.

Distribution of target variable with these source features tells us that users prefers to listen to their favourite songs from local playlists, local library, my library and artist.



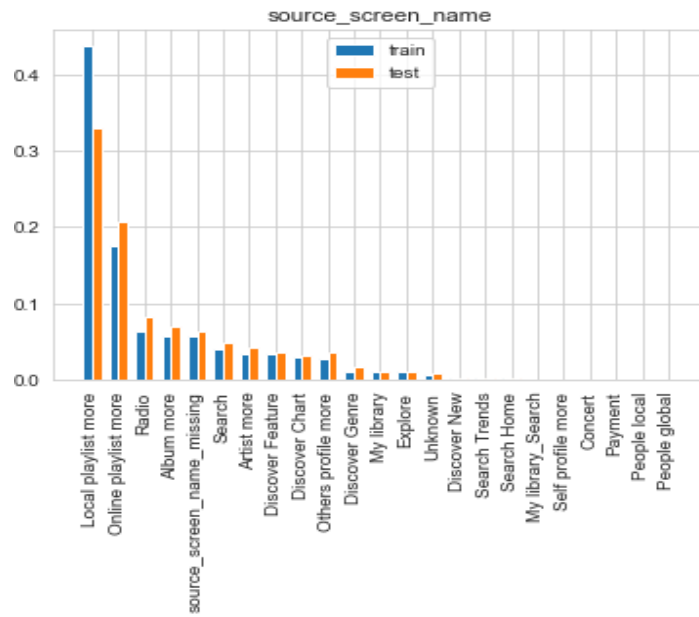


Figure 6: Distribution of song sources between test and train data

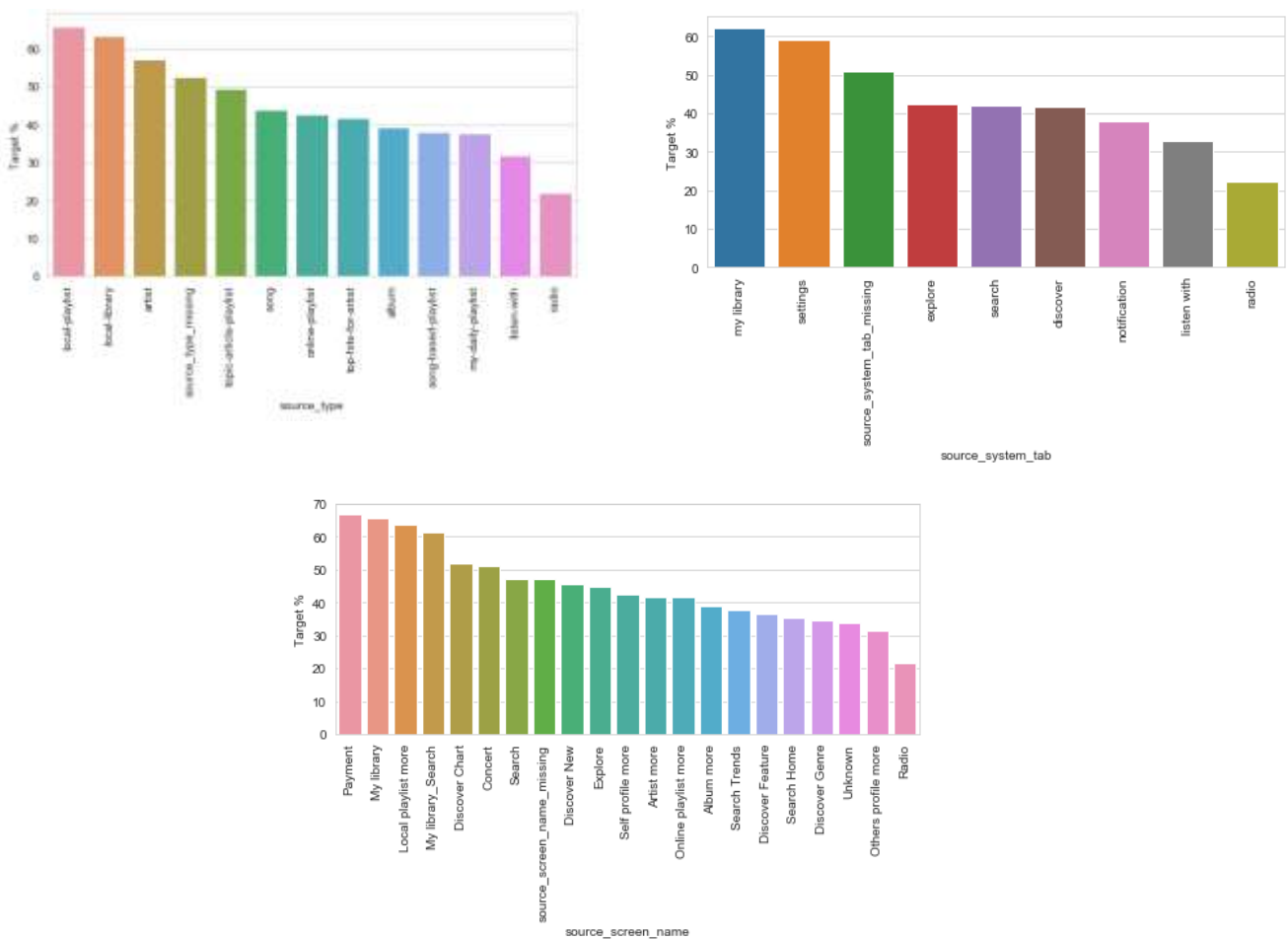


Figure 7: Influence of Song Source on repeat listening of songs

4.2. Gender

There are three categories in gender i.e. male, female and gender_missing. There is fairly equal distribution of male and female with male being 31% and female being 29% in the training set. There was no correlation found between gender and target variable, though users with missing gender prefer repeat listening less than male or female ([Figure 8](#)).

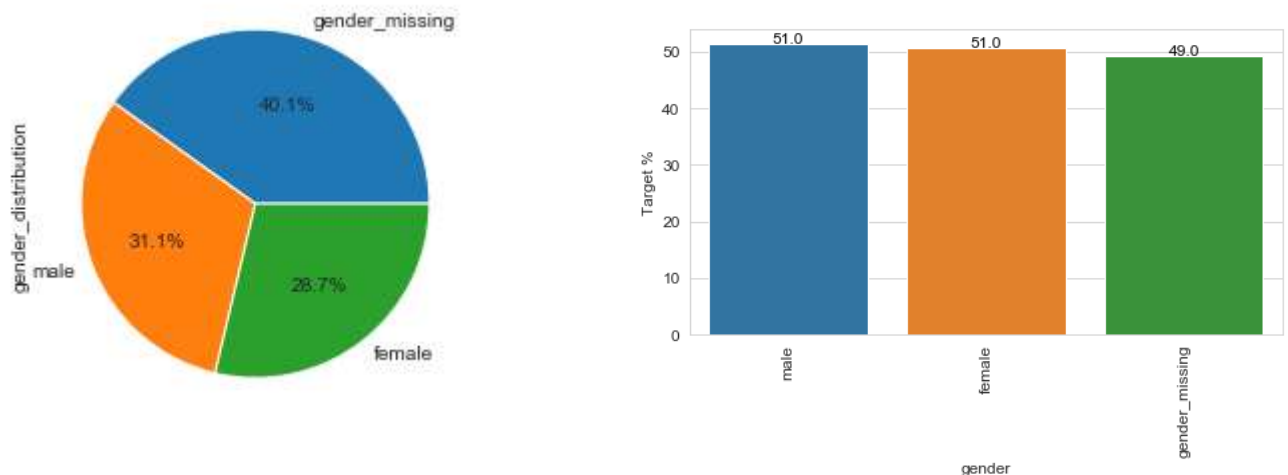


Figure 8: Gender Distribution and its influence on the repeat listening

4.3. Song Language and City

In language, 52 and 3 are the most listened to (~80%). On Comparing with cities and countries, we can infer that these two languages dominate uniformly across all cities and 53 might be English while 3 might be Taiwanese or Chinese. This is as expected because KKBox is a Taiwanese company targeting East and Southeast Asia. More details can be found in [IPython Notebook](#). It seems like languages "-1", "17" and "45" lead to a lower chance of repeated listens. Language "38" does that too, but there are only 100 songs in it, which in total haven't received a lot of plays too.

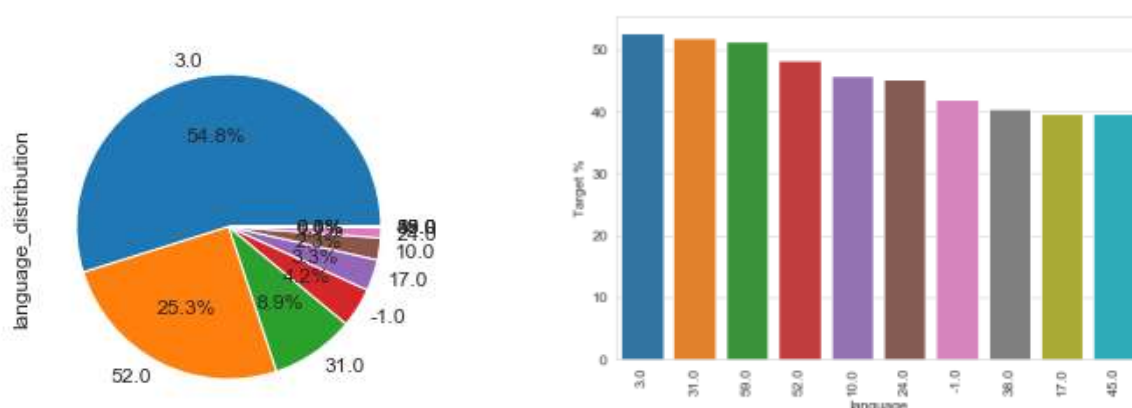


Figure 9: Language Distribution and its influence on the repeat listening

On the other side, city doesn't seem to have much influence on repeated listening. Cities like 20, 16, and 19 do affect the target but their frequency is low (Figure 10).

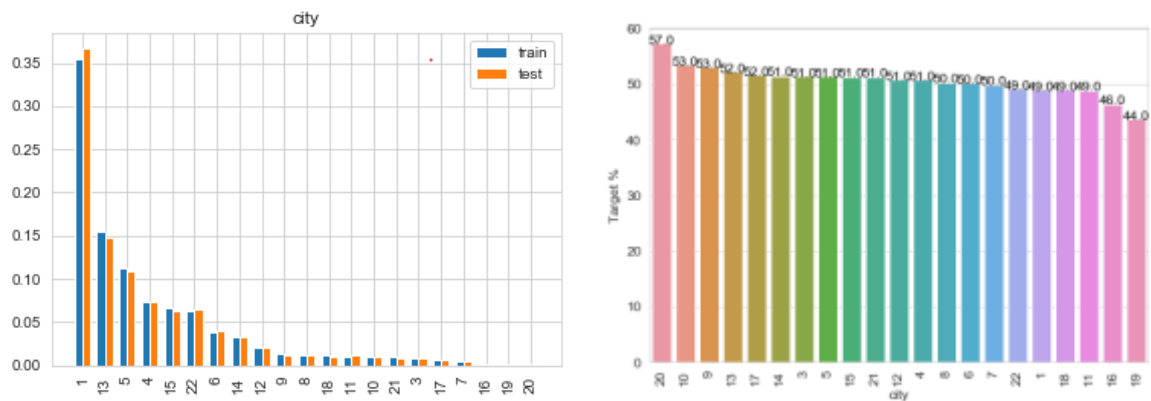


Figure 10: Language Distribution and its influence on the repeat listening

4.4. Count Features

Features analysed here were artist count, lyricist count, composer count and genre count for a song. On exploring the relationship between these variables and repeated listening, it was observed that genre count and artist count will have more influence than others. There were some exemptions for higher count values in composer and lyricist but these can be ignored due to their low frequency.

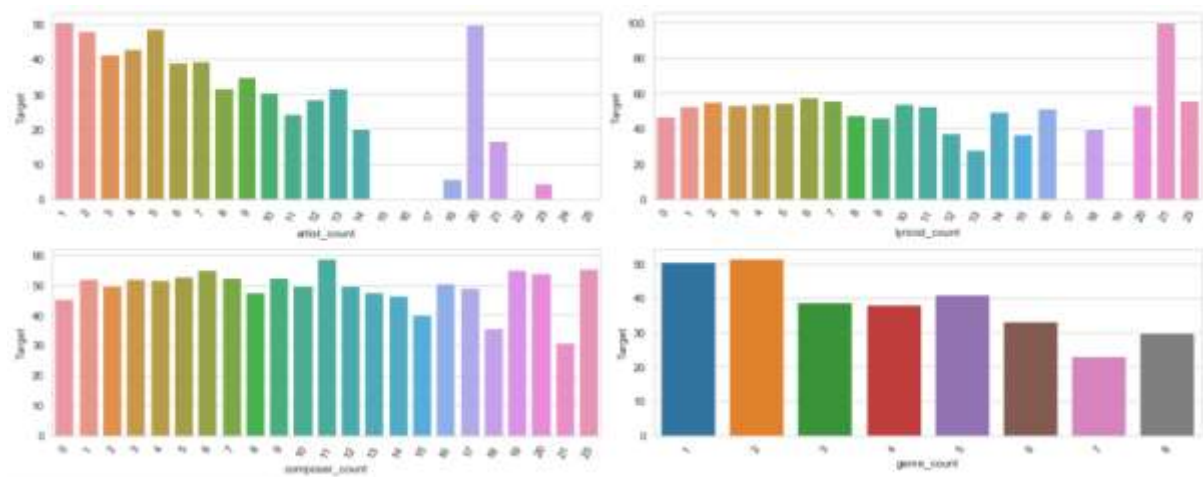


Figure 11: Count Feature and their relationship with repeated listening

We also created additional feature whether artist, composer and lyricist are same or not. It was observed that not many songs (<0.5%) had artist, composer or lyricist same. But there is some influence on the repeated listening.

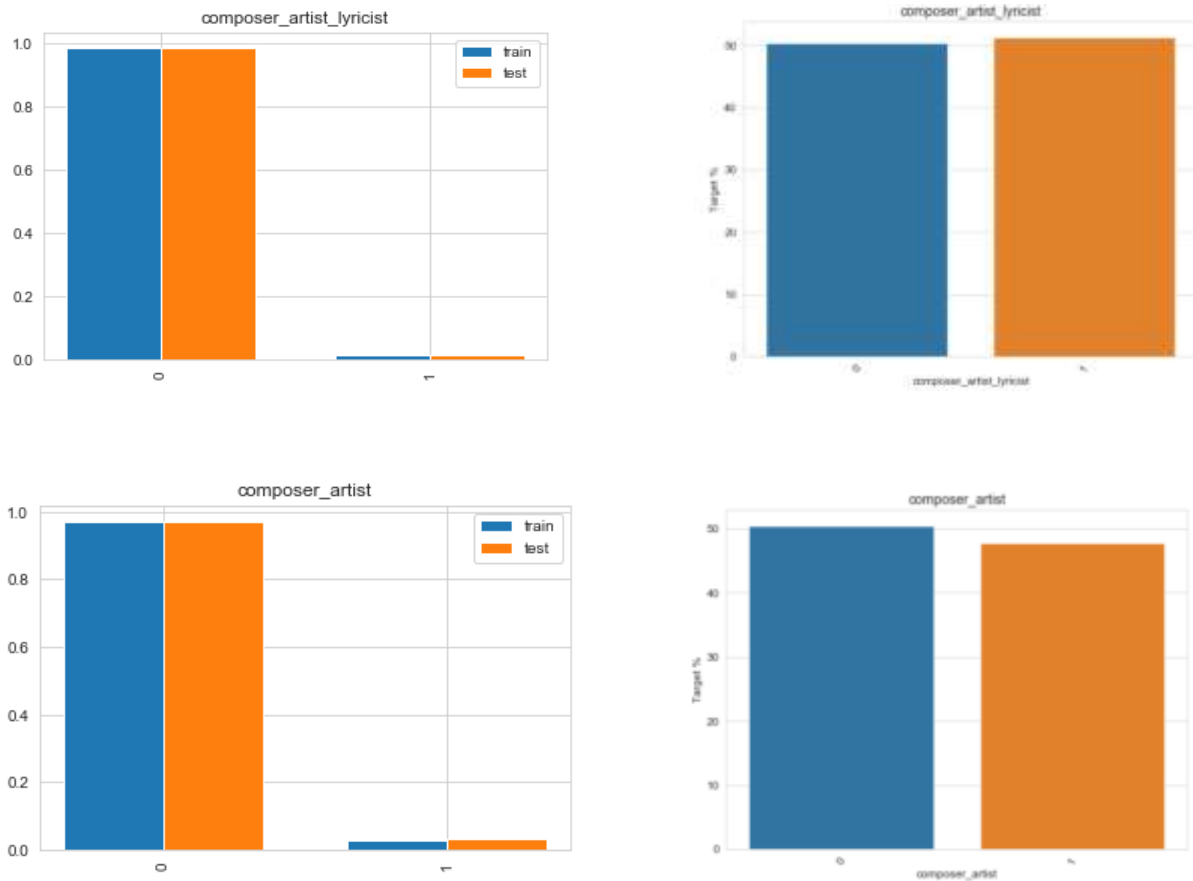


Figure 12: Composer, artist and lyricist similar feature distribution and its influence on repeated listening

4.5. Date and Time Features

We extracted day, month and year from registration and expiration time. The aim was to capture some time series and evolving customer behaviour through these features. Fig shows that the company's customer base expanded very fast from 2009 onwards. Also, almost 99% of the customers' subscription is ending in either 2017 or 2018.

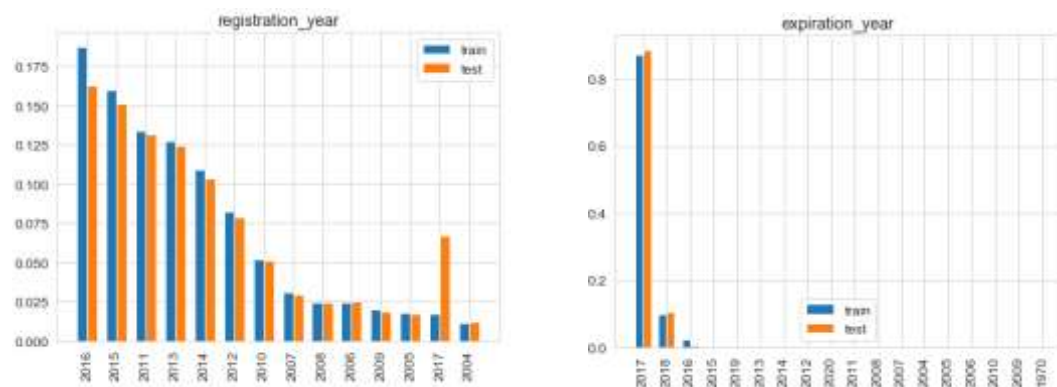


Figure 13: Distribution of date features between test and train data

With respect to repeat listening behaviour, we observe that no concrete trend was observed for features except for expiration year. Expiration year also might not have big influence because the frequency of years other than 2017 and 18 are very less as discussed above.

There were few months, years or days which has lower chance of repeated listening but nothing concrete can be said. We decided to keep the features and let model do its work on extracting information from these columns. A combination of registration and expiration can give more information. So we created a feature of membership duration.

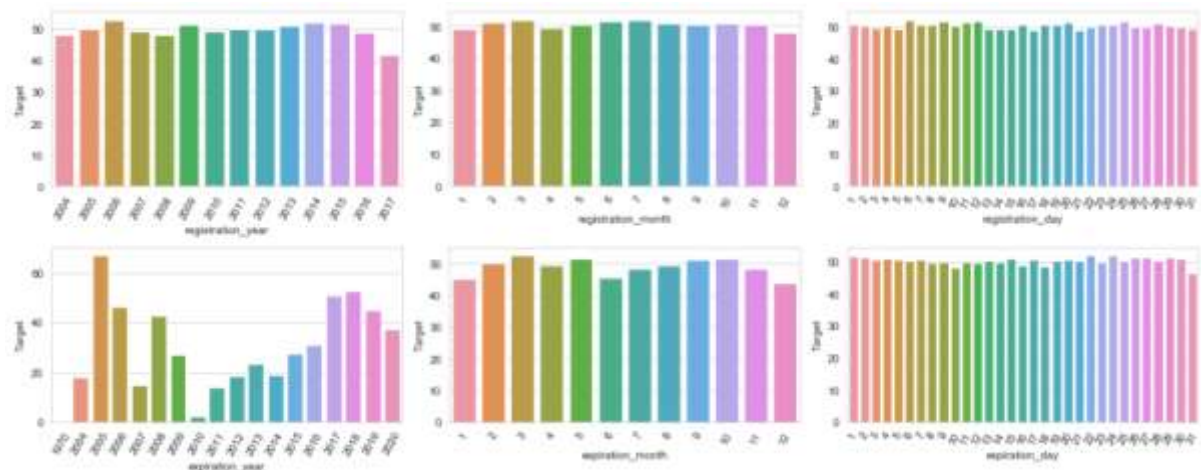


Figure 14: Date Features and their relationship with repeated listening

4.6. Song Type

Song type consist of 3 categories – long, short, medium. These are classified based on song length. As it will be expected, long and short duration songs will be less compared to medium length songs. Length of the song, has influence on the repeated listening. Short and long songs have less chance of getting repeated compared to a medium duration song (Figure 15).

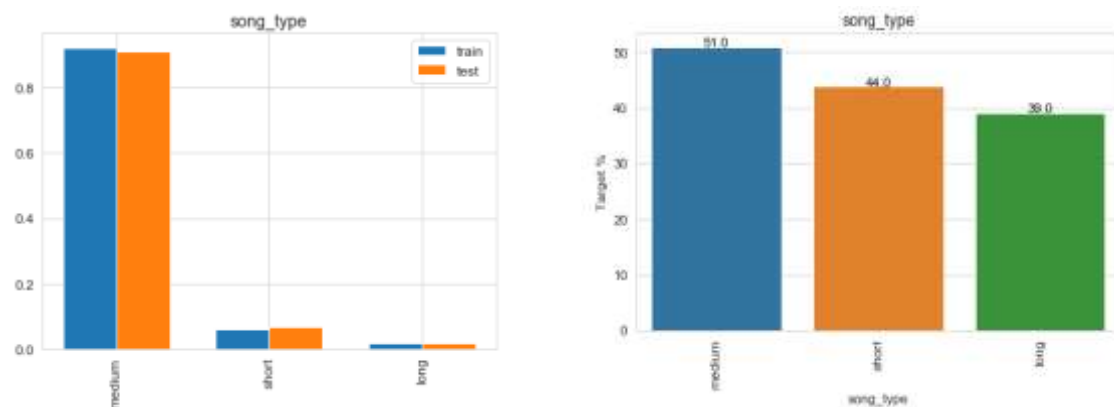


Figure 15: Song Type and its influence on the repeat listening

4.7. Registered Via

This features give information on how did the customer registered for the membership. 9 and 7 are the most frequently used ways to get registered. Fig shows that this feature will not have significant influence over the repeated listening behaviour.

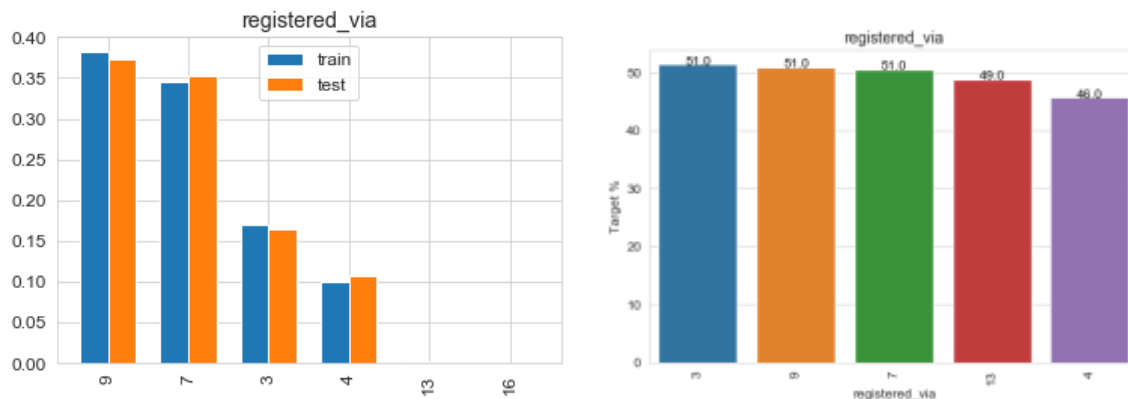


Figure 16: Registered Via and its influence on the repeat listening

4.8. Numerical Features

Following numerical features were explored:

- song_length
- age
- count_song_played
- count_artist_played
- duration
- age_of_song
- isrc_year
- song_freq

Above features are log normally distributed except for duration which shows a bimodal distribution (Figure 17). Song length shows variation in distribution for target variables, supporting the song type hypothesis. Age doesn't show significant influence on the repeated listening behaviour. Variables like count_song_played, count_artist_played, age of song, isrc_year and song_freq shows different distribution between target variables.

Figure 18 is the heat map of correlation between all the numerical variables and target. This quantifies and support some of the findings from above exploratory analysis. There are some collinearity observed for extracted features. The prominent was registration year and duration that shows 0.99 correlation, implying that we can drop one of the feature. So we decided to drop registration year.

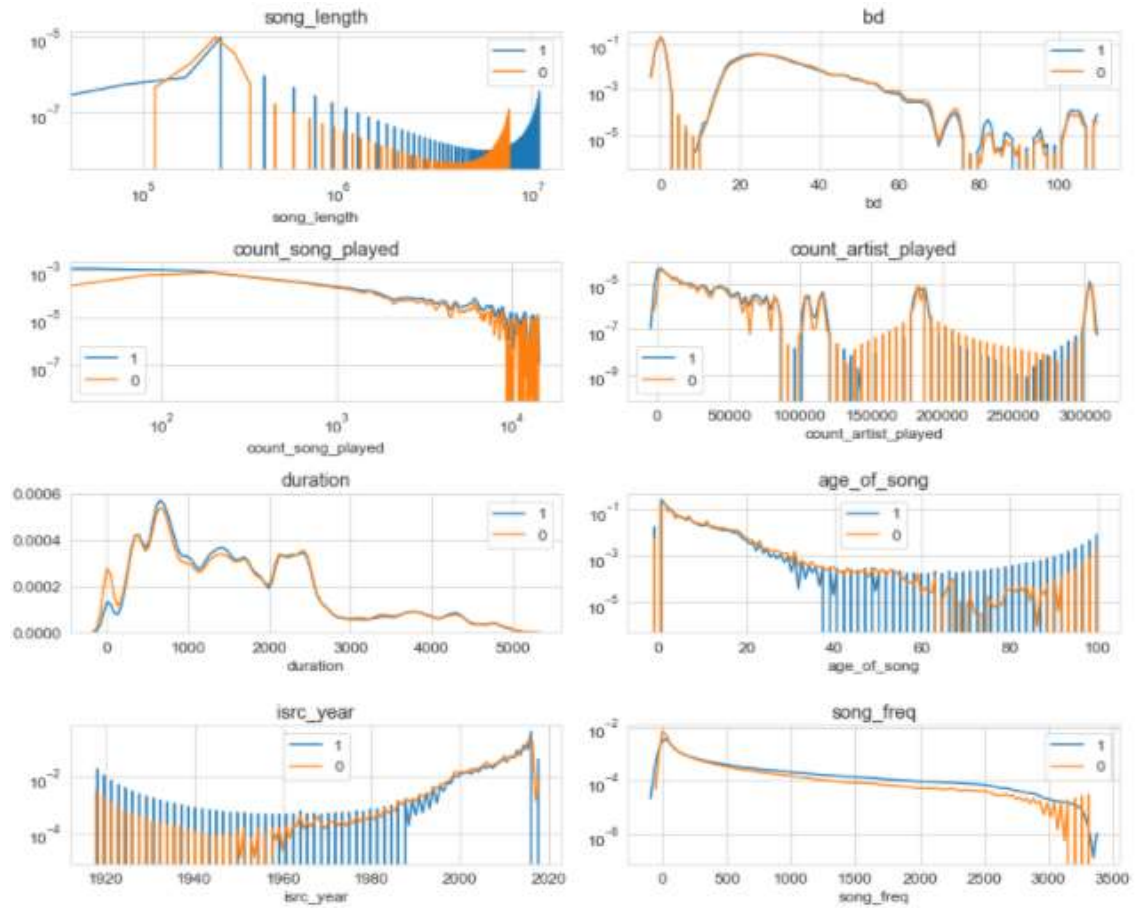


Figure 17: Distribution numerical features between repeated and no-repeated listening

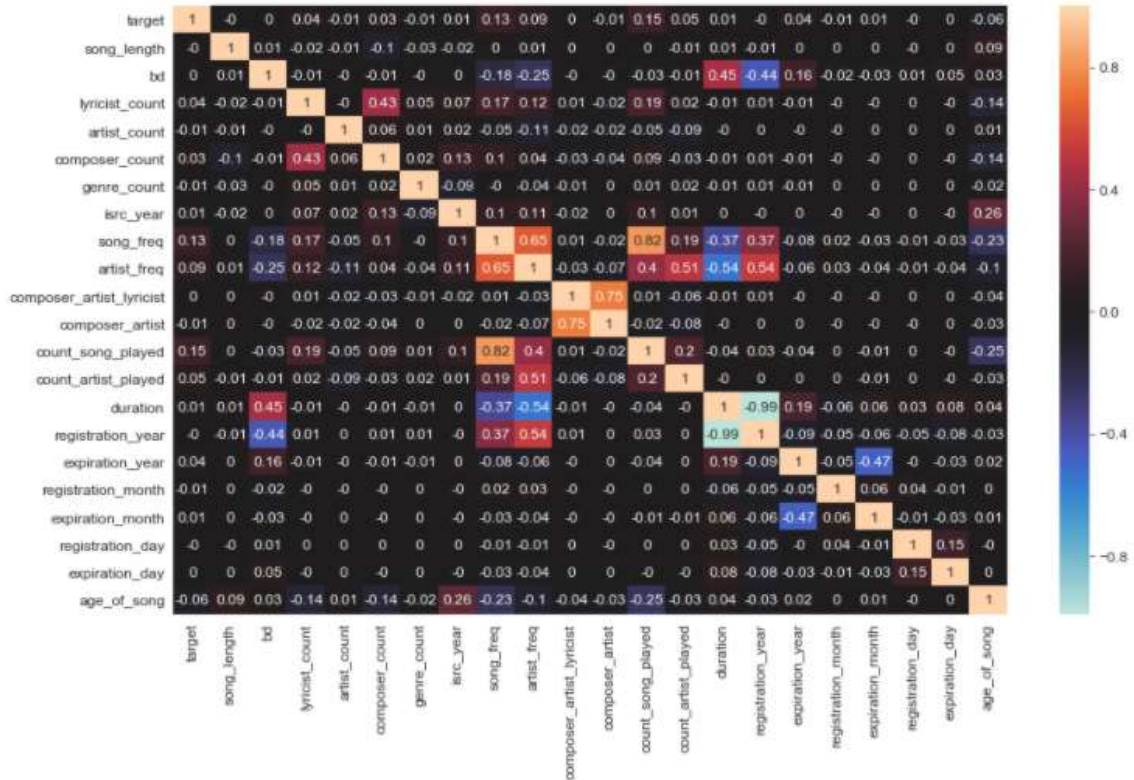


Figure 18: Heat Map of Correlation between numerical features

4.9. Artist and Song Frequency

Artist and song frequency were created to capture time element and the number of times the song or artist was played. The feature measures the cumulative number of times a song was played as the time passed by. Intuition says, the more a song is played higher is the chance that this will be listened frequently by the users (Figure 19 and Figure 20). Also as the song gets old, the frequency might decrease. This is supported by the negative correlation of -0.23 between age of song and song frequency.

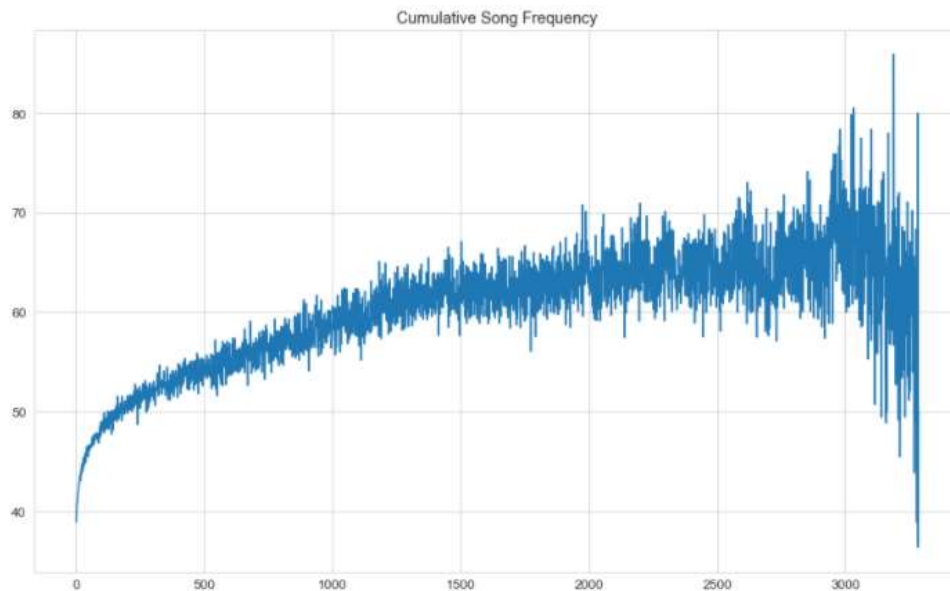


Figure 19: Song Frequency and Target

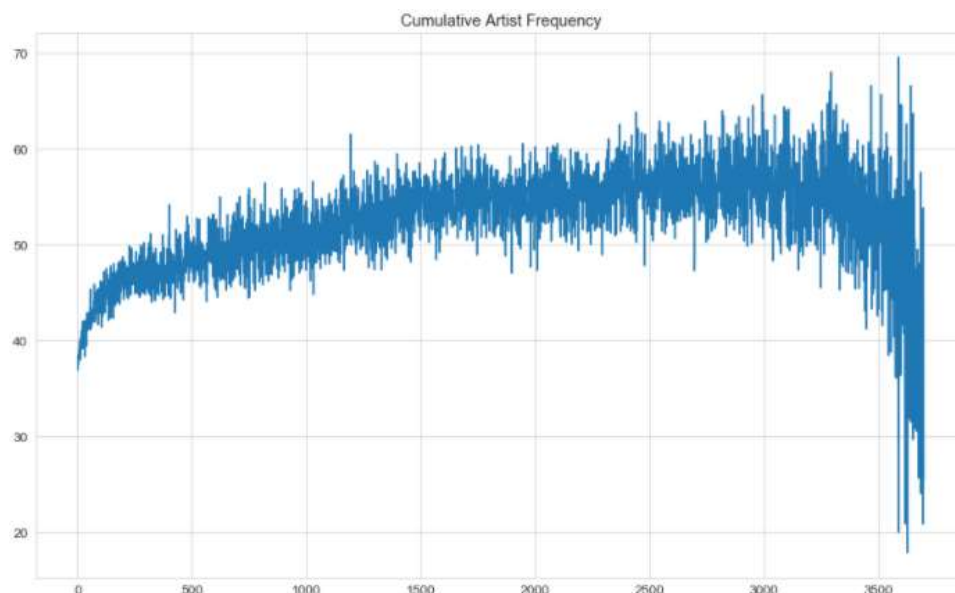


Figure 20: Artist Frequency and Target

4.10. Summary and Key Findings

- Exploring testing data tells us that we will have to deal with cold start problem during modelling to make a robust model.
- This is a balanced data set
- Distribution variations between test and train says there has been slight change in user behaviour over time
- Features that will significantly influence repeated listening are:
 - source_type, source_system_tab, source_screen_name
 - language
 - composer_count, artist_count and genre_count
 - expiration year
 - song_type
 - song_length
 - song and artist frequency
 - count_song_played, count_artist_played
 - membership duration
 - age of song
 - isrc_year
- Features that will not significantly influence repeated listening are:
 - Age
 - Lyricist count
 - Composer_artist_lyricist
 - Composer_artist
 - registration year, registration_day, registration_month
 - expiration_day, expiration_month
 - city
 - gender
- Except for registration year, all features will be kept. I will leave upto the algorithm to select the best features
- Categorical features like lyricist, artist, composer, were not touched in this analysis. They will be used to later in the project.