KKBox's Music Recommendation Analysis

Ana Villalba
November, 2018

1. Introduction

Music is an important part of the daily life for lots of the people. Currently music is a billion-dollar global industry due to the high demand of the public of music consumption.

In 2017, 38 percent of the music industry revenue worldwide was attributed to streaming sector. Big players as Google with Youtube Music, Apple with Apple Music, Spotify or KKBox are trying to get their piece of the cake. In this highly competitive environment, being able to satisfy users' needs by implementing and optimizing music recommendation systems will be crucial.

By improving recommendation systems, business metrics could be as well improved. For example, offering songs adjusted to the user preferences can improve retention which is highly related to monetization. Many companies in the music industry have subscription business models where increasing retention is key to improve the revenue.

2. Client and problem definition

KKBOX was founded in 2004 and it is leading the music streaming industry in Asia. Its wide Asian-Pop music library contains more than 30 million tracks. The service allows you to listen to your music playlists from all your devices and uses a recommendation system combining matrix factorization and word embedding methods.

Nowadays, user music preference could vary from classical music to the latest hit. Recommendation systems need to predict whether a person will enjoy a new artist or a new song. This is especially challenging when the listener has recently joined the service, since there is not enough historical data. Predicting user behaviour is always a difficult task. This project will focus on answering the following question:

How likely is that a user listen to a song again in less than a month after the first time he/she has listened to it?

Answering the previous question could help improving retention, and therefore reducing churn. The service could, for example, prepare automatic playlist including songs that the user is likely to play again or suggest songs similar to those ones.

This problem was proposed by the International Conference on Web Search and Data Mining (WSDM 2018). One of the challenges about this problem is the fact that there are several datasets provided, so having to combine those pieces of information makes the analysis harder. The available dataset it is very recent, from 2017, so there are not many references about its analysis.

3. Data

3.1 Data description

The dataset contains a sample of over 7.4 million listening events performed by more than 30,000 users. Each event corresponds to the first one performed for a particular user-song pair in the selected time period. It also contains user and song metadata. The dataset is organised in different files:

- **train.csv** (7.4m rows, 30.8k users) lists the listening events which include usability information such as the screen or tab the song was listened from and the output (whether the song was listened again or not by the user).
- **members.csv** (34.4k rows) contains registration and demographic details about the users.
- **songs.csv** and **song_extra_info.csv** (2.6m rows) include song information such as the name, artist or duration.

Data and full description of each field are available on Kaggle: https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data

We assume that the data is a sample representative of the population. If there was any selection bias in the data collection process, the results of this project won't be relevant to the service.

3.2. Data wrangling and cleaning steps

Missing values

In the train file, missing values that are recorded as 'Unknown' were replaced by NaN. Rows containing missing source_system_tab and source_type values were dropped since they represent less than 5% of the data. Missing language values in songs.cvs file were replaced by -1.

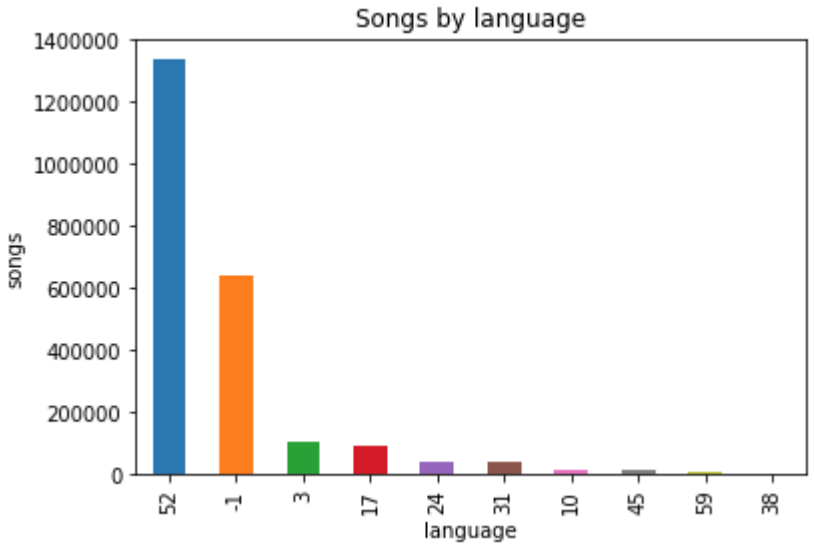


Figure 3.1

Normalizing text

Text in fields such as source_screen_name, artist_name, composer, lyricist or song name was normalized by either transforming everything into lower letters in the case of the source_screen_name or capitalizing the first letter of each word and removing unnecessary spaces in the rest.

Assigning proper data types

Many fields were treated as object when in reality they were categories (source_system_tab, source_screen_name, source_type, song language, city, gender and registered_via), datetime (registration_at and expiration_at) or lists of objects (genres_ids, composer and lyricist).

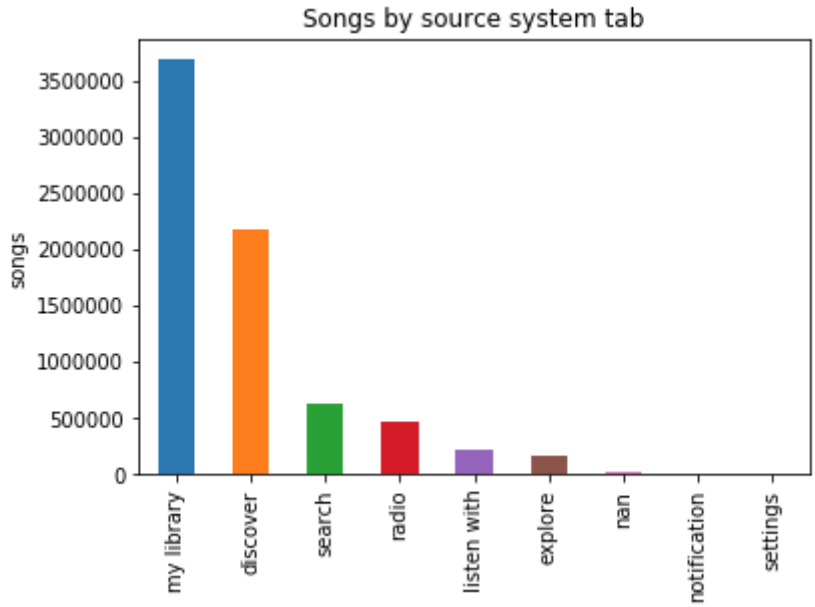


Figure 3.2

By transforming categorical variables into category type we also reduce memory usage and make the process more efficient.

Treating outliers

Rows where expiration date is before 2004 were removed since KKbox was launched in 2004. Age values less or equal than 0 and greater than 120 were converted into missing values (NaN).

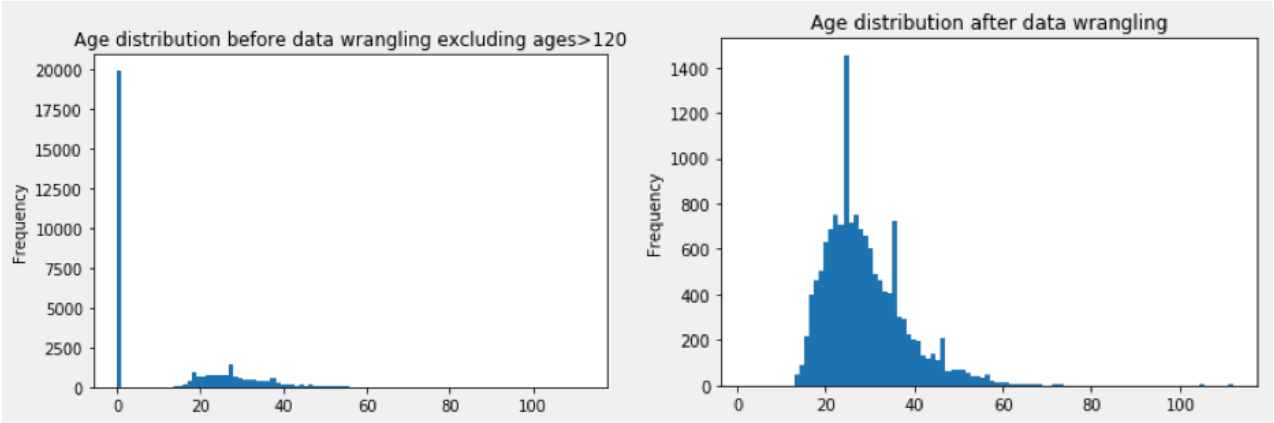


Figure 3.3

Other transformations

Song length was converted to minutes instead of milliseconds and an extra column genre_count was created displaying the number of distinct genres a song belongs to.

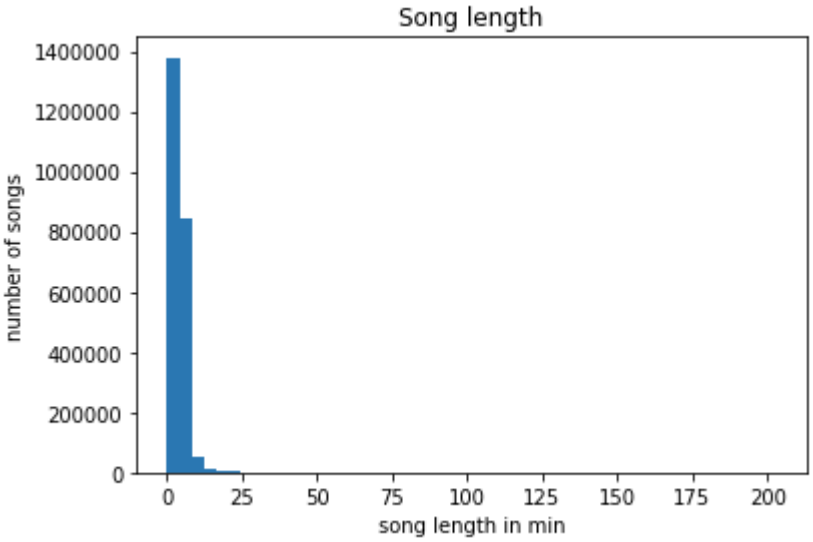


Figure 3.4

3.3. Additional data sources

In order to improve the quality of the data that we already have, we could try to find missing values related to the song information. In particular, there are many missing values in the composer, lyricist and language fields. We could try to find a website where the language of the song is accessible by looking at the combination of name and isrc code, since in most of the cases is available.

4. Initial findings

Our initial analysis was focused on trying to identify factors that affect the probability of a song to be played again within a month after the user's very first observable listening event. Many times we will refer to the 'replay probability' for convenience. We can divide the factors that we analysed in three categories: usability, user properties and song properties. Inside each category we tried to answer some questions.

4.1. Usability

- **Do users who listen to more songs have higher chances to replay songs in general? Basically, is there any pattern for users who are more engaged?**

You could expect that highly engaged users are more likely to replay a song. In order to see if there is any correlation in this case, we have first considered that a user is more engaged if he has listened to more unique songs (in the dataset it is not available how many times each song was listened). In the following histogram, we see that most of the users have played few different songs.

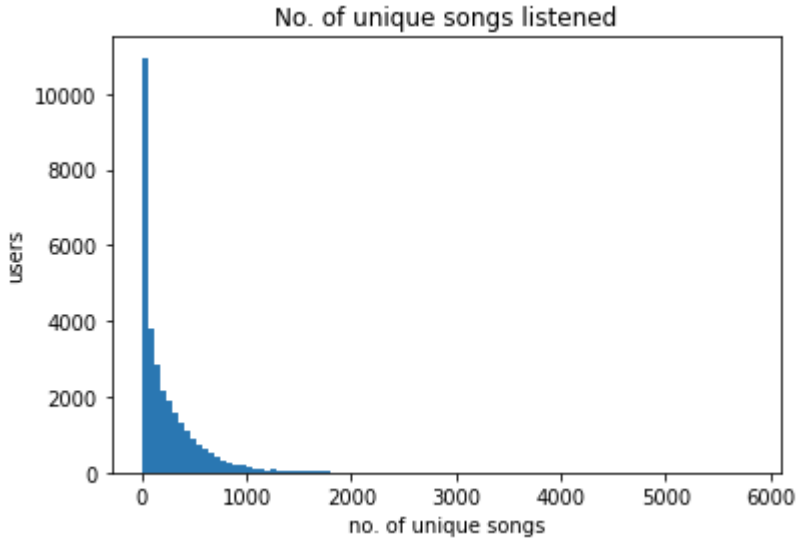


Figure 4.1

We could not find visual evidence of any correlation between the engagement and the replay probability (Figure 4.2).

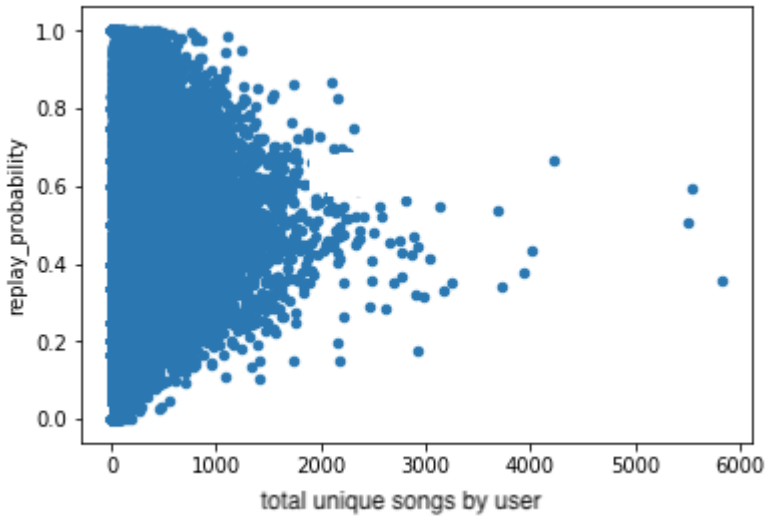


Figure 4.2

- **Are songs that are listened from 'My library' tab replayed more than songs that are listened from other tabs? Are there more chances that a user plays again a song if it is accessed from a specific screen or section in the application?**

The more interesting findings were related to the screen/tab from which the song is accessed from. Songs listened from 'My library' tab, which happens to be the most popular one, and source types 'local-library' and 'local-playlist' seems to have the highest chances to be replayed. On the other hand, songs listened from radio tab have the lowest chances to be replayed.

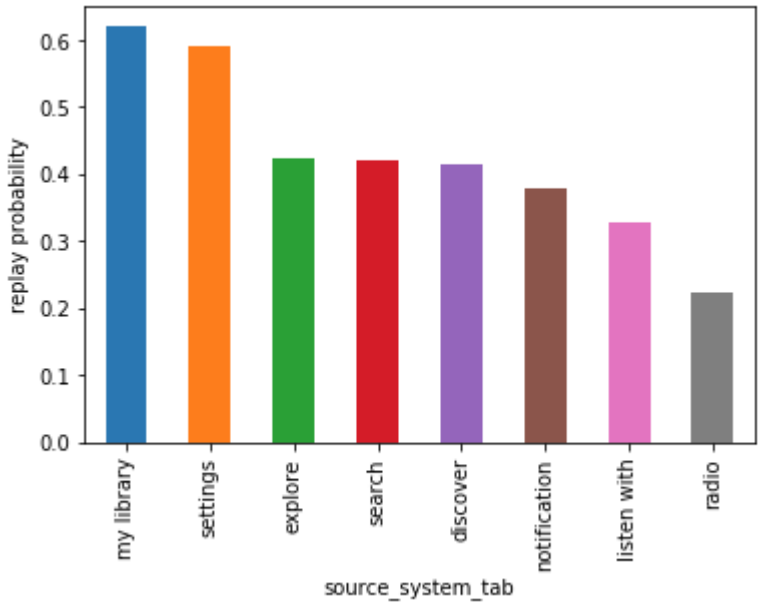


Figure 4.3

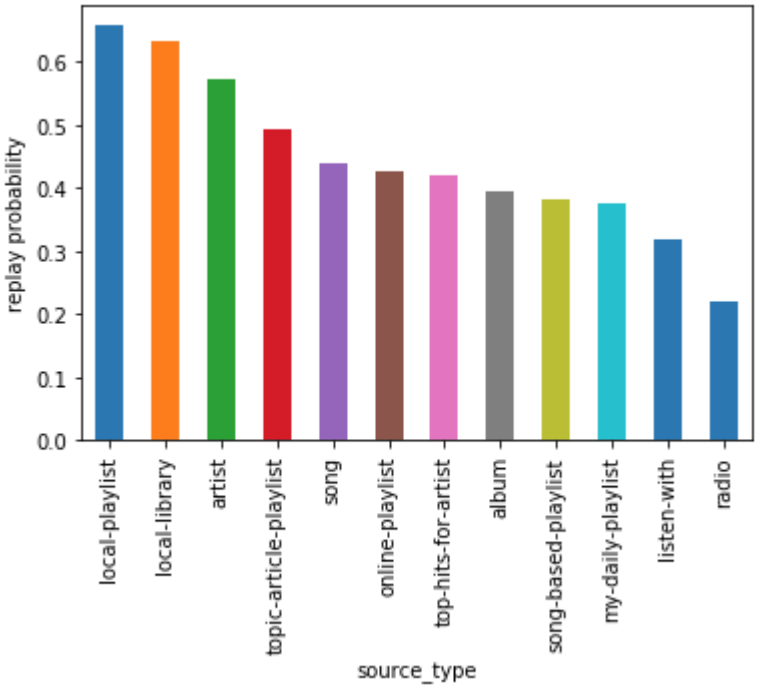


Figure 4.4

We can observe as well those insights when we plot the combined probability from source_system_tab and source_type features. Note that some combinations with high probability have very few observations and therefore are not relevant, as for example discover/artist, explore/song-based-playlist or radio/artist.

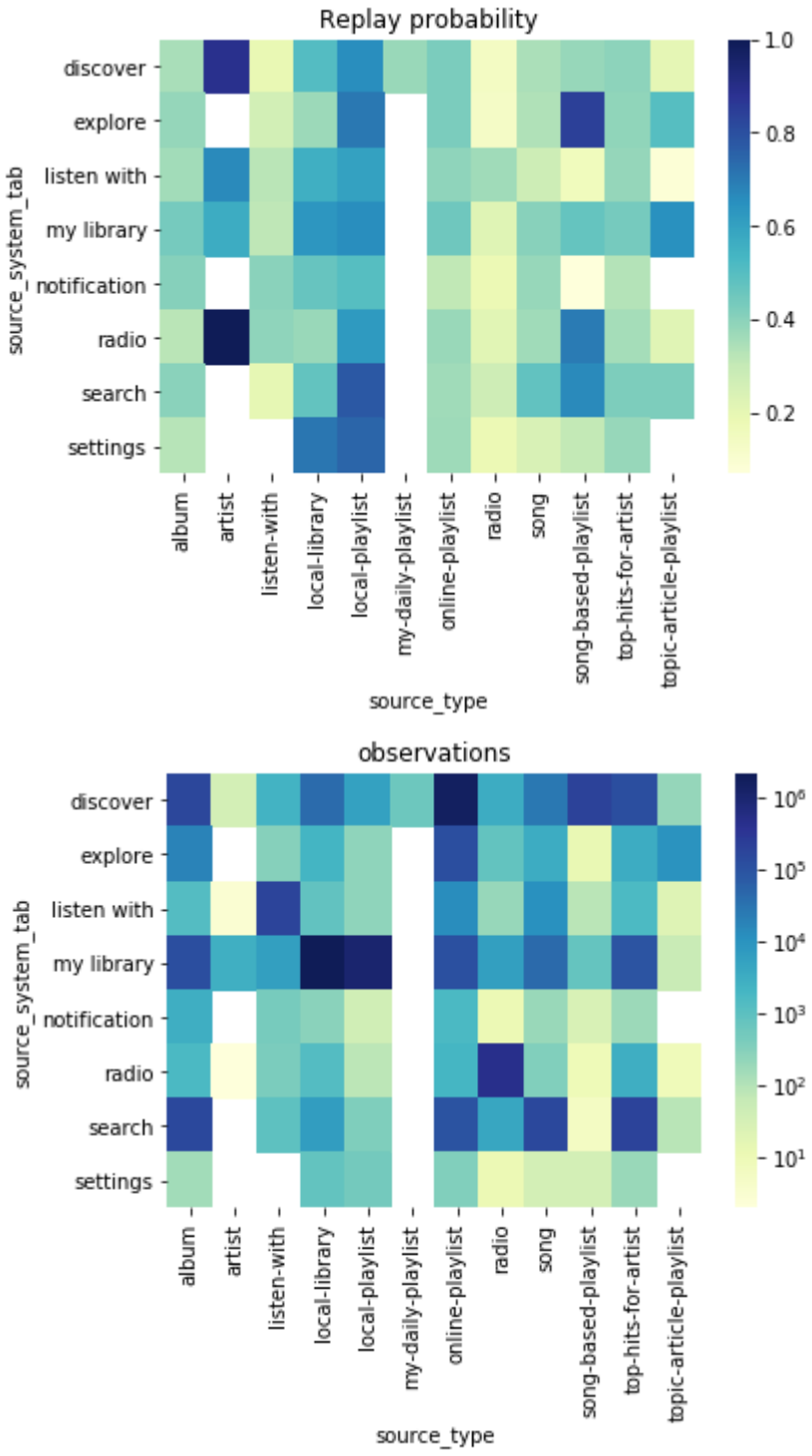


Figure 4.5

After running a hypothesis test, we demonstrated that ‘My library’ tab has higher probability than the rest of the tabs. We obtained a p-value of 1. Since the p-value is really high, we rejected the null hypothesis that assumes the probabilities for ‘My library’ tab and the rest are the same. The minimum difference observed in the test between 'My library' tab probability and the others is around 23.2%.

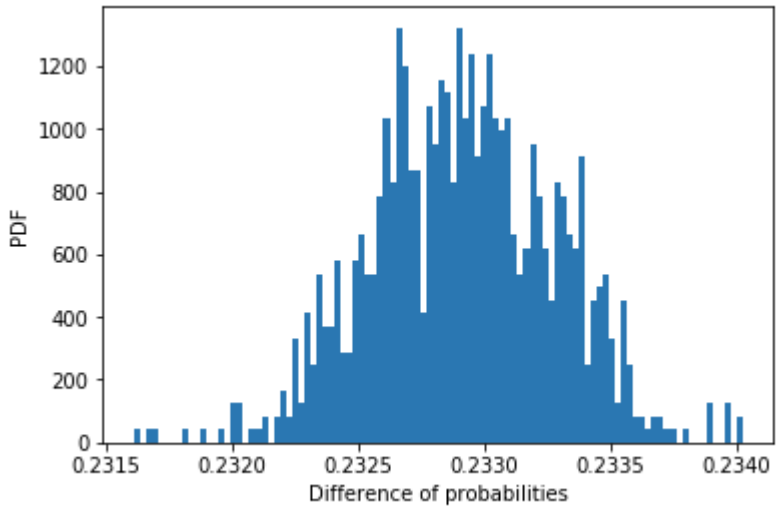


Figure 4.6

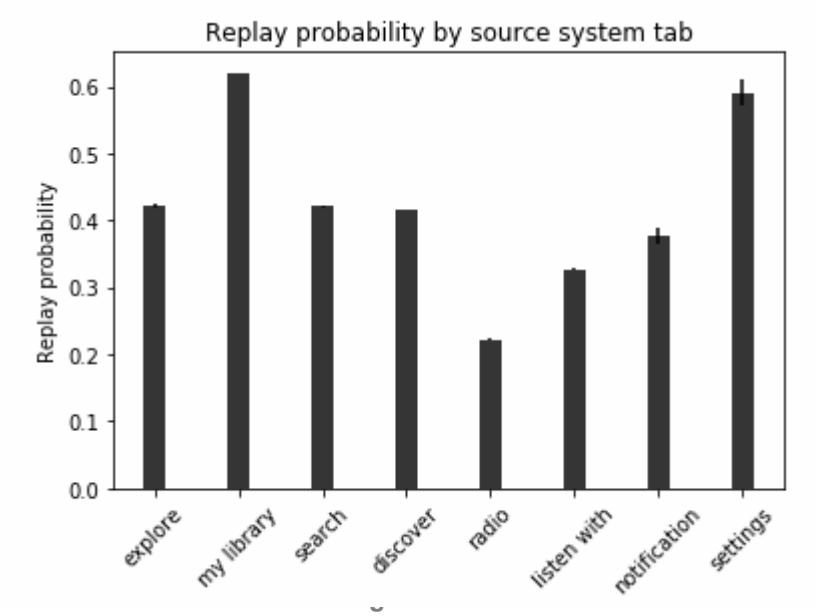
We conclude that the probability of replaying a song when it is listened from 'My library' is higher than when it is listened from other tabs.

We wanted to analyze all tabs individually to see if we could spot any other interesting differences.

For that purpose, we computed the probability of a song to be listened again depending on the tab the song was listened from. We calculated 95% confidence intervals of the probabilities. Here are the results:

- Explore tab: [0.420, 0.425]
- My library tab: [0.619, 0.620]
- Search tab: [0.420, 0.423]
- Discover tab: [0.415, 0.416]
- Radio tab: [0.221, 0.224]
- Listen with tab: [0.325, 0.329]
- Notification tab: [0.366, 0.390]
- Setting tab: [0.570, 0.611]

In the following graph, the mean probabilities together with the confidence intervals are presented:



From the previous results, we can conclude that there are clear differences in probability depending where the song was launched from. The best tabs are ‘My library’ and ‘Settings’ and the worse one is ‘Radio’.

It makes sense that users visit frequently playlists and songs that are saved in their library. It is also logical that songs from the radio tab are not listened that frequent, because it is not the user who chooses what song is going to be played.

4.2. User properties

- **Do users who registered before have higher probability to listen to songs again?**

It could be expected that users who have been using the service more would have higher chances to replay songs. Nonetheless, we did not find any visual correlation between the registration date of the users and their probability to replay songs.

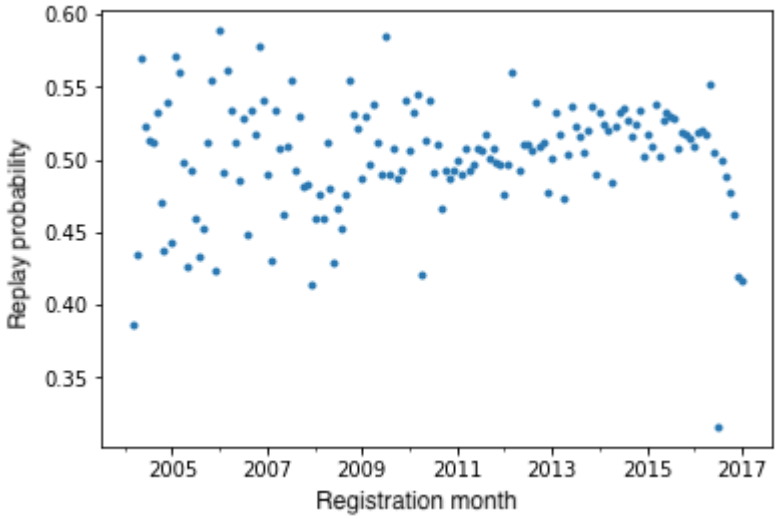


Figure 4.8

- **Is there a significant difference between males and females in the probability of replaying songs?**

We were as well interested to check if there were differences between males and females. Below it is displayed the empirical cumulative distribution for men and women.

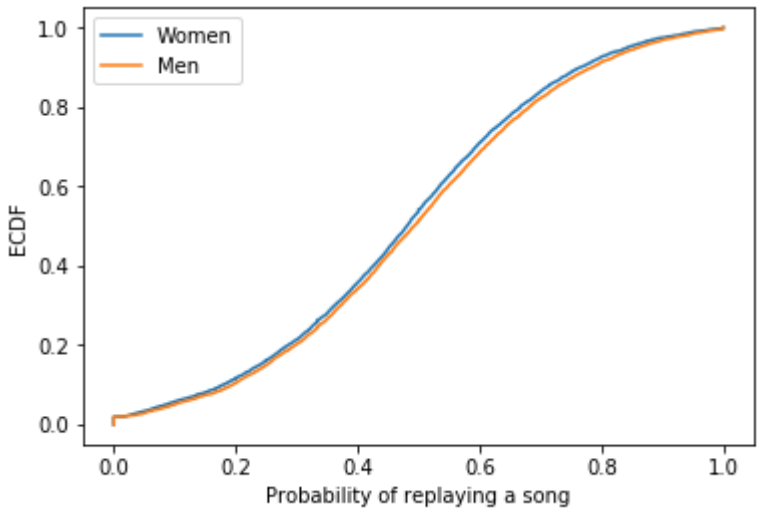


Figure 4.9

We did a hypothesis test where the null hypothesis assumes the mean replay probability in men and women is the same.

The mean observed probability in men is 48.94% and in women is 47.66%.

We obtained a p-value of 0.9998. At a confidence level of 95%, provided that this is a two tailed test, the null hypothesis would be rejected. The p-value is higher than 97.5%. We conclude that the difference is statistically significant and men have a higher probability of replaying songs.

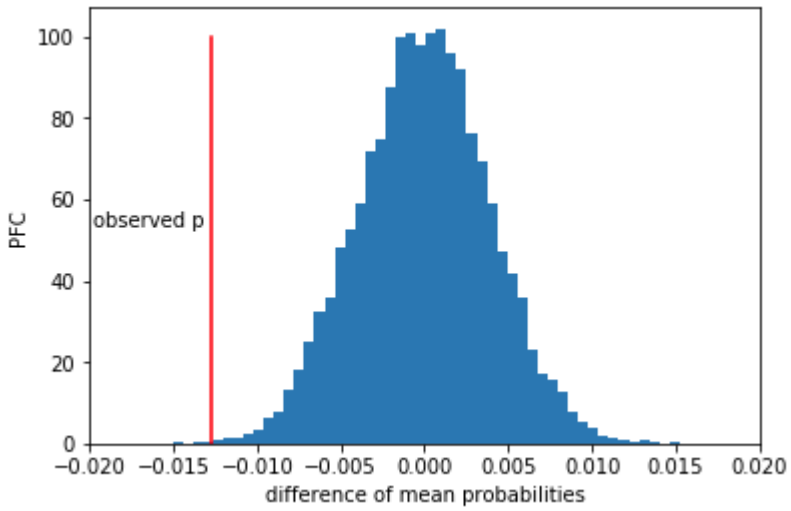


Figure 4.10

- **Do younger users replay songs more often than older users?**

Finally, we wanted to analyze if younger users were replaying songs more often than older users. From the plot below, there is no clear correlation between the age and the replay probability.

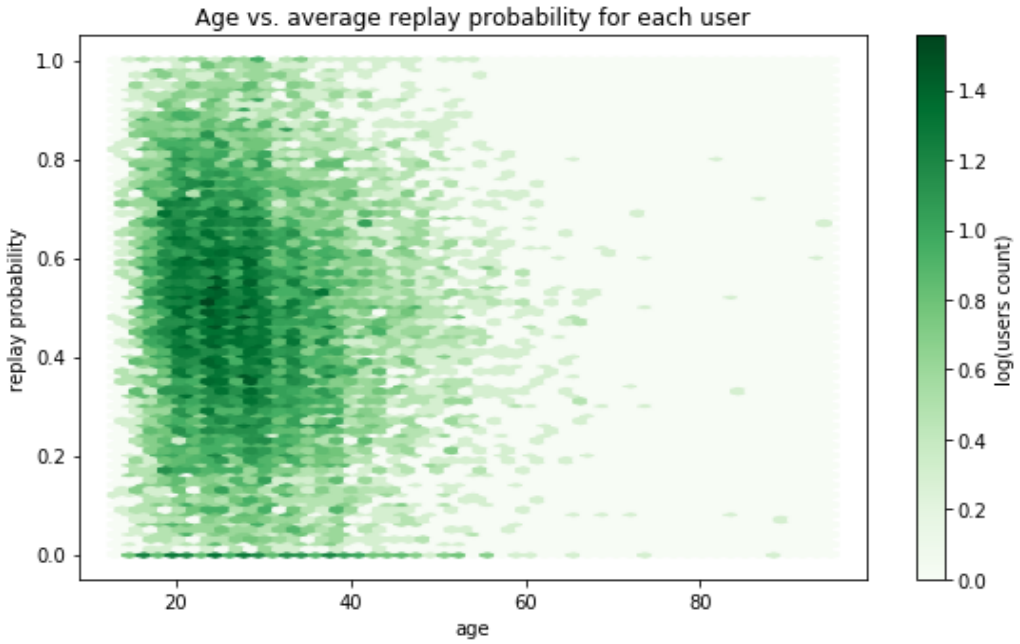


Figure 4.11

We run another hypothesis test to check it. In this case, the null hypothesis states that there is no correlation between the user's age and the replay probability.

The observed Pearson correlation between the variables is -0.09.

We obtained a p-value of 1. Since the p_value is really high, we rejected the null hypothesis in favor of the alternative one. There is a weak negative correlation between the users' age and the mean probability of a user to replay songs. Overall the probability or replaying songs decreases as users get older.

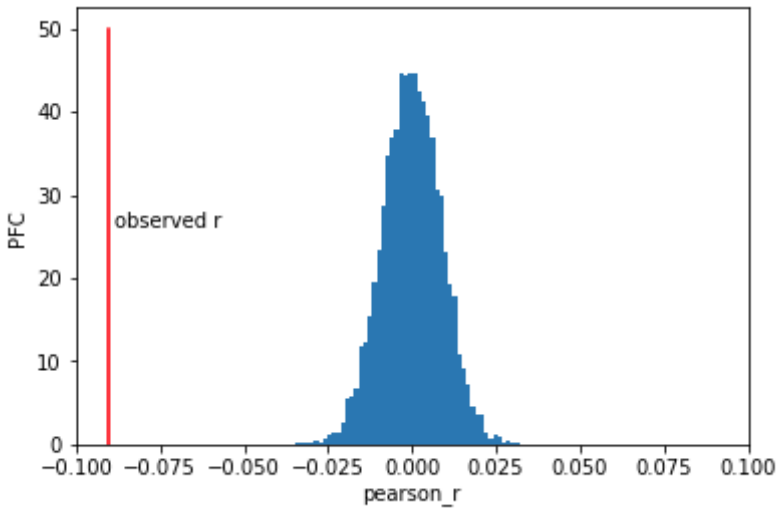


Figure 4.12

4.3. Song properties

We analysed as well some characteristics of the songs such as language, number of different genres that belong to or duration.

- **Do users like (listen again) more long songs or short songs?**

As shown in the plot below, we can't find any visual correlation between duration and replay probability.

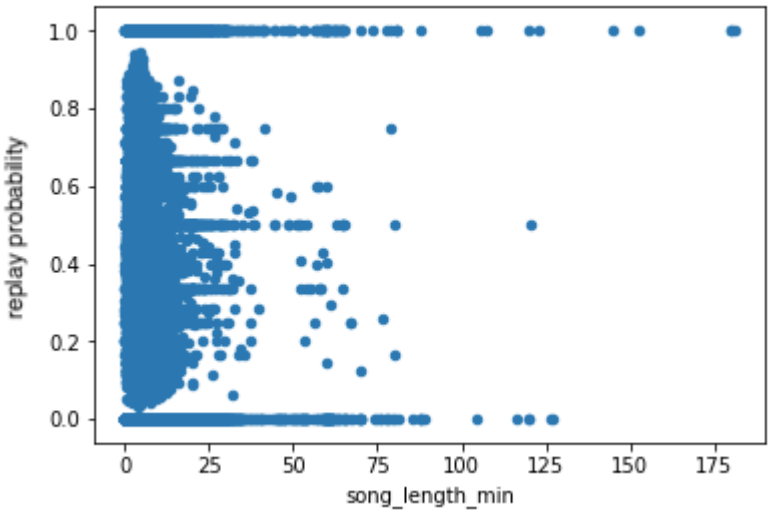


Figure 4.13

- **Do songs who belong to more than one genre have more possibilities to be listened again?**

In the plot below, it is represented the mean replayed probability depending on the number of genres a song belongs to.

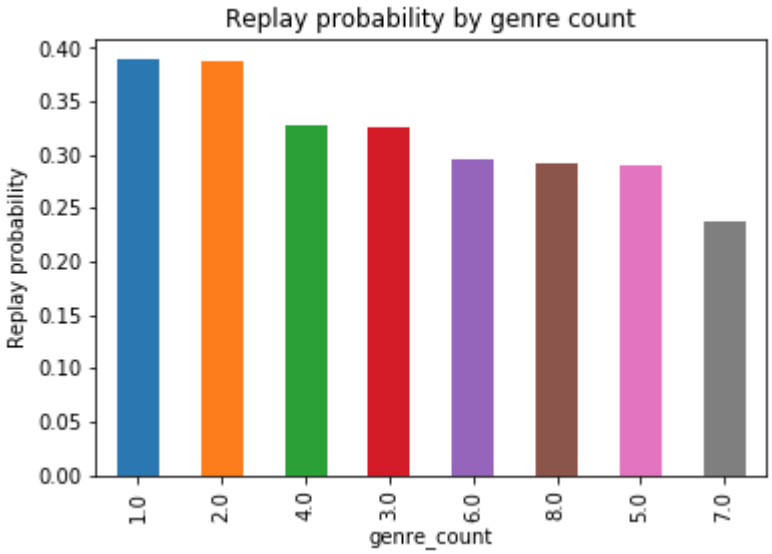


Figure 4.14

Although songs that are belonging to 1 or 2 genres seem to have higher probability, there is not a clear negative correlation between the number of genres and the probability.

In order to confirm if there is any correlation, we did a hypothesis test. Our null hypothesis assumed that there is no correlation between the number of genres a song belongs to and the replay probability. We obtained a p-value of 1 and rejected the null hypothesis in favor of the

alternative one. There is a negative correlation between the number of genres a song belongs to and the probability of that song to be replayed.

Nonetheless, the observed correlation (-0.021) is very small and therefore is not useful for practical purposes.

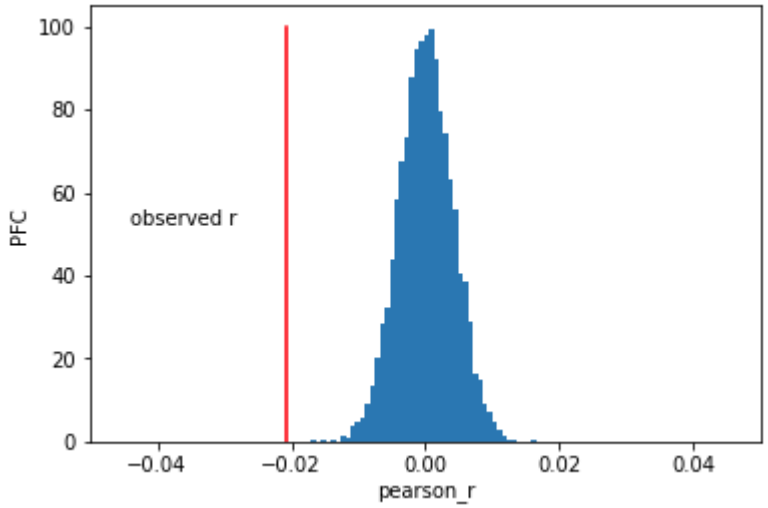


Figure 4.15

- **Are songs in one language listened more than others in other languages?**

We did not observe any interesting patterns regarding the song's language. Most of the songs belong to language with tag 52 or the language is unknown.

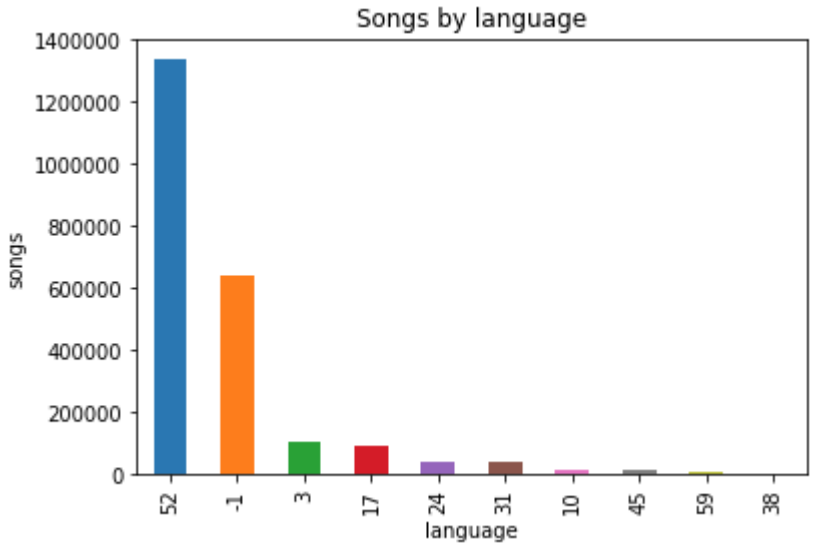


Figure 4.16

5. Prediction using machine learning

5.1. Feature engineering and subsampling of the training data.

Since the dataset is very large, in order to train and test the models a subset of the data has been selected considering the users who have listened between 20 and 30 unique songs. This number is large enough to increase the chance for users to appear both in the training and test sets but it is not too large to end up having a segment with few users that have listened many different songs. The subset contains 37.7k listening events and includes 1.5k users.

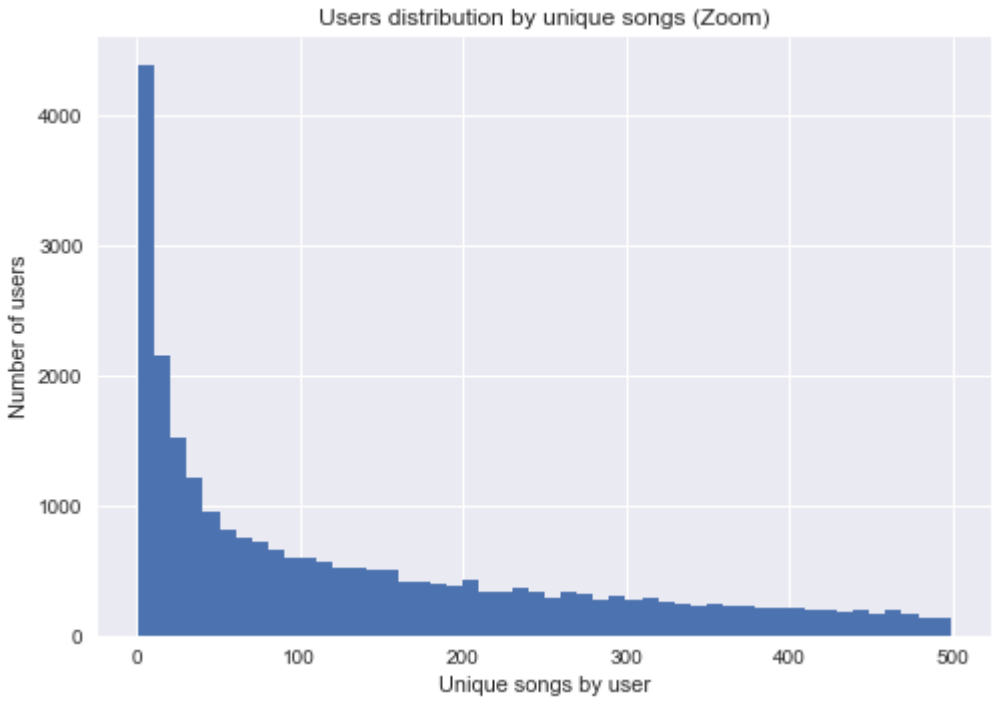


Figure 5.1

Some preprocessing steps were needed before fitting the models. Missing values in most cases were replaced by -1 so they will represent an extra category; member's age missing values were filled by the most common value. Binary variables were created with get_dummies pandas function for all categorical columns and with MultiLabelBinarizer sklearn function for columns containing lists such as gender_ids.

In order to capture the effect of time spent in the platform, the variable days_from_reg_to_exp (days from registration date to expiration date) was created. Also the expiration and registration years were extracted.

5.2. Feature and model selection

In this project we tried to predict if a user would listen again a song after the first listening event and within a time window. Since the dataset is very large, classifiers such as k-nearest neighbors have been discarded because of the slow performance.

Four models were chosen:

- **Logistic regression:** the output is a function of the different features which are weighted by coefficients. This method is suitable for large datasets.
- **Linear SVC:** this is a support vector machine model that uses a linear kernel. The linear kernel usually scales better for large number of samples than other kernels.
- **Decision tree:** determines which features have the most importance to the decision. It performs really good in large datasets and the results are easily interpretable.
- **Random forest:** creates several fully grown decision trees selecting a number of features and returns the best performing one. The trees can be trained in parallel reducing computational time. It usually outperforms decision tree classifier.

The analysis is repeated considering different number of features:

1. **Train** (36 features): only information included in the train.cvs file is considered. This table contains usability features such as the tab or screen the song is listened from.
2. **Train + members** (67 features): member features such as age, city or registration year are added to the usability features.
3. **Train + members + songs** (164 features): on top of the previous features, song properties such as duration or genres it belongs to are included.
4. **Dimensional reduction** (93 features): considering all features in point 3, dimensional reduction using PCA is performed.

5.3. Results

In order to evaluate the models, accuracy and AUC-ROC scores have been selected. These metrics are adequate since we are equally interested in classifying properly the positive and negative classes.

Random forest followed by decision tree are the best models to use. Song features did not improve the performance of the model, so it is better to just consider usability and member properties. In this case, the accuracy is 0.7618 and the ROC-AUC score is 0.8021. Adding member features to the usability ones increases the scores, as shown in Table 5.1.

Table 5.1

	Logistic Regression	Linear SVC	Decision Tree	Random Forest
Train	0.6644 0.6448	0.6645 0.6456	0.6674 0.6541	0.6670 0.6543
Train + members	0.6712 0.6595	0.6684 0.6589	0.7605 0.7880	**0.7618** **0.8021**
Train + members + songs	-	-	0.7238 0.7072	0.7430 0.7707
Dimension reduction (PCA)	-	-	-	0.7305 0.7588

- Accuracy score
- ROC-AUC score

Among the features with the highest importance are mainly **user properties** such as days from registration to expiration, registration year, age, expiration year or gender and usability features related to **my library** and **playlists** such as source system tab my library, source screen local playlist more or online playlist more or source type local library or online playlist (as seen in Figure 5.2).

It is interesting to observe that our initial findings in the exploratory and interferal analysis regarding the relevance in the replay probability of usability features, such as source system tab ‘my library’ or source type ‘online playlist’, are also identify by the model.

In fact, in the decision tree classifier the source system tab ‘my library’ feature was the first decision (see Figure 5.3). In the initial findings we also pointed out that ‘my radio’ tab had the lowest replay probability among all tabs. In the decision tree the feature source type radio (similar) appears as one of the most important features.

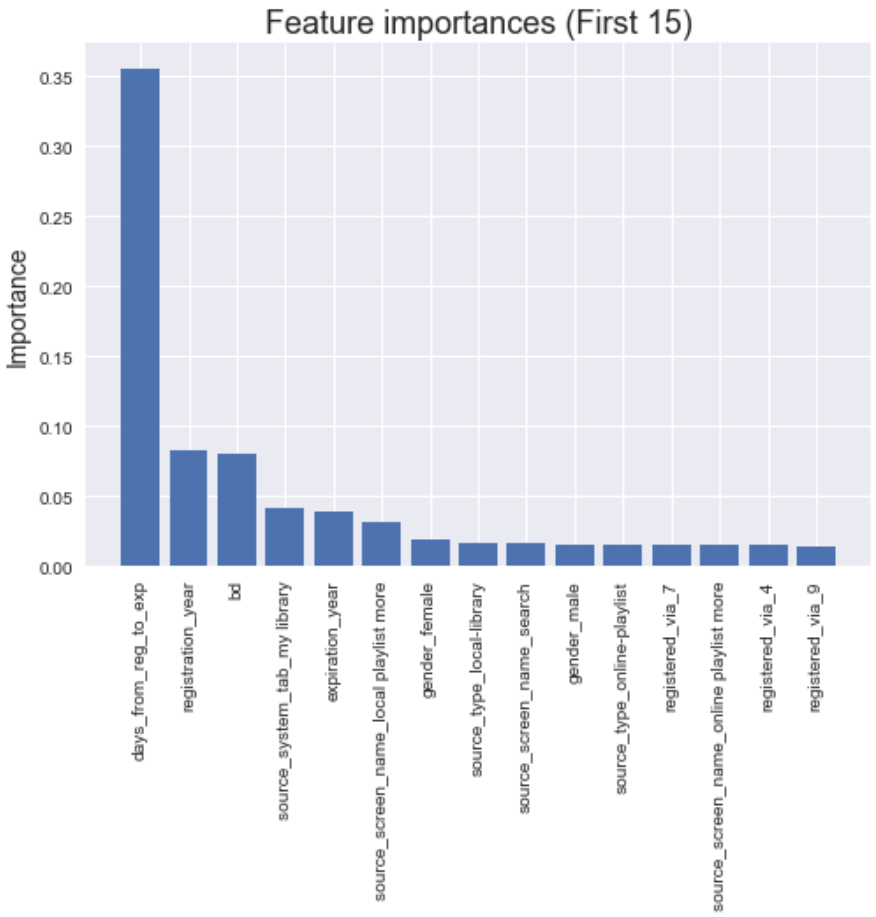


Figure 5.2

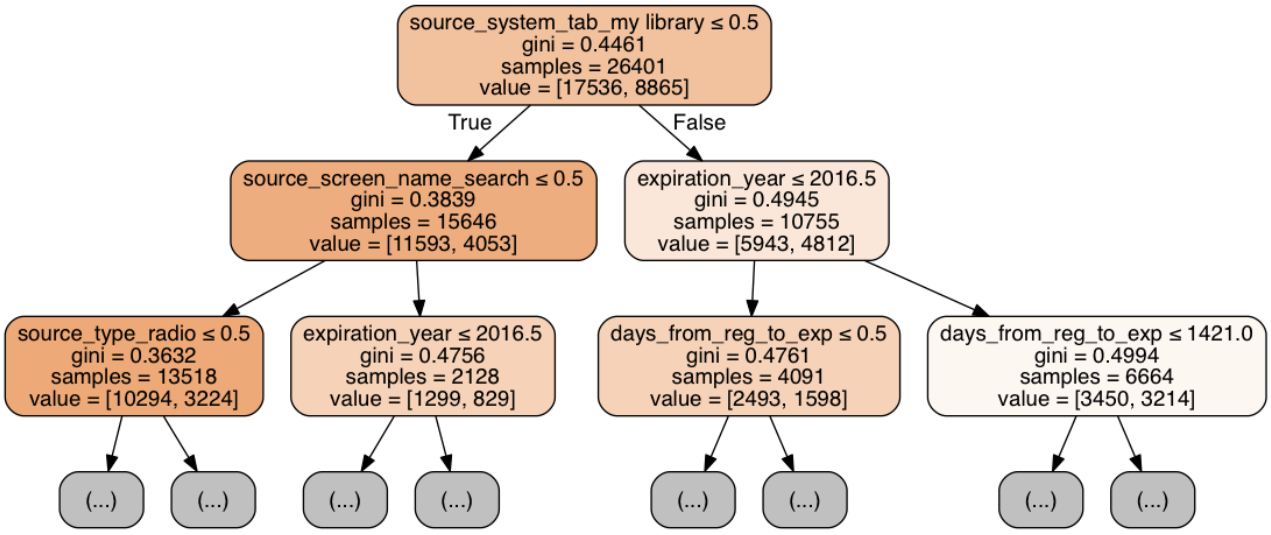


Figure 5.3

5.4. Hyperparameter tuning and overfitting

GridSearchCV function was used to fit the models. This function searches over a parameter grid and returns the mean k-fold cross-validated score of the best estimator.

In logistic regression and linear SVC models we have tuned the C parameter which is the inverse of the of regularization strength. Smaller values specify stronger regularization.

In decision tree the maximum depth of the tree was modified. In random forest both the number of trees (estimators) and the maximum number of features considered when looking for the best split were tuned. We also tried different number of features in the PCA dimension reduction.

In the logistic regression model, the tuning of the parameter C did not have much impact. Training and validation scores are very similar both when only usabilities features are considered and when also member features are included (see Figure 5.4).

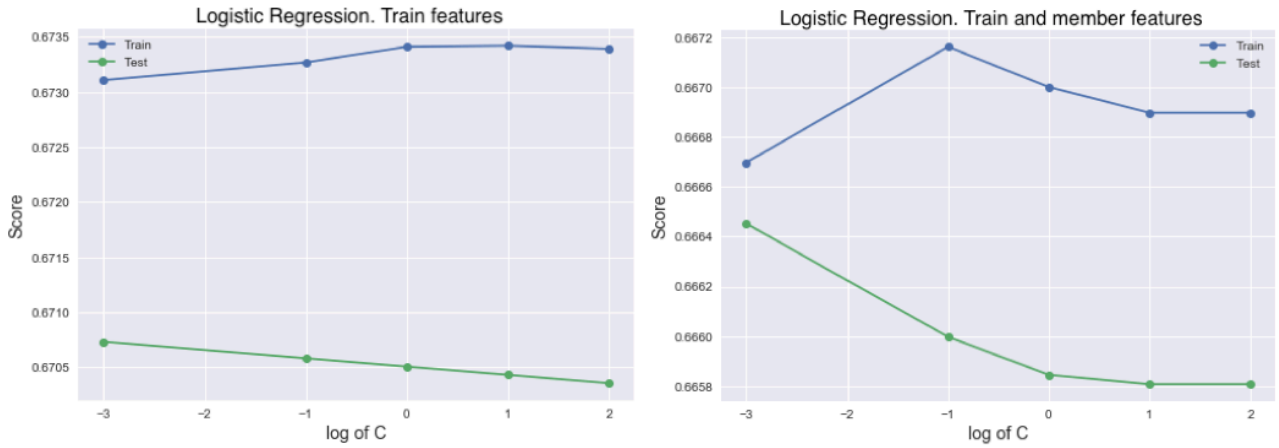


Figure 5.4

In the linear SVC lower values of C perform better (see Figure 5.5). Training and validation sets return similar scores so there are no overfitting problems.

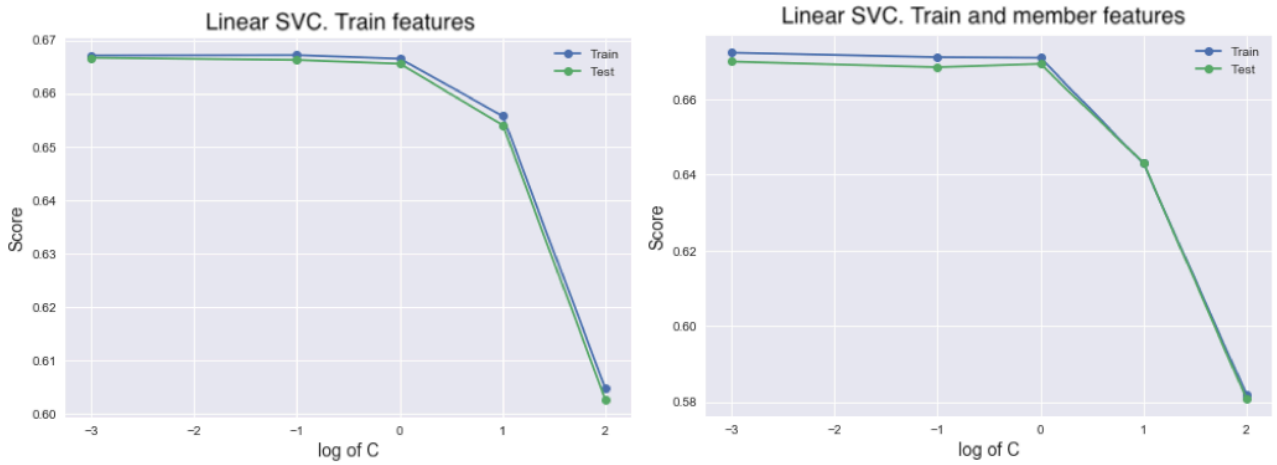


Figure 5.5

It is important to restrict the maximum depth of the tree. When only train and member features are considered, the test score is stable for maximum depth above 30. When we also introduce song features the test score for maximum depth above 20 decreases and the train score continues increasing, which proves that the model is overfitted (see Figure 5.6).

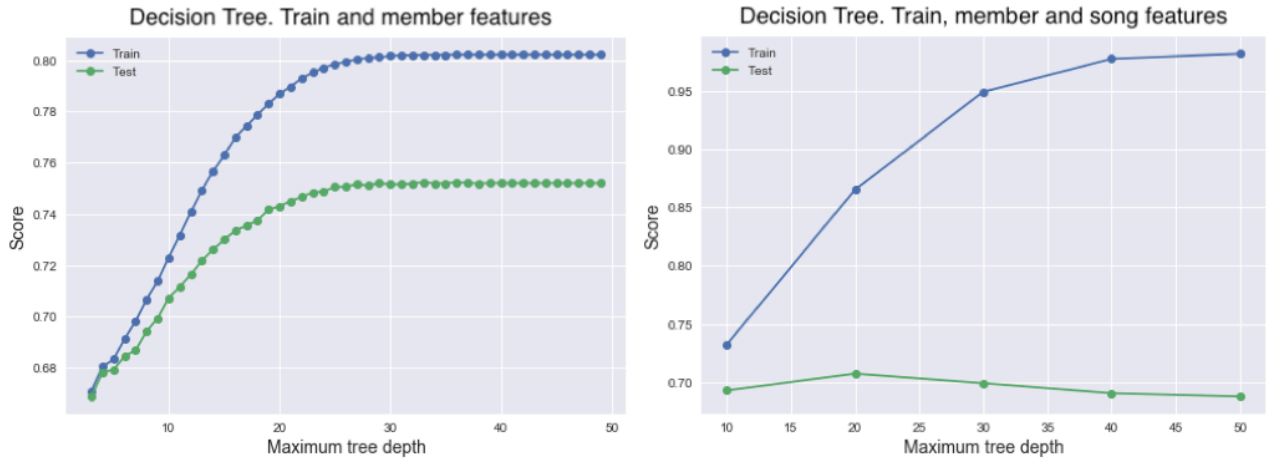


Figure 5.6

There is not a big change in the scores when the number of features is modified in the random forest model, it is more relevant the number of trees. When only usability (train) features are considered there are no big differences in training and test scores (see Figure 5.7). When also member features are included the number of trees affects the score more, being stable after 30 trees (see Figure 5.8).

Figure 5.7

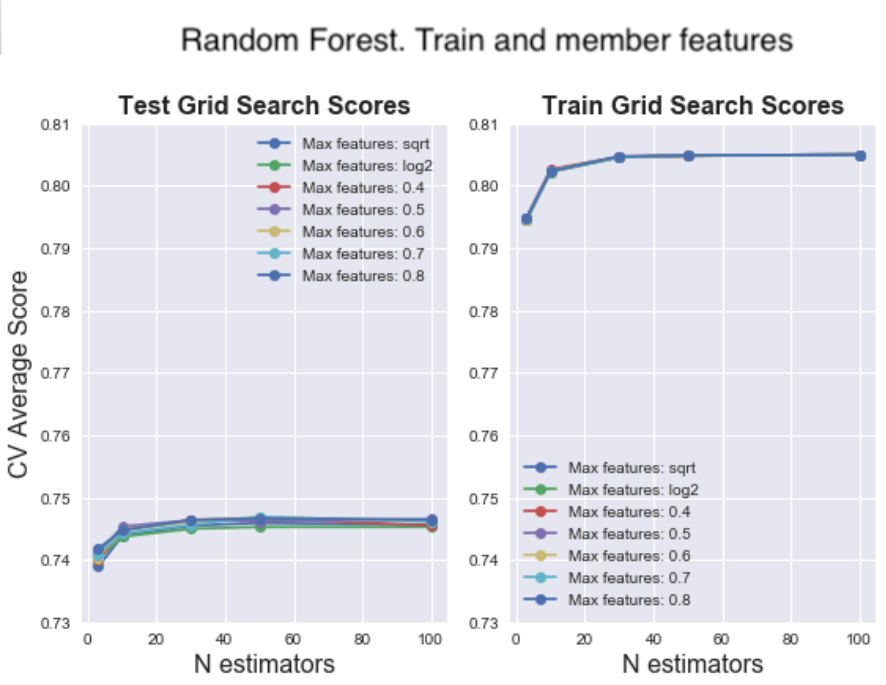


Figure 5.8

Dimension reduction with PCA was applied in the set including usability, member and song features. The model fitted was random forest with 1000 estimators. As we can see in Figure 5.9, if we select more than around 80 features the model starts overfitting.

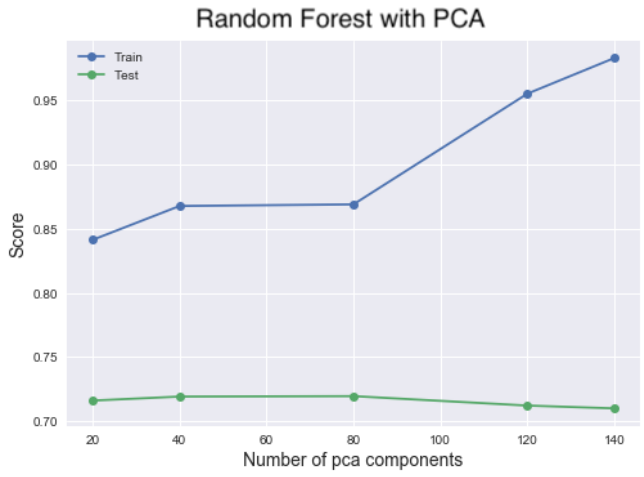


Figure 5.9

In Table 5.2 the best hyperparameters found in each case are summarized:

Table 5.2

	Logistic Regression	Linear SVC	Decision Tree	Random Forest
Train	C: 0.001	C: 0.001	Max depth: 18	Trees: 50 RF Features: sqrt
Train + members	C: 0.001	C: 0.001	Max depth: 33	Trees: 50 RF Features: 0.7
Train + members + songs	-	-	Max depth: 20	Trees: 2000
Dimension reduction (PCA)	-	-	-	PCA components: 80

6. Client recommendations

Improve member data collection

Only around 42% of users have gender and age values. Member features are the most relevant for the model performance. In the best classifier (random forest including member and usability features) the age was the 3rd most important feature and gender_female and gender_male, the 7th and 10th.

Teach users how to add songs to 'My Library' and interact with playlists

As we have observed in the feature importance results, features related with my library and playlists are very relevant.

In the first user experience, the application should make a big effort on teaching users how to add songs to 'My library' and how to interact with playlists. As alternatives, it is suggested that a tutorial could be added in the onboarding, users could be prompted to add first listened songs to their playlists, first listened songs could be added directly to their library or different icons to add songs to their playlist could be tested implementing the one that performs better.

Make less prominent the 'Radio' tab

Radio tab could be included inside another tab, since it is not that popular and also it is less likely that users will listen again these songs. E.g. Spotify mobile app only has 3 tabs ('home', 'search' and 'my library') and radio option is inside 'my library'.

7. Limitations and next steps

As mentioned before, machine learning models were only applied to a subset of the data. It would be interesting to **scale the analysis to bigger training datasets** and check if the performance of the model improves.

We concluded that the random forest was the best model from the ones we chose, but there are **other ensemble algorithms** that can be tested. Boosting algorithms such as XGboost or LightGMB are very popular in Kaggle competitions and very powerful. Another option would be to try voting ensemble which weights predictions from different sub-models.

It would be interesting as well to have the total number of times a particular song has been listened by a user. Currently we only have data for the first time the user has listened to a particular song and whether that user has listened or not that song again in a period of time. Because we are missing the total number of times that song was replayed, we can't identify **popularity** among the songs which definitely would improve our predictions.