Music Recommendation Analysis

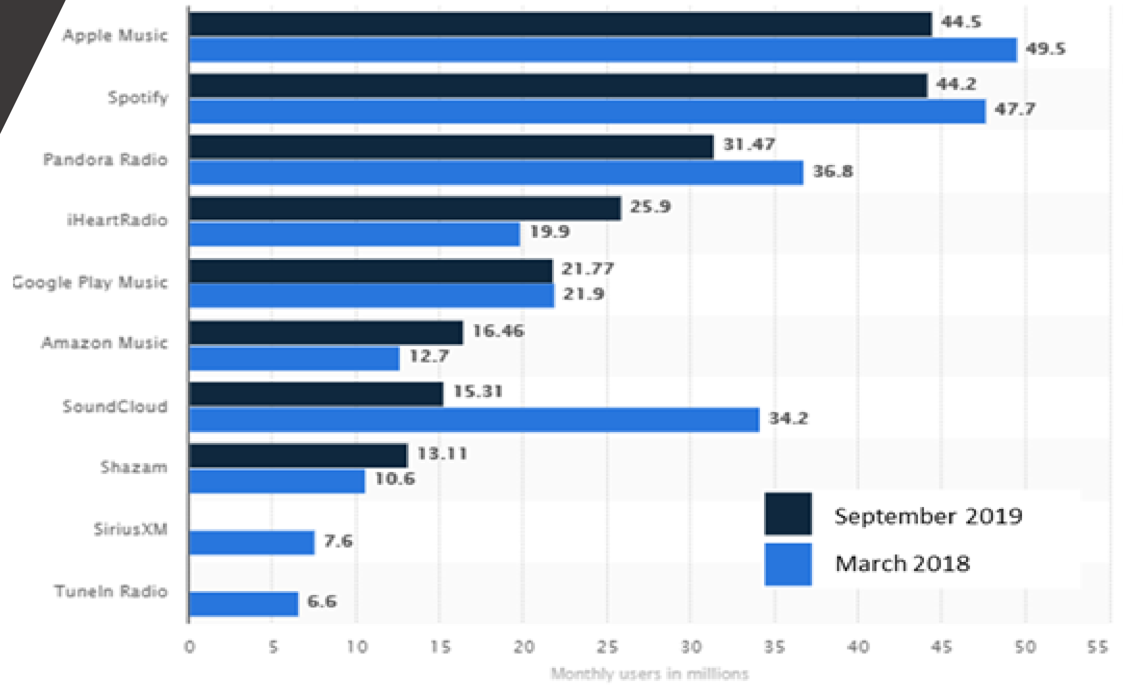
IST718

Professor Lando

Tim Abrams, Telly Cooper, Ryan Fischer & Brian Hogan

Specification

- The IoT age, in combination with an almost global access to internet streaming bandwidth has created a consumer base with unprecedented exponential growth in demand for audio content.
- Revenue from streaming services reached $8.8 billion in 2019, and now accounts for 79.5% of all recorded music revenues.
- A rapidly expanding and competitive market has been developed almost entirely in the last decade
- From March 2018 to September 2019 the streaming service providers Soundcloud, Shazam, SiriusXM and TuneIn Radio lost significant market share, and potentially long-term viability.
- As the top providers compete for a larger share of revenues it is critical that they “must get playlists right” through relevant recommendations based in each users’ preferences



Top U.S Music streaming services (millions of users)

Problem Statement

The music streaming market is highly competitive with little differentiation beyond the core product (songs and artists). This results in rapid loss of market share and business viability in ancillary areas- primarily song recommendation algorithms to encourage streaming and product consumption beyond the intended consumer time frames. Companies that cannot leverage their user and service data to provide highly relevant song recommendations to their listeners risk irrelevance in as little as eighteen months, as the average consumer has almost zero barriers to transition.

Hypothesis

Analysis of data about the listener, songs and services enables a music streaming service to predict if a given user will listen to a song more than once.

Data

The data was sourced from the music recommendation system Kaggle competition hosted by The ACM International Conference on Web Search and Data Mining (WSDM) and based on a donated dataset from KKBOX

Situation

- Team worked on assessing user likelihood of listening to a new song based on listening preferences
- Prediction focused on likelihood a user would like to a new song a minimum of 2 times
- Key factors include:
    - Membership information
        - Age, location, membership timeline, and listening history
    - Listening behavior
        - Genre music selected from
    - Song information
        - Artist, song, length, and genre

Obtain

- Data was sourced from the following
  - Challenge website: https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data
  - Additional resources used:
    - https://www.kaggle.com/myonin/music-recommendation-random-forest-xgboost
    - https://www.kaggle.com/rohandx1996/recommendation-system-with-83-accuracy-lgbm
    - https://www.kaggle.com/freshwater/basic-of-lgbm
- The initial data was then imported from csv files into dataframes for analysis:

df_train		df_test		df_songs		df_song_extra_info		df_members	
Shape: (7377418, 6)		Shape: (2556790, 6)		Shape: (2296320, 7)		Shape: (2295971, 3)		Shape: (34403, 7)	
msno	obj	id	int	song_id	obj	song_id	obj	msno	obj
song_id	obj	msno	obj	song_length	int	name	obj	city	int
source_system_tab	obj	song_id	obj	genre_ids	obj	isrc	obj	bd	int
source_screen_name	obj	source_system_tab	obj	artist_name	obj			gender	obj
source_type	obj	source_screen_name	obj	composer	obj			registered_via	int
target	int	source_type	obj	lyricist	obj			registration_init_time	int
				language	float			expiration_date	int

Scrub

The data is then cleaned:

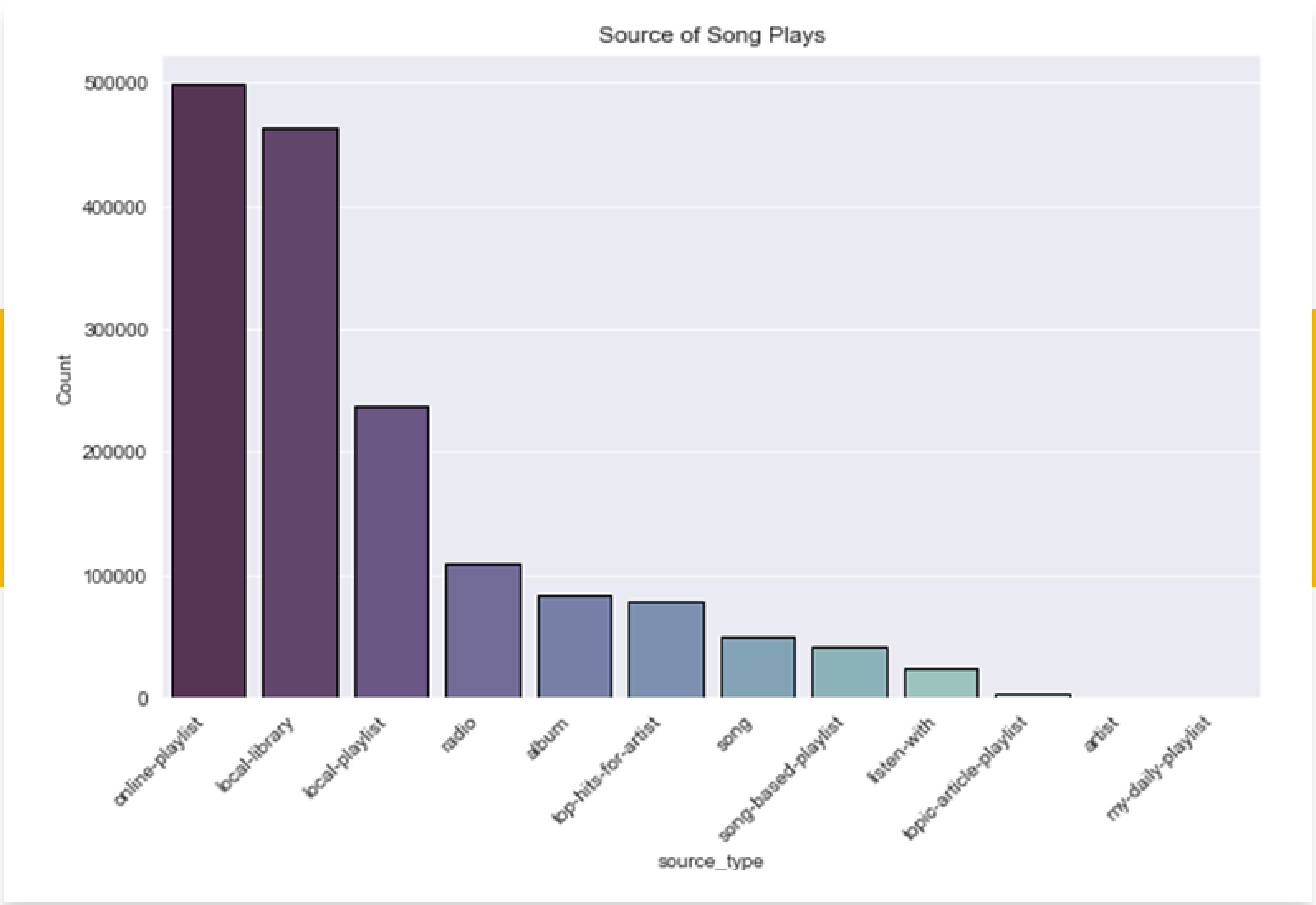
- Remove all non-English entries (as indicated by the float 52 in df_songs
- Remove all non-Roman characters and punctuation from artist_name, composer and lyricist in df_songs
- Replace all spaces with “_”
- Create new feature for membership days (expiration_date - registration_init_time)
- Remove all columns with NA values contributing to more than 1% of total values
- Remove all ages younger than 15 years and older than 55

Scrub

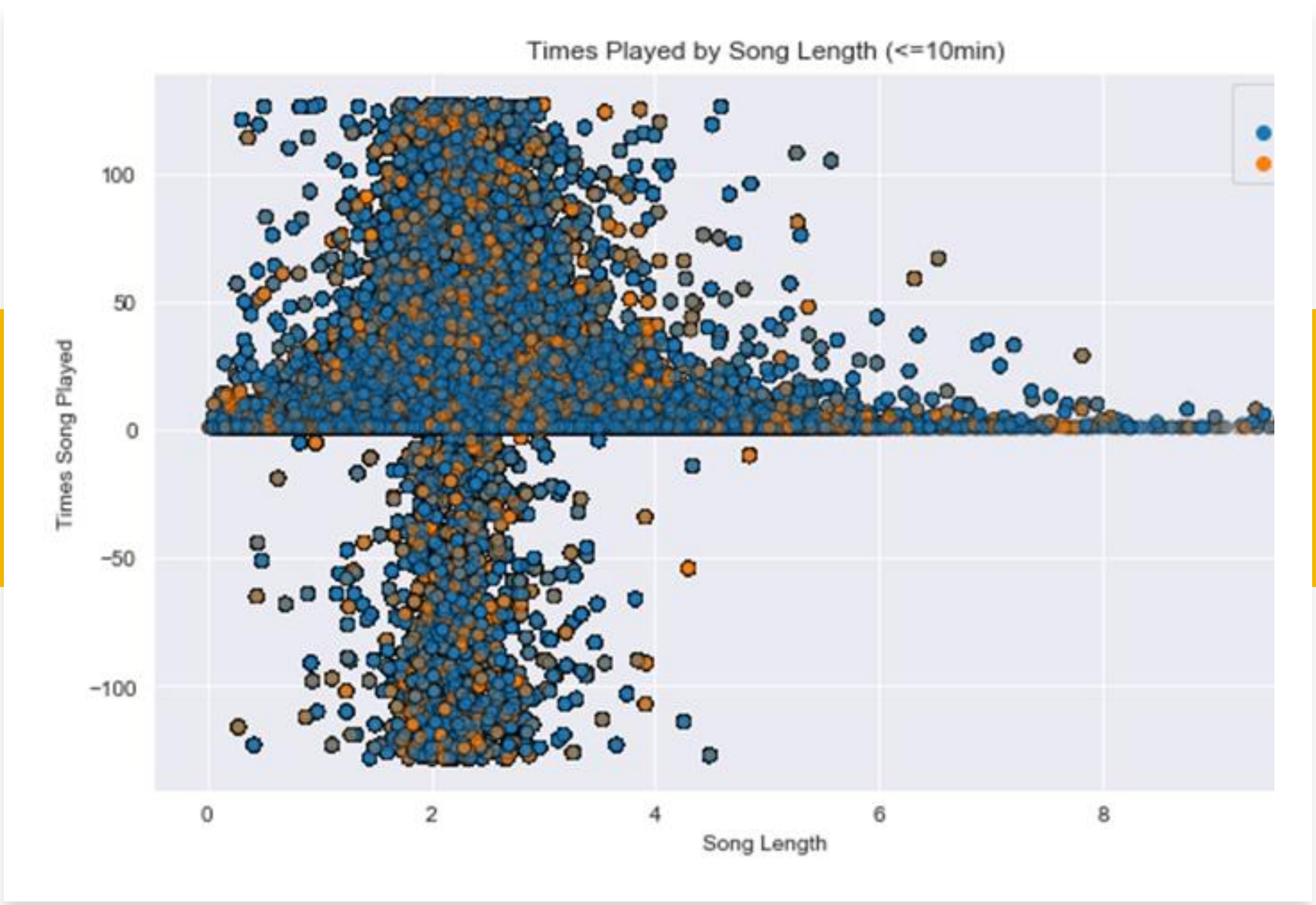
The df_train dataframe is then merged with features from df_songs, df_song_extra_info and df_members and the df_test dataframe is merged with features from df_songs, df_song_extra_info and df_members:

df_train_clean	Shape: (336702, 14) • msno: unique user ID • song_id: unique song id • source_system_tab: music app tab where the event was triggered • source_type: first play source (album, playlist etc) • song_length: in ms • genre_ids: genre category. Some songs have multiple genres and they are separated by | • artist_name: name of person that performed the song • composer: name of person that wrote the music for the song • lyricist: name of person that wrote the lyrics for the song • city: city where user lives • age: age of the user • registered_via', • membership_days: number of days the listener has been a member of the service • count_song_played (new feature) • count_artist_played (new feature) • target: target variable for prediction. Values: ○ 1 = member has listened to a song 2 or more times ○ 0 = member has listened to a song < 2 times

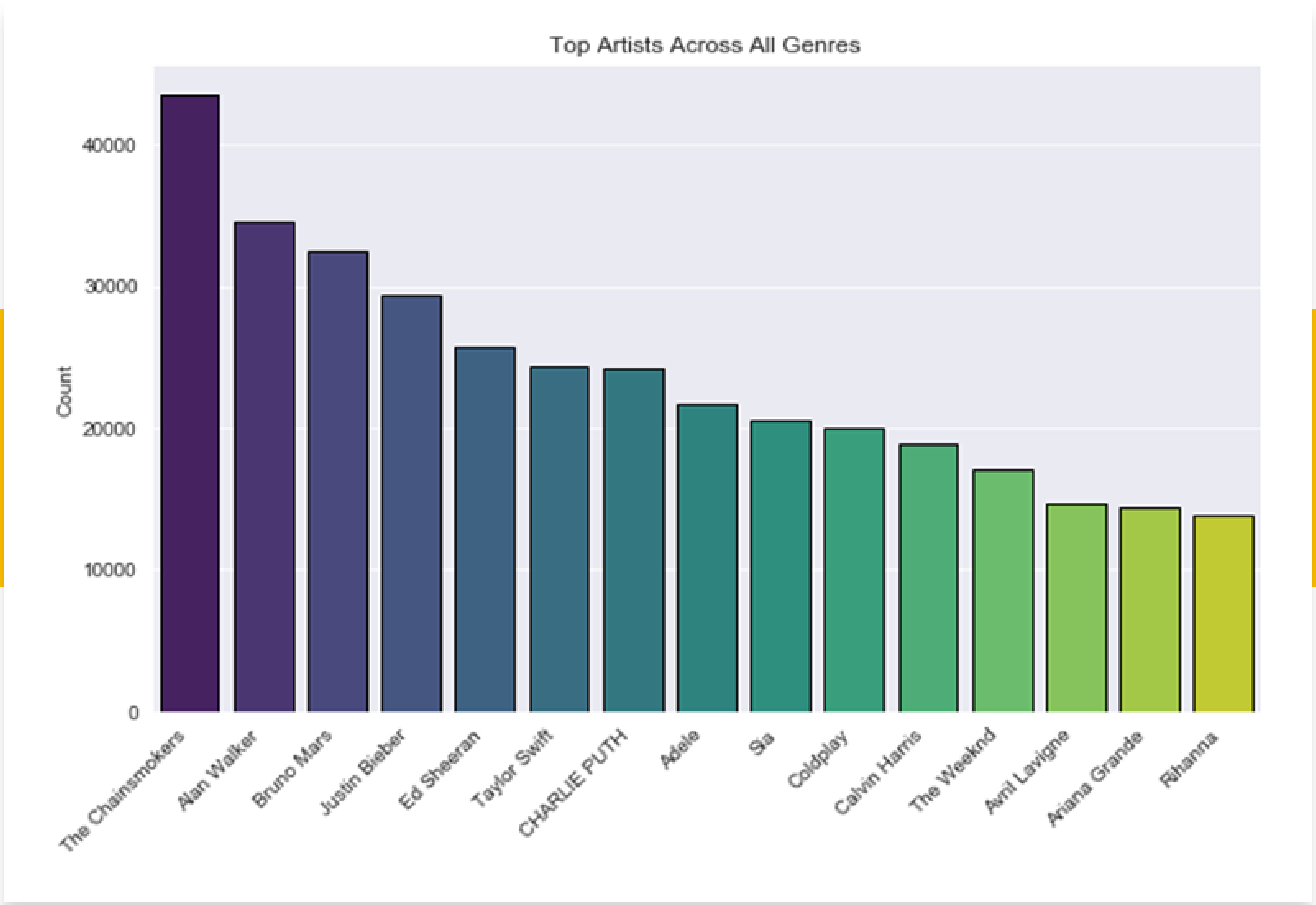
df_test_clean	Shape: (118280, 14) • id: row id • msno: unique user ID • song_id: unique song id • source_system_tab: music app tab where the event was triggered • source_type: first play source (album, playlist etc) • song_length: in ms • genre_ids: genre category. Some songs have multiple genres and they are separated by | • artist_name: name of person that performed the song • composer: name of person that wrote the music for the song • lyricist: name of person that wrote the lyrics for the song • city: city where user lives • age: age of the user • registered_via', • membership_days: number of days the listener has been a member of the service • count_song_played (new feature) • count_artist_played (new feature)


Source of Song Plays
Count
500000
400000
300000
200000
100000
0
online-playlist
local-library
local-playlist
radio
album
top-hits-for-artist
song
song-based-playlist
listen-with
topic-article-playlist
artist
my-daily-playlist
source_type

Explore


Times Played by Song Length (<=10min)
Times Song Played
100
50
0
-50
-100
0
2
4
6
8
Song Length

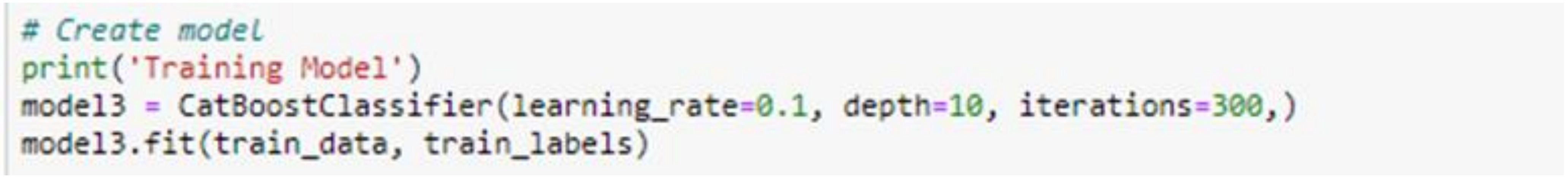
Explore


Top Artists Across All Genres
Count
40000
30000
20000
10000
0
The Chainsmokers
Alan Walker
Bruno Mars
Justin Bieber
Ed Sheeran
Taylor Swift
CHARLIE PUTH
Adele
Sia
Coldplay
Calvin Harris
The Weeknd
Avril Lavigne
Ariana Grande
Rihanna

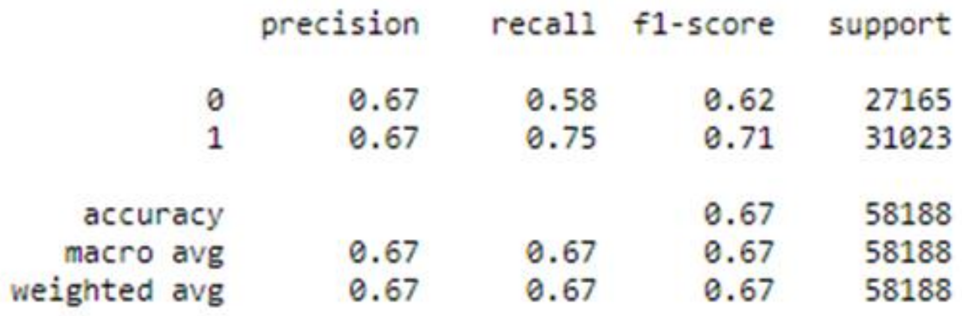
Explore

Model

- CatBoostClassifier
- The CatBoost Classifier was chosen as it is a performant classifier that can easily handle categorical features



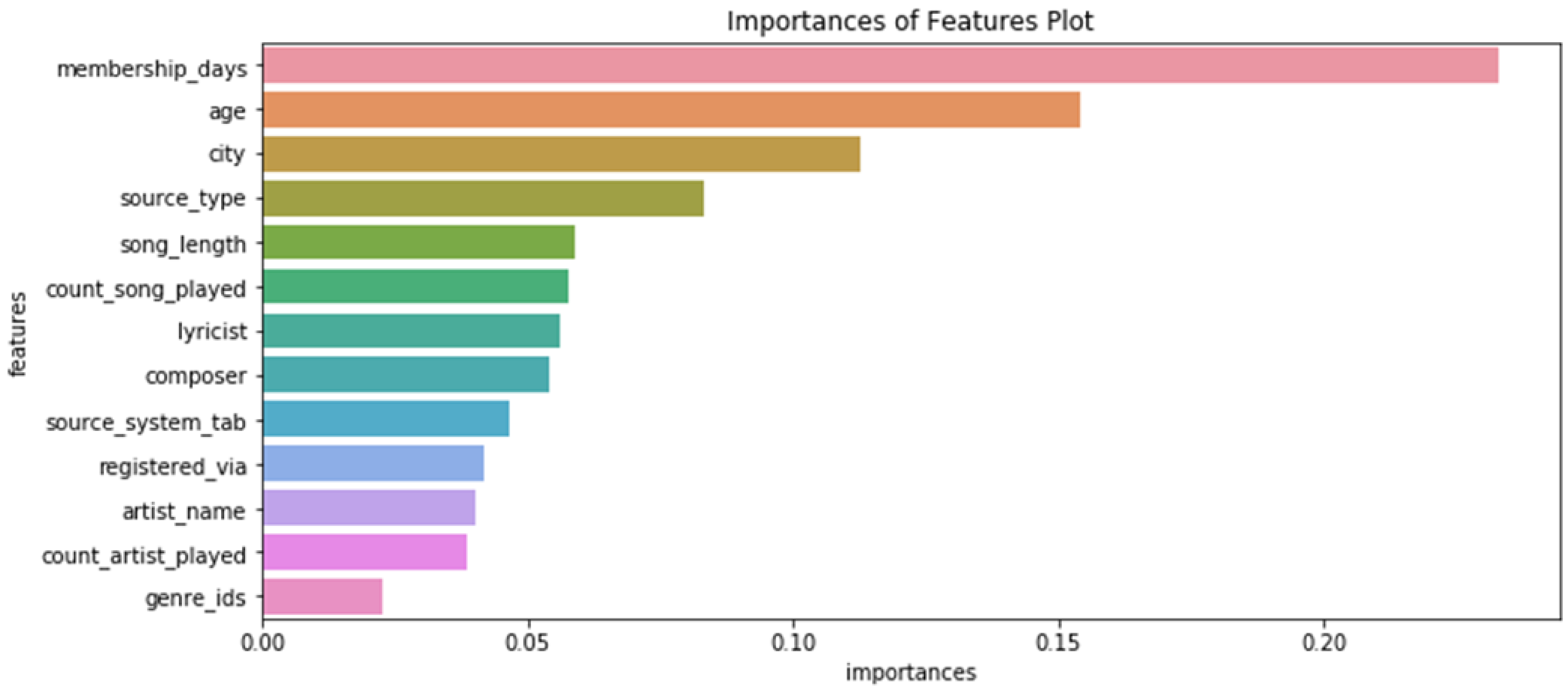
```
# Create model
print('Training Model')
model3 = CatBoostClassifier(learning_rate=0.1, depth=10, iterations=300,)
model3.fit(train_data, train_labels)
```



```
              precision    recall  f1-score   support

           0       0.67      0.58      0.62     27165
           1       0.67      0.75      0.71     31023

    accuracy                           0.67     58188
   macro avg       0.67      0.67      0.67     58188
weighted avg       0.67      0.67      0.67     58188
```

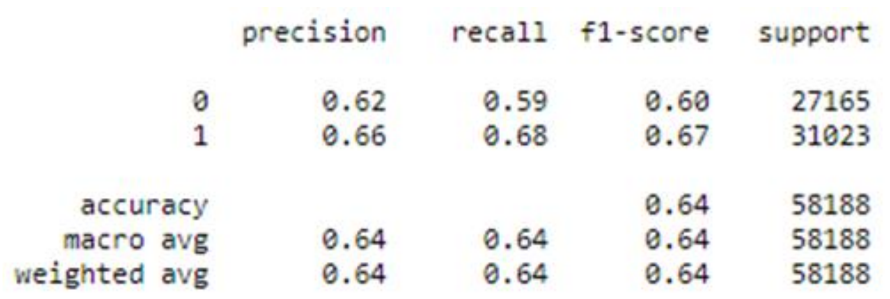
Model

- A Random Forest Classifier is used to identify the importance of features. User ID {mnso} and the song id (song_id) are removed from the analysis as they will not scale analysis across new users or songs. Additionally, features with importance of < 0.05 are removed due to their limited impact on analysis.



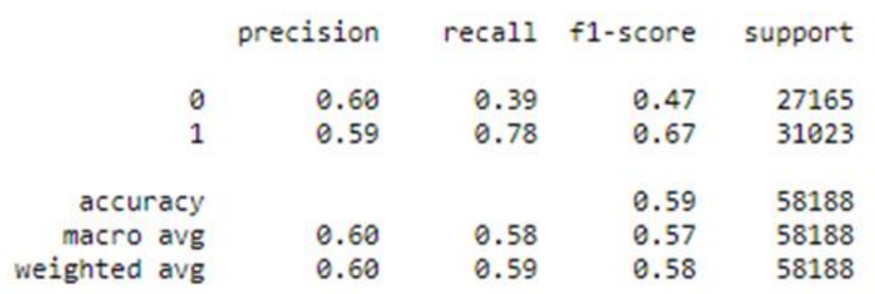
Model

- Random Forest Classifier
  - A Random Forest algorithm is chosen for its ability to handle large datasets with high dimensionality, the ability to manage missing values without significantly impacting the outcome and to minimize overfitting by averaging results.



```
              precision    recall  f1-score   support

           0       0.62      0.59      0.60     27165
           1       0.66      0.68      0.67     31023

    accuracy                           0.64     58188
   macro avg       0.64      0.64      0.64     58188
weighted avg       0.64      0.64      0.64     58188
```

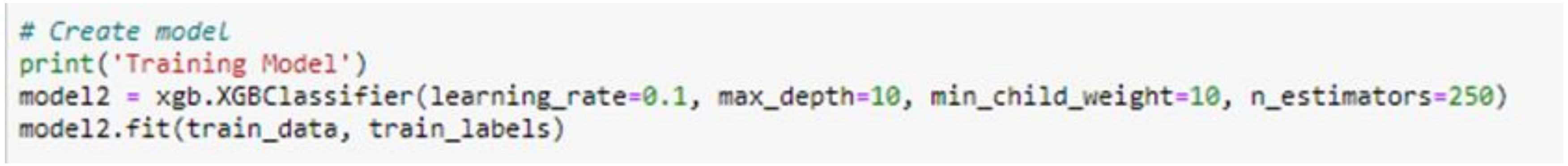
- Naïve-Bayes
  - Naïve-Bayes was chosen due to the fast training and low resource usage



```
              precision    recall  f1-score   support

           0       0.60      0.39      0.47     27165
           1       0.59      0.78      0.67     31023

    accuracy                           0.59     58188
   macro avg       0.60      0.58      0.57     58188
weighted avg       0.60      0.59      0.58     58188
```

Model

- XGBClassifier
  - Boosting ensemble techniques weight votes with a collection of classifiers by averaging the prediction over a collection of classifiers and combining a set of heterogeneous classifiers.
  - The XGB Classifier was chosen due to its speed and focus on minimizing potential loss for all possible splits and creates a new branch



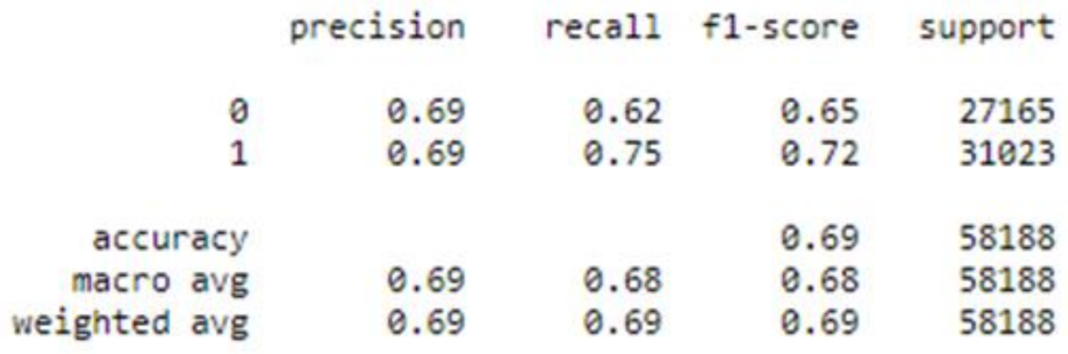
```
# Create model
print('Training Model')
model2 = xgb.XGBClassifier(learning_rate=0.1, max_depth=10, min_child_weight=10, n_estimators=250)
model2.fit(train_data, train_labels)
```

Parameters

n_estimators = dictates how many subtrees will be trained and helps to manage overfitting.

max_depth = maximum tree depth each individual tree h can grow to

learning_rate = manages through local minima

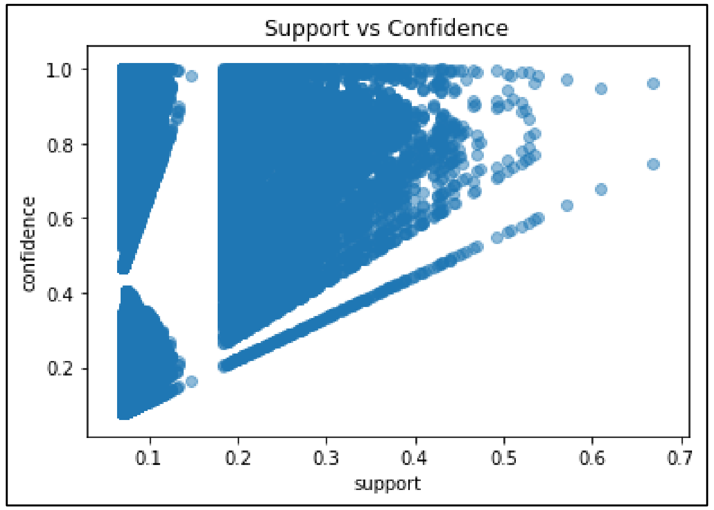


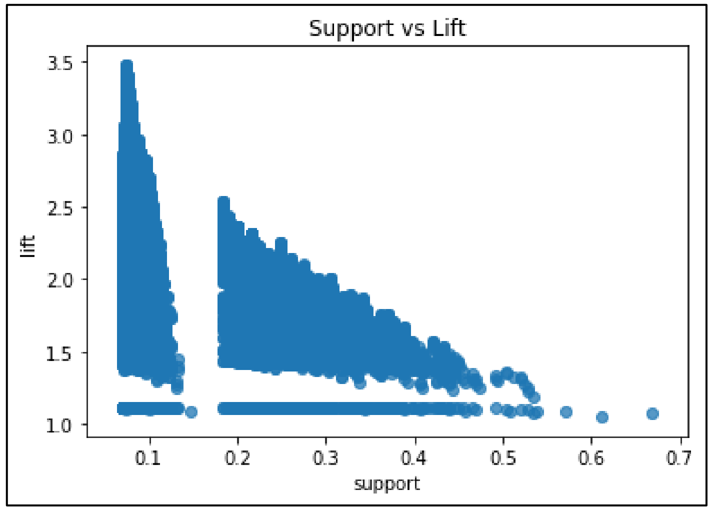
```
              precision    recall  f1-score   support

           0       0.69      0.62      0.65     27165
           1       0.69      0.75      0.72     31023

    accuracy                           0.69     58188
   macro avg       0.69      0.68      0.68     58188
weighted avg       0.69      0.69      0.69     58188
```

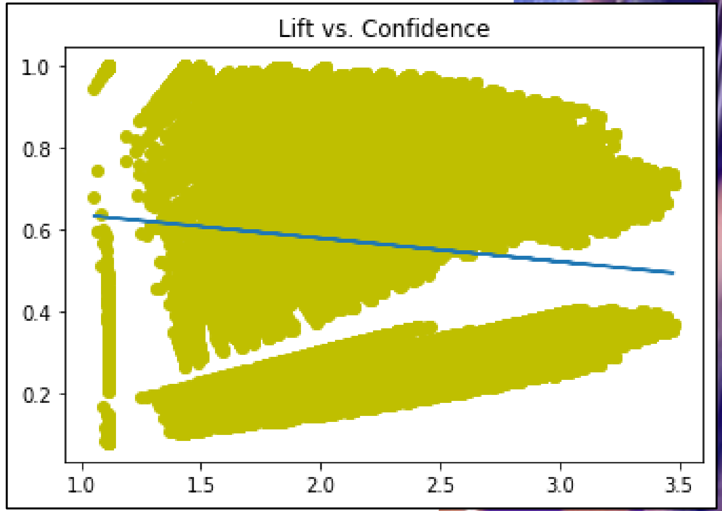
Association Rules

- Out of 300 genres country music is easiest to predict what new genre a listener would be most receptive to

Top Rules	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	Country	HipHop	0.347539634	0.588840848	0.301461808	0.867417062	1.473092541
1	HipHop	Country	0.588840848	0.347539634	0.301461808	0.511958042	1.473092541
2	Country	Contemporary	0.347539634	0.518797612	0.281737698	0.810663507	1.562581417
3	Contemporary	Country	0.518797612	0.347539634	0.281737698	0.543058973	1.562581417
4	Country	Dupstep	0.347539634	0.696191064	0.318632901	0.916824645	1.316915271
5	Dupstep	Country	0.696191064	0.347539634	0.318632901	0.457680251	1.316915271
6	Country	Electronic	0.347539634	0.557092856	0.278402306	0.801066351	1.437940448
7	Electronic	Country	0.557092856	0.347539634	0.278402306	0.499741296	1.437940448
8	Country	Pop Rock	0.347539634	0.471607988	0.253325098	0.728909953	1.545584406
9	Pop Rock	Country	0.471607988	0.347539634	0.253325098	0.537151838	1.545584406

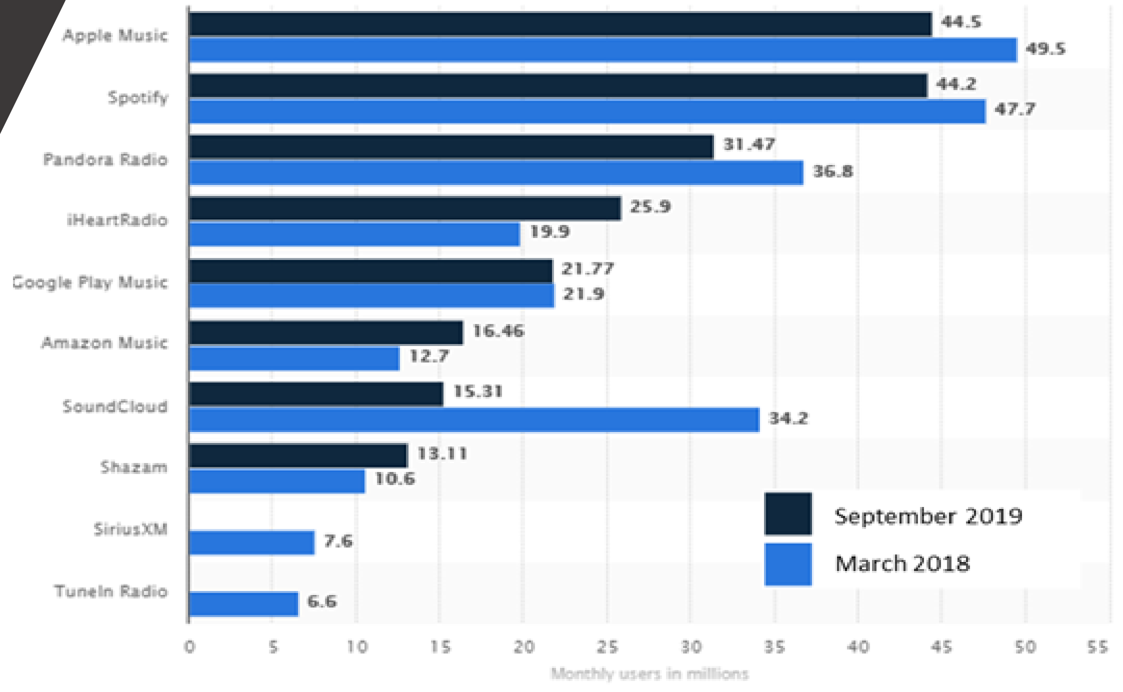






Conclusion

- XGBClassifier had the greatest success in properly classifying
- Length of membership and age had greatest impact on classifiers



Top U.S Music streaming services (millions of users)