Automated Music Playlist Generator

Arka Sarkar
arka18222@iiitd.ac.in

Pankil Kalra
pankil18061@iiitd.ac.in

Daksh Thapar
daksh18137@iiitd.ac.in

November 22, 2020

Abstract

With the growing popularity of music streaming services like Spotify, Apple Music and Wynk, the number of songs have skyrocketed globally. Creating personalized playlists for users has become tedious and challenging as it involves individual listening to various songs and categorizing them based on their audio features. The objective is to sort songs with similar musical characteristics into playlists automatically. Modern machine learning techniques and visualization tools should help us find accurate models that categorize millions of songs into user playlists based on song choices. Related works on this problem didn't consider Lyrical Analysis while making playlists. We are using Topic Modelling techniques on lyrics, and will be using the extracted topics as a feature for generating playlists.

1 Introduction

We propose Binary Classification models that we can use for every playlist that can classify whether a new song belongs to that playlist or not. Also, we have implemented a single Multi Class Classification model which can classify a song into one of many different playlists. This classification process has diverse practical applications as it automates the assignment of playlists to songs of similar characteristics, improving the user experience and easing the process of music playback. Graph-Based approaches will be used to generate playlists containing songs with similar features from a collection. In retrospect, our work can find key applications among music enthusiasts and in mobile applications to make song collections for not only individuals, but for age groups and people with similar interests for song genres and artists

2 Literature Survey

Playlist generation is a broad problem which has been approached in many different ways.

1. Automated Playlist generation from Personal Music Libraries [1] by Diana Lin and Sampath Jayarathna provides a method to use K-means clustering, Affinity Propagation and DBSCAN on various audio features (such as "danceability", "energy", "acousticness") to generate an automated playlist.

2. A Comparison of Playlist Generation Strategies for Music Recommendation and a New Baseline Scheme [2] by Geoffray Bonnin and Dietmar Jannach uses multiple methods such as KNN, Popularity-based, Same artists - greatest hit, Collocated artists - greatest hit, Content-based approaches to generate playlists.

3. Music Playlist Generation based on Community Detection and Personalized PageRank [3] by Bangzheng He, Yandi Li and Bobby Nguy uses song similarity graph approaches and apply community detection and Personalized PageRank to generate playlists.

3 Dataset Features

We have picked the Million Song Dataset (MSD lyrical) which consists of 2,33,662 songs with lyrics. We collected the Top Search Results of playlists using phrases like 'Summer', 'Love,Party', 'Breakup', 'Rock', 'Country', 'Romance', 'Rap', 'Metal', 'Hip-Hop', 'Latin', 'Blues', 'Soul', 'Classic', 'Pop', 'Jazz', 'Folk', 'RB' from Spotify API which have a significant song overlap with the MSD dataset. In total, we obtained 169 playlists amounting to 11,159 unique songs.

We extracted 14 audio features for each song using Spotify API as shown below.

Feature	Datatype
danceability	float
energy	float
Key	int
loudness	float
mode	int
speechiness	float
instrumentalness	float
liveness	float
valence	float
duration_ms	int
release_date	int
popularity	int

Table 1: Raw Features

We performed Topic Modelling and extracted additional 20 features using Latent Dirichlet Allocation (LDA) on the lyrics which will be described in the subsection below.

Example song :
{ 'song_id': 73etijhz7pV4Wx7GTANLpq, 'danceability': 0.332, 'energy': 0.371, 'key': 4, 'loudness': -9.884, 'mode': 1, 'speechiness': 0.0396, 'acousticness': 0.729, 'instrumentalness': 0.00021, 'liveness': 0.176, 'valence': 0.375, 'tempo': 173.273, 'duration_ms' : 249093 , 'release_date': 2002, 'popularity' : 51 }

The class distribution of playlists is shown below :

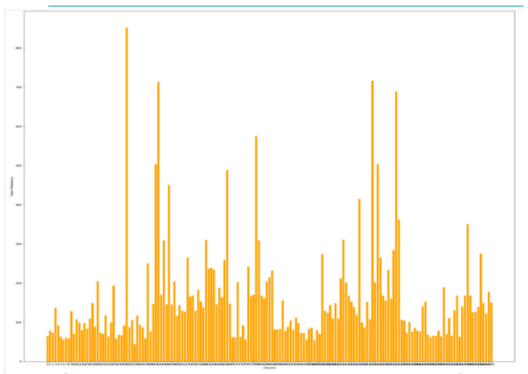


Figure 1: Class Distribution of Playlists.

3.1 Preprocessing, Feature Extraction

The lyrics obtained from Million Song Dataset were already preprocessed to some extent. The words had been stemmed and lyrics were converted to Bag of Words format. We further removed Stopwords and words with less than 3 characters for better topic creation.

3.1.1 Topic Modelling using Latent Dirichet Allocation (LDA)

We preprocessed our lyrics, applied topic modelling using Latent Dirichlet Allocation (LDA) and extracted the probabilities of each topic in every song. Latent Dirichlet allocation is a generative statistical model which provides us with a fixed number of unobserved topics which would help in analysing some similarity between playlists.
Latent Dirichlet Allocation (LDA) was applied on the lyrical corpus, we tried different values for the number of topics ranging from 3 to 30. The topics obtained were evaluated using Covariance 'c_v' score. *"20 number of topics"* gave the best c_v score of 0.58 and we extracted the probabilities of each topic in every song for the same. 20 new columns were added to our dataset containing the topic wise probabilities for each song.

Sample topic

Topic 18: 0.035*"dead" + 0.032*"kill" + 0.024*"head" + 0.022*"black" + 0.022*"blood" + 0.016*"fight" + 0.014*"hate" + 0.013*"lyric" + 0.013*"readi" + 0.013*"death"

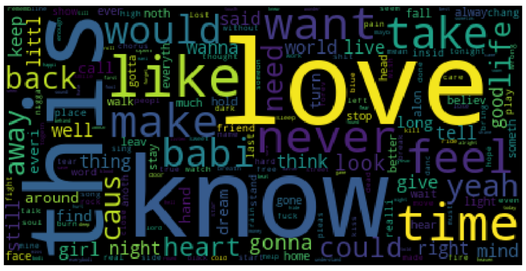


Figure 2: Word Cloud on Lyrics.

3.1.2 Dimensionality Reduction using Principal Axis Component (PCA)

Principal Component Analysis is a method of reducing dimensions of a dataset by transforming a large set of variables into a smaller set of variables that still contains most of the information in the larger data set. The various steps in PCA include standardization, computation of covariance matrix and eigenvectors to identify principle components. PCA can be thought of as fitting a " p-dimensional ellipsoid" to the data; every axis of the ellipsoid means a single principal component. For our project, we are working with 30 principle components.

3.1.3 Visualizing High Dimensional Data using t-SNE

t-Distributed Stochastic Neighbour Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data. t-SNE, unlike PCA, is not a linear projection. It uses the local relationships between points to create a low-dimensional mapping. This allows it to capture non-linear structure. t-SNE creates a probability distribution using the Gaussian distribution that defines the relationships between the points in high-dimensional space.

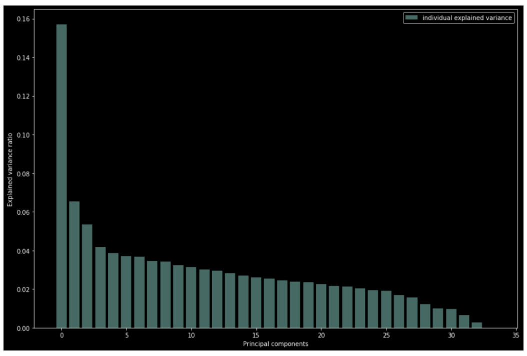


Figure 3: PCA Explained Variance

3.1.4 Data Standardization

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. Here's the formula for standardization:

$$z_i = \frac{x_i - \bar{x}}{s} \tag{1}$$

$\bar{x}$ is the mean of the feature values, σ is the standard deviation of the feature values.

4 Methodologies

Our objective is to generate automated song playlists for users using Label Classification and Graph-based techniques. For the Classification problem, two different methodologies were used. We tried different ML based classification algorithms both for a single-class prediction for 169 playlist and multi-class prediction for 13 playlists.
For playlist generation, we used clustering and graph-based approaches to construct playlists based on the feature similarity of songs from certain seed songs.

4.1 Classification

We applied binary classification on a total of 169 playlists. We used a One vs All approach where the target playlist was considered positive and all other playlists were considered negative and randomly undersampled. We performed 60:20:20 train-validation-test split and stratified sampling on the dataframe constructed.
For multi-class classification, we took all the non overlapping songs among various playlists; the total songs reduced from 11159 to 6120 for 169 playlists. We filtered out playlists having more than 100 songs for our model training.

We used the following classification models: Logistic Regression, Decision Trees, Random Forests, Linear SVC, XGBoost Ensemble technique, K-Nearest Neighbours Classifier and Artificial Neural Network. To optimise various parameters in the aforementioned models, we applied GridSearchCV and 10-fold cross validation.

4.2 Playlist Generation

We performed data standardization, dimensionality reduction using Singular Value Decomposition (SVD) and used the following graph- based techniques for Playlist visualization and generation.

4.2.1 K-Means Clustering

We implemented K-Means Clustering algorithm, one of the most common unsupervised Machine Learning algorithm.K-Means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. The algorithm identifies k number of centroids, and then each data point is assigned to its nearest cluster, while keeping the centroids as small as possible.

4.2.2 K Nearest Neighbours

Each song was represented in an N-dimensional space according to our features. The we selected the next K songs for that playlist based on the Euclidean distances measured from the average(centroid) of all of the seed songs and returned these nearest songs as a playlist collection to the user with song and artist names.

5 Results and Analysis

5.1 Classification

5.1.1 Binary Classification

For Binary Classification we used all 169 playlists.We used a One vs All approach where the target playlist was considered positive and all other playlists were considered negative and randomly undersampled.
The XG Boost algorithm outperforms all the models with an Average F1 score of 81.0% across all playlists. Linear SVC and Logistic Regression also gave very good results posing F1 (macro) scores 78.6% and 79.2% respectively. Metrics for each model are shown in the table below:

Model	Precision	Recall	F1
SVC	0.791	0.787	0.786
LR	0.806	0.788	0.792
DT	0.751	0.750	0.748
RF	0.839	0.758	0.774
XGB	0.834	0.800	0.810
KNN	0.774	0.728	0.726
ANN	0.769	0.781	0.771

Table 2: Binary Classification Scores

The training and Validation learning curve for SVC is shown below :

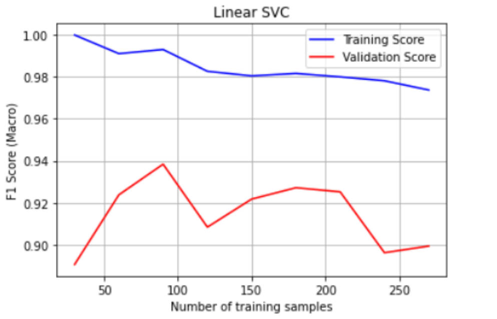


Figure 4: Learning Curve for SVC

The ROC curve is also plotetd below and shows an excellent area under the curve.

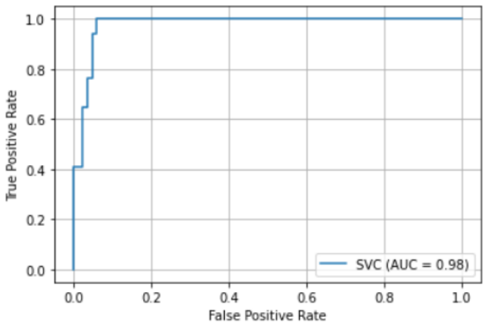


Figure 5: ROC curve for SVC

Model	Precision	Recall	F1	Acc
SVC	0.526	0.494	0.502	0.515
LR	0.506	0.503	0.499	0.524
DT	0.423	0.419	0.419	0.440
RF	0.623	0.549	0.565	0.586
XGB	0.602	0.511	0.587	0.602
KNN	0.454	0.430	0.419	0.480
ANN	0.476	0.500	0.401	0.577

Table 3: Multi Classification Scores

5.1.2 Multi Class Classification

For multi-class classification we took 13 non-overlapping playlists each having more than 100 songs. XGB in this case also got the best performance posing a F1 score of 58.7%.
XBG performed well in our case as it is an ensemble method and XGBoost improves upon the basic Gradient Boosting Method framework through systems optimization and algorithmic enhancements.
We also computed the feature importance of the features which are shown below :
As we can see from the above feature the topics obtained from the lyrics data had the highest importance (by a significant margin) for song classification. This implies that lyrics of a song are an important attribute which an user takes into account while creating a playlist. Also ”release_date”, ”acousticness” and ”energy” are important attributes of a song.

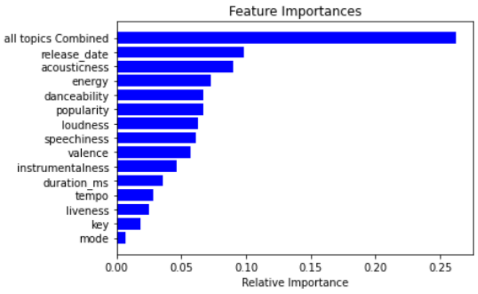


Figure 6: feature Importance for Random Forest multi classification

5.2 Playlist Generation

5.2.1 K-Means Clustering

We took the song corpus consisting of 11159 songs and applied SVD to get 15 component features for each song. We ran K-Means clustering algorithm for k ranging from 2 to 10. We would report the average distances of the songs in each playlist .
As a baseline we would be using the average distance of songs in our training playlist. The results are as follows :

We can observe lower mean Euclidean distance between the song nodes of our clustered playlists which implies we have achieved really good quality playlists comparable to actual top playlists on Spotify. The playlist generated from can be visualised using t-SNE as shown in figure 7 below :

Measure	K-Means	Spotify
Mean Distance	3.435	3.468

Table 4: Mean Distances

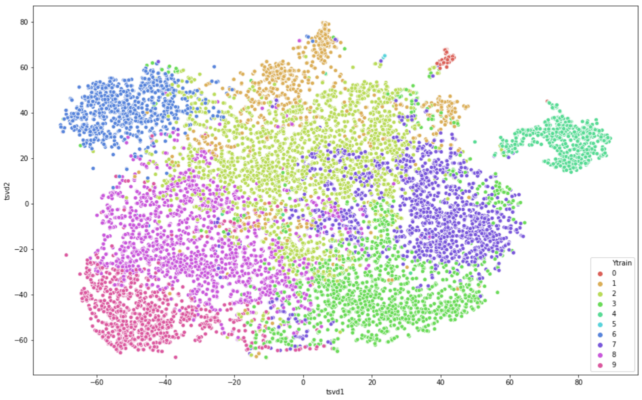


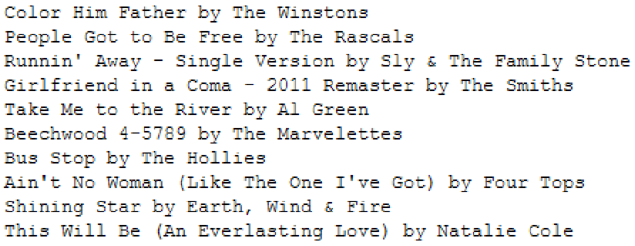
Figure 7: Cluster for k = 10

The Figure 7 cluster diagram shows us a beautiful distribution of playlist clusters and shows clear distinction between the K playlists. We can clearly distinguish individual playlists with decision boundaries. The above diagram signifies the success of our clustering algorithms and feature extraction processes; we are able to distinguish songs into playlists on the basis of our extracted features from Spotify.

5.2.2 K Nearest Neighbours

For playlist generation, we sent as an input the information of a country genre song: 'On the Road Again', Bob Dylan

For this corresponding input, the automatic playlist generator based on K Nearest Neighbours, where K=50 returns the following generated playlist as output (10 songs here):



```
Color Him Father by The Winstons
People Got to Be Free by The Rascals
Runnin' Away - Single Version by Sly & The Family Stone
Girlfriend in a Coma - 2011 Remaster by The Smiths
Take Me to the River by Al Green
Beechwood 4-5789 by The Marvelettes
Bus Stop by The Hollies
Ain't No Woman (Like The One I've Got) by Four Tops
Shining Star by Earth, Wind & Fire
This Will Be (An Everlasting Love) by Natalie Cole
```

Figure 8: Songs Generated

We can clearly observe that our output generated playlist contains songs that are very similar to the input songs. On closer manual observation for this example, we see that the output playlist's songs are majorly of the 'country' genre, have slower rhythmic melodies, similar classic instrumentals and having similar lyrics and wordings.

6 Conclusion

6.1 Outcomes

We are pleased to inform that our initial hypothesis that lyrics play an significant role in curating playlists was in fact true. This can be clearly observed by looking at the feature importance graphic shown in the results section. The metric scores obtained on our binary and multi classification were very promising and posed a higher or similar score with respect to the previous work done on this topic.

One of the limitations of the MSD dataset is that these contain information including audio features and song lyrics before 2011, and it does not contain all the trending songs from this era. These datasets mostly contains songs of the genres 'Rock' and 'Metal'.

The scores obtained by us shows the importance of audio features and lyrics in classifying playlists. Future improvements on the approach can be made by gathering more features and even creating derived features to improve the performance.

6.2 Work Left

The project progress has been on schedule and a lot of work has been done, the only part remaining from our initial proposed timeline is Hyperparameter Tuning and optimising our model for Overfitting and Underfitting.

6.3 Member Contribution

1. Arka Sarkar: Dataset Preprocessing- Data Extraction from Spotify API Topic Modelling- LDA Dimensionality Reduction- PCA EDA- Word Cloud Literature Review, GridSearchCV, Logistic Regression and Analysis, Support Vector Machine and Analysis, K-Means Clustering, Metrics comparison for Graph- based Techniques

2. Daksh Thapar: Dataset Preprocessing- Feature Extraction, removing stop words Topic Modelling- LDA Data Visualization for Playlists- t-SNE EDA- Data Correlation Heatmap Literature Review, GridSearchCV, Decision Tree and Analysis, XGB and Analysis, K- Nearest Neighbours, Metrics comparison for ML classification models

3. Pankil Kalra: Dataset Preprocessing- Extracting song lyrics from Million Song Dataset (MSD) Topic Modelling- LDA Dimensionality Reduction- PCA EDA- Feature Histogram Literature Review, GridSearchCV, KNN and Analysis, ANN and Analysis, K- Nearest Neighbours, ROC Plots

7 References

[1] Lin, D. and Jayarathna, S. (2018). Conferences 2018 IEEE International Confe... Automated Playlist Generation from Personal Music Libraries. IEEE International Conference on Information Reuse and Integration (IRI). [online] Available at: https://ieeexplore.ieee.org/abstract/document/8424710.

[2] Bonnin, G. and Jannach, D. (2013). AAAI 2013 Workshop. [online] Available at:https://www.aaai.org/ocs/index.php/WS/AAAIW13

[3] Music Playlist Generation based on Community Detection and Personalized PageRank. Stanford University Social and Information Network Analysis Autumn 2015. [online] Available at: http://snap.stanford.edu/class/cs224w-2015/

[4] Pichl, M., Zangerle, E. and Specht, G. (2017). Understanding Playlist Creation on Music Streaming Platforms. IEEE International Symposium on Multimedia (ISM). [online] Available at: https://ieeexplore.ieee.org/document/7823674.

[5] Pampalk, E. and Gasser, M. (2006). An Implementation of a Simple Playlist Generator Based on Audio Similarity Measures and User Feedback. ISMIR 2006, 7th International Conference on Music Information Retrieval. [online] Available at: https://www.researchgate.net/publication/