MIE368 - Analytics in Action: Preliminary Report

## Content-Based Spotify™ Playlist Recommendation System

November 8, 2024

*Team 20*

### Hussein Ismail (1008747910)

### Marco Mastrangelo (1009133518)

### Devraj Solanki (1009065707)

### Yahia Hussein (1008956444)

**1.0 Background:**Providing good song recommendations to users is a challenge that many music streaming platforms face [1]. On popular streaming platforms such as Spotify, collaborative filtering techniques are mainly used to recommend new songs to users [2], where recommendations made for a user are based on a similar user’s likings. The team proposes a content-based recommendation system [3] that suggests new songs to add to a playlist based on its existing content.

**2.0 Data Collection:**

To collect the music data for the models, the team utilised Spotify’s Web API [4] to scrape data from 22081 unique songs. To retrieve these songs, a dataset of the top 2500 most streamed Spotify artists was utilised [5]. For each of these artists, the artist’s top 10 most streamed tracks were retrieved, resulting in a total of 25000 songs. However, because of collaborations between artists, multiple artists shared top 10 tracks. Therefore, these duplicated tracks were removed from the dataset, resulting in 22081 songs. General information about the songs such as the track name, artist name(s), and release date were retrieved. Some relevant numerical metrics that were retrieved for each song include: ‘instrumentalness’, ‘acousticness’, ‘speechiness’, ‘danceability’, ‘valence’ (all numeric values, ranging from 0 to 1) , ‘popularity’ (ranging from 0 to 100), and ‘tempo’ (measured in BPM)[4]. Some other metrics that were scraped were deemed to be irrelevant for the project, such as 'uri', 'id', 'track\_href', 'analysis\_url'.

**3.0 EDA:**

The following section outlines each of the processes involved in EDA, including data cleaning, various plots, and analyzing feature interactions.

*3.1 Data Cleaning:*

To clean the data, routine checks were carried out, such as confirming data types and removing remaining duplicates. Furthermore, after checking for missing data, the team discovered 4 null values in the ‘release date’ column. The team imputed by cross referencing the ‘tracks names’ column with the music database called *discogs* [6] to find their release date. After printing the release date column, the team noticed that the formats were either a full date formatted as (year/month/day) or just a year. In order to ensure consistency, the team truncated the release date column to just the release year for all tracks.

*3.2 Histograms:*

A histogram for popularity score across the dataset was created to visualise the distribution of popularity scores. The scores are approximately normally distributed with a mean of approximately 60 with a slight left skew. This left skew is expected as the data was generated from top artists, therefore most tracks should have a high popularity score. The histogram of track duration seems to be normally distributed with a mean of approximately 3.3 minutes. The plot also exhibited a strong right skew.

*3.3 Violin plot:*

The team used a subset of the features to generate violin plots, in order to visualise how they are distributed. The features included 'danceability', 'energy', 'acousticness', 'speechiness', 'instrumentalness', 'liveness', 'valence', 'tempo', and 'loudness'. Based on the visualisation, it seems that certain song metric features such as valence, energy, danceability and acousticness capture a greater variability of values, as seen in Figure 1(a). Other features such as instrumentalness and speechness capture less variability as seen in Figure 1(b). Therefore, these features could be important when explaining the variability of the data (and will be considered during model creation).

|  |  |
| --- | --- |

*Figure 1: Violin Plots to visualize distribution of (a) Valence and (b) Instrumentalness.*

Furthermore, from the violin and histogram plots, it is clear that many of the features are either non-normally distributed or skewed. This would impact the types of models we can use on our data. Specifically, models that assume normally distributed, (ideally) non-skewed variables such as linear models (i.e. Linear Regression) can not be used. Additionally, highly skewed variables can impact distance calculations, therefore before using ML models that use distance metrics, data should be log transformed inorder to account for the skewed data.

*Time-series:*  
To understand how ‘release\_date’ impacts other features, the team plotted the time series of some of the features. From the time series plots, it seems that track release date can capture a significant amount of variability in our data, seen by the fact that decade seems to be correlated with danceability and acousticness (see Figure 2(a) and (b)). Therefore, track release date may be an important feature to include in the final model.

|  |  |
| --- | --- |

*Figure 2: Time-Series Plot of key features, (a) danceability and (b) acousticness.*

*Feature interactions:*

In order to understand which features might be collinear, the team created a heatmap. From the correlation heatmap, it seems that loudness and energy are highly positively correlated (0.71) and acousticness and energy are highly negatively correlated (-0.62). Additionally, the team used a pairplot to visualise the relationships between features. From these pairplots, the correlation heatmap, and previously established domain knowledge, it seems that the variables: acousticness, energy and loudness are correlated with one another; and may introduce multicollinearity in our model. Therefore feature selection should be performed on these variables to remove such a possibility.

**4.0 Initial Model:**

For the initial model, the team decided to use 4000 of the 22081 songs to train on (this was done to reduce the computation time of model training, in order to produce results in a timely manner). The standardized data was then used to fit a k-means clustering algorithm with 10 clusters. The 3-D representations of the clusters were created using T-SNE, and can be seen in Figure 3(a).

A classic rock playlist dataset was then retrieved using spotify’s API. The playlist data was standardized, and tracks randomly sampled from the originally 22081 songs were added to the playlist data to create a new, balanced dataset (songs in the playlist have class = 1 and songs not in the playlist (sampled from the original 22081 songs), have class = 0). A decision tree was fit on this dataset.

Tracks in the playlist were aggregated together (using the mean/most common value for each feature) to create a single vector, representing a summary of the playlist features. The vector was then passed into the k-means model, and the songs in the cluster that the playlist vector belonged to were extracted. The 10 nearest songs to the playlist vector in this cluster were found using euclidean distance. These songs were passed into the fitted decision tree to produce final predictions of whether the songs should be a part of the playlist or not (hence whether they should be recommended or not). The corresponding prediction probabilities can be seen in Figure 3(b) where the 10 nearest songs outputted from k-nearest neighbours were test inputs to the decision tree. Using a threshold value of 0.5, the decision tree model believes that 7 of these 10 songs are in our playlist. This serves as a sanity check that our predictions were fairly relevant to the playlist content.

|  |  |
| --- | --- |

*Figure 3: (a) Visualizing clusters using T-SNE. (b) Prediction probabilities for decision tree recommendations*

**5.0 Next Steps:**

With the initial model complete, the team will move forward to build the final model, training on the full data set. The team plans on training an autoencoder to create song embeddings that capture the features of each song as a vector. Clustering song embeddings may provide better clustering performance and thus better song recommendations. The team also aims to research alternate clustering methods such as density based clustering, and methods to optimise clustering algorithms and performance. From analyzing feature interactions, the team will use feature selection to improve the classification model. The team will also research alternate classification models such as Random Forests and SVMs, and will research model optimization methods such as Regularization, Boosting/Bagging etc. to improve classification performance. The team also plans on using more playlists for validation to analyse the robustness of models on different style playlists (i.e. larger, smaller, different genres).

**References**

[1] A. M. A. Ahmed, “5 Challenges in Building Recommender Systems,” *Medium*, Nov. 04, 2023. https://awadrahman.medium.com/5-challenges-in-building-recommender-systems-77acbc0948cb

[2] N. Torabi, “The Inner Workings of Spotify’s AI-Powered Music Recommendations: How Spotify Shapes Your Playlist,” *Medium*, Aug. 28, 2023. https://neemz.medium.com/the-inner-workings-of-spotifys-ai-powered-music-recommendations-how-spotify-shapes-your-playlist-a10a9148ee8d

[3] A. Kharwal, “Content Based Filtering and Collaborative Filtering: Difference,” *The Clever Programmer*, Apr. 20, 2023. https://thecleverprogrammer.com/2023/04/20/content-based-filtering-and-collaborative-filtering-difference/#:~:text=Content%2Dbased%20filtering%20is%20like,with%20similar%20preferences%20have%20liked.

[4] Spotify, “Web API | Spotify for Developers,” *developer.spotify.com*. https://developer.spotify.com/documentation/web-api.

[5] “Spotify top artists by monthly listeners,” *Kaggle.com*, 2023. https://www.kaggle.com/datasets/meeratif/spotify-top-artists-by-monthly-listeners (accessed Nov. 07, 2024).

[6] Discogs - music database and Marketplace, https://www.discogs.com, Nov. 7, 2024.