AN MINI PROJECT REPORT

on

Splaylist: A Group-Based Music Recommendation System Using Spotify Listening Data

As a part of

Data Science (CE0630)

Submitted by

Student Name

Rishi Joshi (IU2341231551)

In fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY AND ENGINEERING

INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ

AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

APRIL 2025

Problem Statement

In today's social environment, music plays an essential role in setting the mood for social gatherings, parties, or road trips. However, selecting a playlist that represents the diverse music preferences of a group can be challenging. Our project, Splaylist – Group Music Taste, aims to solve this problem by automatically generating a collaborative music playlist based on the individual listening histories of users logged into our platform via their Spotify accounts.

The objective is to:

- **Aggregate Music Preferences:** Collect and merge each user's personal top track data (including track name, artist name, track ID, and popularity) along with additional attributes such as genres.
- **Data Integration:** Combine data from multiple users so that the resulting dataset meets the minimum requirement of 2000 rows. In cases where the live data from users is insufficient, synthetic data can be generated to reach this threshold.
- **Descriptive Analysis:** Generate descriptive statistics (measures of central tendency, dispersion, and distribution) to understand the characteristics of the combined dataset.
- **Inferential Analysis:** Apply hypothesis testing (e.g., t-tests or ANOVA) and regression analyses to infer relationships—such as whether music popularity or genre trends are statistically significant among different groups.
- **Clustering with Machine Learning:** Use unsupervised machine learning algorithms (for example, KMeans clustering) to group similar songs based on features like popularity and genre distribution. This clustering can help refine recommendations and ensure that the final playlist represents a balanced mix of musical styles.
- **Visualization & Reporting:** Create visualizations (bar charts, histograms, scatter plots, and cluster visualizations) and compile these along with the statistical summaries into a comprehensive PDF report.

By addressing these objectives, the project provides a scalable solution for generating a group playlist that reflects the collective music taste. It supports both exploratory data analysis and predictive modeling, ensuring that the final product can be justified statistically and machine learning techniques are applied effectively.

Dataset Description

1. **Spotify API Data:**

   When a user logs into our platform, we retrieve their personalized music data through Spotify's Web API. This data includes:

   - **Track Name:** The title of the song (obtained via /me/top/tracks or /me/tracks endpoints).
   - **Artist Name:** The name of the primary artist.
   - **Track ID:** A unique identifier for each track, which allows for linking to more detailed track information.
   - **Popularity:** A numeric measure between 0 and 100 that indicates the track's popularity.
   - **Genres:** Although not directly available at the track level, artist genres are fetched separately and associated with the track.

2. **Supplementary (Synthetic) Data:**

   In order to meet the minimum row requirement (i.e., 2000+ rows), if the total number of tracks from the logged-in users is insufficient, a synthetic dataset is generated. The synthetic dataset simulates similar data fields (track name, artist, genres, popularity) and is merged with the actual data collected through the API.

Key Attributes of the Dataset

- **Track_Name:** The name of the song.
- **Artist_Name:** The name of the artist.
- **Track_ID:** The unique identifier from Spotify; this is crucial for fetching further audio features when needed.
- **Genres:** The genres associated with the track or the artist, stored as a comma-separated string.
- **Popularity:** A numerical value between 0 and 100 that represents the track's popularity score.
- **User_ID and User_Name:** (Added from the individual user data during CSV creation) These fields help track which user's data contributed to the dataset.

Dataset Size

- **Row Count:** The dataset aims to compile data for at least 2000 songs. This is achieved by aggregating the music tastes of multiple users. If user data alone is not sufficient, synthetic data is appended.

- **Column Count:** At a minimum, the dataset contains columns for track name, artist name, track ID, genres, popularity, user ID, and user name.

How Does the Analysis Fulfill the Project Requirements?

- **Descriptive Analysis:**

  We calculate measures like the mean, median, and mode of track popularity. Distributions (histograms and boxplots) are used to identify trends and outliers within different genres. This helps summarize key characteristics of the music data.

- **Inferential Analysis:**

  We test hypotheses such as whether there is a significant difference in popularity between genres using t-tests. We also explore correlation patterns between attributes (for example, checking if a track's popularity is related to its genre or if there's any relation between user preferences and track popularity).

- **Machine Learning Component:**

  The project employs KMeans clustering to group songs based on numerical features (after converting genre information using one-hot encoding). This clustering is crucial to identify natural groupings in the data, which can be used to generate a balanced playlist that represents multiple musical tastes in a group setting.

- **Visualization and Reporting:**

  Several visualizations are generated (e.g., genre distribution, popularity distribution, clustering results). These are compiled into a PDF report using ReportLab, fulfilling the requirement for comprehensive data visualization and ensuring that insights are clearly communicated in the final report.

Code and Output

This project is implemented using Python, Flask (for web interface), Pandas, Scikit-learn, Seaborn, and Spotipy (Spotify API wrapper). Below are the major components of the codebase and their roles, along with a summary of output produced at each stage:

1. Web Application (app.py)

Purpose: Allows multiple users to log in with their Spotify account and aggregates their top tracks.

Main Features:

- **/login and /logout:** Authenticates with Spotify and resets session data.
- **/add_user:** Fetches top 50 tracks of a user and appends them to a CSV.
- **/create_playlist:** Generates a collaborative playlist based on all added users.

- Stores user info (user_id, user_name, top_tracks, genres, popularity) in user_music_data.csv.

Output:

- **A CSV file located at:** data/user_music_data.csv
- **Spotify playlist:** Created live on the Spotify account of the last logged-in user.

2. Synthetic Data Generator (generate_songs.py)

Purpose: Generates synthetic music data to ensure at least 2000 rows of music entries.

How it Works:

- Appends realistic-looking songs with random genres and popularity.
- Pulls user IDs and names from the real CSV for authenticity.

Output:

- Merged CSV file at: data/synthetic_user_music_data.csv
- Ensures dataset meets the minimum requirement for further analysis.

3. Machine Learning Model (model.py)

Purpose: Performs KMeans clustering based on genres and popularity.

Workflow:

- Encodes genres using MultiLabelBinarizer.
- Normalizes data using StandardScaler.
- Applies KMeans to group songs into clusters (e.g., mood-based or genre-based groups).
- Appends cluster column to dataset.

Output:

- Clustered data saved as: data/clustered_music_data.csv
- Used later for cluster visualizations.

4. Data Visualization and PDF Report (visualize.py)

Purpose: Performs statistical analysis and generates plots & a final report.

Includes:

- 📊 Descriptive Statistics: Mean, median, std dev of popularity.
- 📈 Inferential Analysis: T-test comparing popularity between genres.
- 📌 Visualizations:
  - Genre distribution (Top 10 genres)
  - Popularity histogram
  - Cluster scatterplot (based on popularity vs number of genres)
  - Genre vs. popularity boxplot
  - User contribution bar chart (songs added per user)
- 📄 PDF Report: Uses ReportLab to generate /visualization_report.pdf

Output Files:

- Graphs stored in: data/img/
- Final report: data/visualization_report.pdf

Technologies Used

Tool	**Purpose**
Flask	**Web app to manage routes**
Spotipy	**Access Spotify Web API**
Pandas	**Data wrangling and CSV export**
Scikit-learn	**Clustering (KMeans)**
Seaborn/Matplotlib	**Plotting visualizations**
ReportLab	**PDF report generation**