Visualization of Spotify User Data

Kale Dodson, Kyle Bomar, Daniel Moon, Jackson Tiemeyer, Brodie Kovach, and Reagan Sanz

Abstract—This project takes the listening history of users and creates easy-to-read graphs and tables to present the information. We will also be using the Spotify API to get current album artworks and artist profile pictures. All this data will be represented on a multi-page website.

I. Objective

The purpose of this project is to compile the information that Spotify gives users in an easy-to-read format. This will include a multi-paged web application that will allow users to input their own data to be displayed. This will be done through tables and graphs that can be altered to show differing information. We will also be using the Spotify API to have artist profile pictures and album art on display. This will allow for the pictures to be consistently updating.

II. Motivation

Spotify provides user with data through features like Wrapped and data exports, but this information is delivered in formats that are difficult to analyze and customize. Users receive static JSON files or limited annual screenshots that lack the flexibility needed for meaningful exploration and analysis of their data and listening habits.

This project aims to bridge the gap between data availability and usability by transforming this raw data into an accessible and interactive web application. Users can upload their personal data to generate customizable tables and graphs that display listening patterns across timeframes and metrics. Integration with the Spotify API ensures visual elements remain current while enhancing user engagement. This approach provides users with a comprehensive tool for analyzing their music preferences without requiring any technical expertise.

III. DATA

This section outlines the Spotify extended streaming history data utilized in the project, including its attributes, collection process, and application in generating visualizations. The data, comprising user listening records such as songs, podcasts, and videos, will be requested from Spotify for each group member and processed to create insightful visualizations of listening behaviors and preferences, supporting the project's goal of delivering an interactive dashboard or report.

A. Description

The project utilizes Spotify's extended streaming history data, which encompasses a comprehensive record of user interactions with the platform, including songs, podcasts, and videos consumed throughout the account's lifetime. The dataset includes attributes such as track or episode title, artist or podcast creator, album or show name, genre, timestamp of playback, duration listened, device used, and user account details. Additional metadata, such as playlist context, skipped tracks, and user interactions (e.g., likes or shares), may also be available depending on Spotify's data export structure. The data is provided in JSON format, which facilitates parsing and analysis for visualization purposes.

B. Collection

Each group member will request their extended streaming history directly from Spotify via the account's Privacy Settings on the Spotify platform. Spotify typically requires up to 30 days to compile and deliver the data, which is then provided as downloadable JSON files. The data will be collected for each group member's account, ensuring a diverse dataset that reflects varied listening habits.

C. Utilization and Visualization

The Spotify extended streaming history data, provided as JSON files, will be preprocessed using Python libraries (e.g., Pandas) to parse and aggregate native fields. Parsing and cleaning will yield Python scripts that handle raw exports, including time zone adjustments and metadata joins (e.g., track names, artists, genres) into a unified dataset. Feature engineering and aggregation will generate derived metrics stored in clean intermediate formats (e.g, Pandas DataFrames) for analysis. Visualization building will produce an interactive, mobile-friendly dashboard using Plotly or React/D3.js, featuring timeline views of listening evolution, heat maps of hours by day, and top-N bar charts for artists/tracks by msPlayed. This will be refined using informal user feedback sessions to enhance usability. Integration and testing will deliver an end-to-end pipeline connecting cleaned data to the visualization layer, validated with multiple user datasets to confirm patterns like routine-based listening or device-specific habits.

IV. Group Responsibilities

Group members will all be responsible for equal amounts of data collection and programming, however, individual roles are assigned as follows to ensure the project goes smoothly:

Leader: Kale Dodson.

- Responsibilities: Assigns tasks and ensures collaboration between group members. Ensures work is turned in.

Timekeeper: Jackson Tiemeyer.

- Responsibilities: Keeping track of deadlines and ensuring group stays on track to submit assignments on time.

Communicator: Brodie Kovach.

- Responsibilities: Ensures productive communication between members and facilitates communication with professor if questions/needs arise.

Recorder: Reagan Sanz.

- Responsibilities: Keeps track of documents and saves backups of files. Records any relevant meeting information if needed.

Proofreader: Kyle Bomar.

- Checks documents/code for spelling/grammar issues. Ensures submitted work is well formatted and code has relevant comments.

Quality Assurance: Daniel Moon.

- Ensures that references and datasets collected are accurate, unbiased, and relevant. Runs final testing on code before submission.

Beyond these roles, all members are expected to split responsibilities of front-end programming, back-end programming, and data collection as seen fit. These roles are less set-in-stone and will more vary depending on where the need arises.

V. Expected Outcomes

- A local-first, open-source toolkit (pipeline + web UI) with clear setup instructions.
- A normalized schema and documented transformations enabling reproducible analyses.
- Interactive dashboards that communicate habits, taste, and behaviors in plain language.
- An anonymized summary export to support sharing without exposing personal data.
- A short evaluation report from two formative usability tests.

VI. Timeline and Milestones

This section outlines the planned phases for the Spotify Listening Data Visualization project. The timeline defines each major stage: data acquisition, parsing and cleaning, feature engineering, visualization data, testing, and final delivery. This enables it so that each task is organized and progress can be tracked systematically over the course of the project.

A. Week 1-2 Data Acquisition & Understandings:

Download sample Spotify user data sets, review JSON/CSV formats, and document all available fields (timestamps, track IDs, genres, etc.). Work on IEEE Paper and finalize to submit.

B. Week 3-4 Parsing & Cleaning:

Implement Python scripts to parse the raw export files. Handle missing values, normalize timestamps to a common time zone, and join metadata (track names, artists, genres) from Spotify's API.

C. Week 5-6 Feature Engineering & Aggregation:

Derive listening metrics such as total play time per day, top artists, genre trends, and peak listening hours. Store results in a clean intermediate format (e.g., SQLite or Pandas DataFrames).

D. Week 7-8 Visualization Data:

Build prototypes of modern, mobile-friendly dashboards (React/D3 or Plotly). Include a timeline view, heat maps of listening hours, and top-N charts. Conduct informal user feedback sessions.

E. Week 9 Integration & Testing:

Connect the cleaned data pipeline to the visualization layer. Perform end-to-end testing with multiple users' data exports to ensure reliability.

F. Week 10 Documentation and Delivery:

Prepare user instructions and a final report. Provide installation scripts or a hosted demo so that Spotify users can upload their exports and instantly view dashboards.