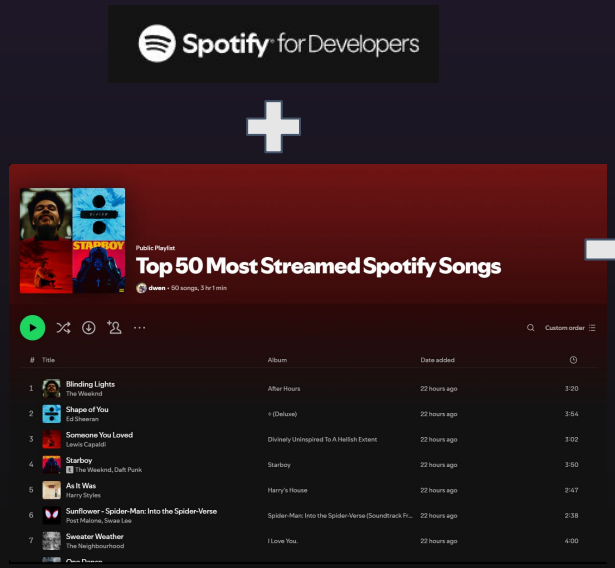
STATS 418 Final Project

Spotify Cover Art Quiz

Derek Wen

The Data

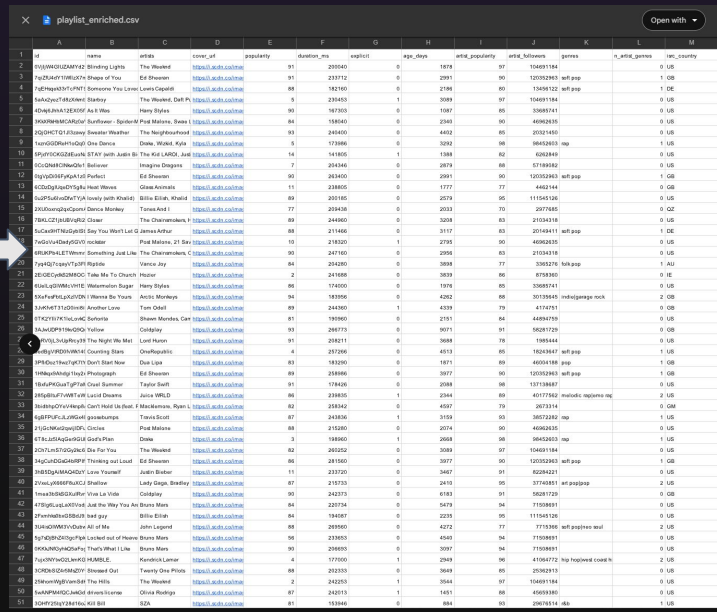
Call Spotify API on Spotify Playlist



Python Script To Fetch and Create Dataset

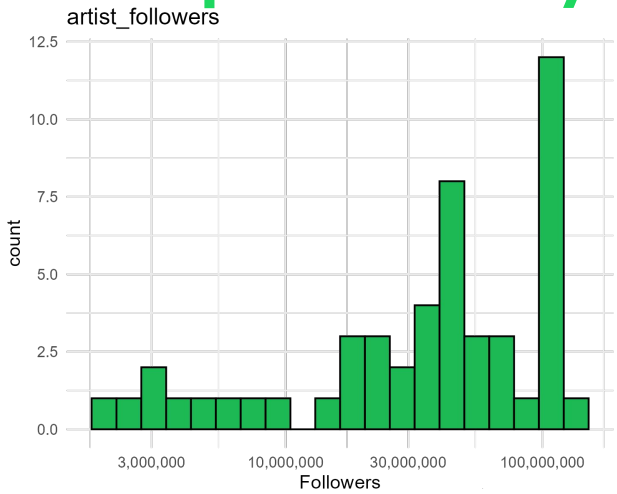
```
C:\Users\derek\top50-spotify-quiz>python src/fetch_top50.py
Fetched 50 tracks
→ playlist_enriched.csv written with 50 rows

C:\Users\derek\top50-spotify-quiz>
```

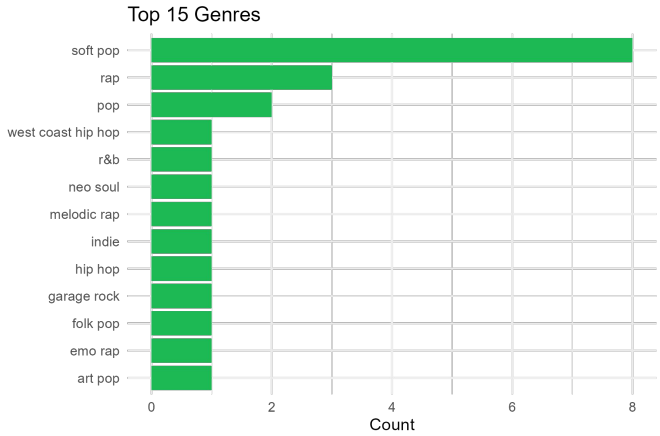
Dataset created as CSV

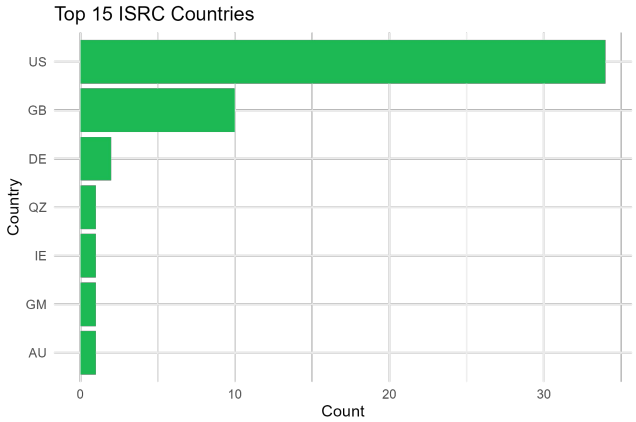


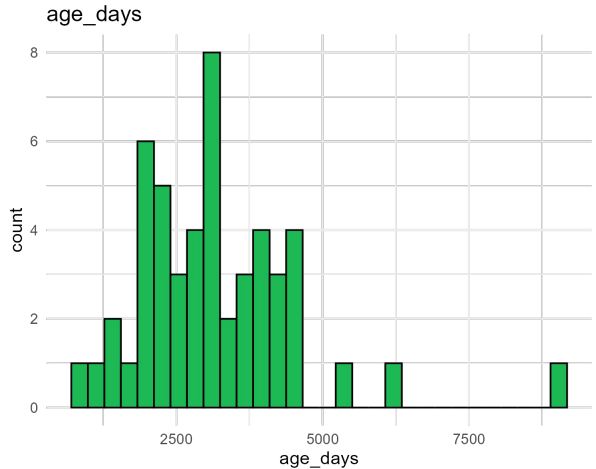
Data includes track id, song name, artist name, cover url, artist popularity, song duration (ms), explicit, song age (days), artist popularity score, genres, number of artist genre, and artist country.

Exploratory Data Analysis



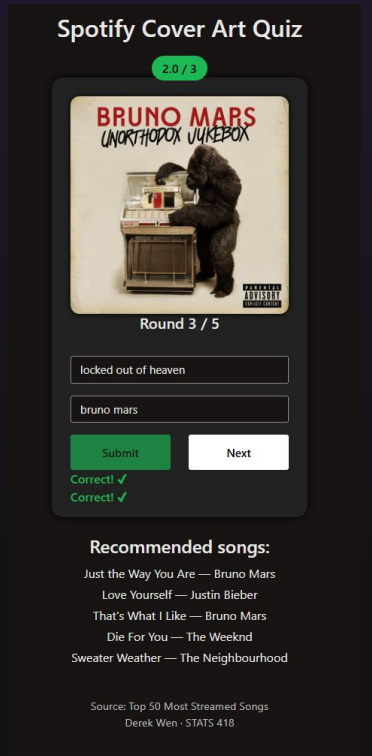






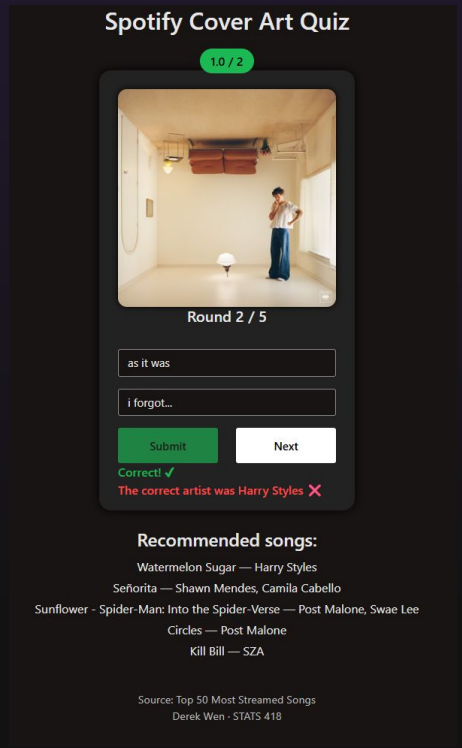
The Recommendation Model

- **k-Nearest Neighbors**
  - Lightweight, very fast, no training labels needed, simple and interpretable
  - Take song vector and compute Euclidean distance to all other songs and return closest ones
- **Features Used:**
  - popularity, duration_ms, explicit, age_days, artist_popularity, artist_followers, n_artist_genres
  - one-hot encode for genre and countries
- **Step by Step**
  - All numerical features scaled using StandardScaler to ensure equal influence in distance calculations
  - Each song is represented by a numeric feature vector
  - k-NN identifies tracks similar to the current song
  - Similarity measured using Euclidean distance after scaling



Deployment

- Frontend: R Shiny (hosted on shinyapps.io)
  - Interactive UI & displays Spotify cover art for user
- Backend: Flask API (hosted on Amazon EC2)
  - Provides recommendations using Docker container
- App
  - User has five rounds in a single session and has to guess song name and artist name based on cover art
  - Shows answers and shows whether or not the user guess the song and artist name correctly
    - + 0.5 for correctly guessing song name or artist name
  - After each round, recommends similar songs based on recommender model



Demo

https://dwen.shinyapps.io/spotify-cover-art-quiz/