
# Spotify UnWrapped: Find Your Next Song

Author: Kailin Xing

Northeastern University

EECE 2140. Computing Fundamentals for Engineers

Fall 2024

Date: Nov 21, 2024

**Abstract**

The Spotify UnWrapped project was initiated because of the one's boredom with existing music that one has been listening to. In hopes of discovering new music and finding the joy of listening to music, the application was started. The methodology consists of consists of 5 pages that are in the application, that deals with different parts of the task. One for the description of the project in the Home page; Kaggle dataset based explore page for some of the popular songs visualized using Plotly; user-given playlist-based song recommendation page using the Perceptron Algorithm; song-based recommendation page using SpotiPy; artist-related feature-based song recommendation. The use of frameworks is what makes the application special, delivering novel tunes to its users. Some of the key finds from this application include the success of the Perceptron model personalizing recommendations using machine-learning and API-driven data. The visualization as well as dashboard visuals is also another advantage of the application. Spotify UnWrapped achieved all of its initially set goals, offering its users the capability to find new songs they will fall in love with.

# Contents

# Chapter 1

# Introduction

## 1.1 Background

At times when one listens too much of the same songs over and over again, the feeling of novelty dissipates. The dissipation of tunes in one's mind lingers for a while until a new favorite song is found.

## 1.2 Problem Statement

Spotify UnWrapped, aimed to aid its users with new songs based on the world data, as well as their existing playlists, songs, and favorite artists, allowing for the never-ending joy of music. To achieve this, the project utilizes the Spotify API for its base of user data, as well as Kaggle based training set data and Streamlit for its interface.

## 1.3 Objectives

- Bringing back the joy of music with new songs at the tip of the user's touch
- Displaying content on world's most famous songs data
- Delivering analysis based on the user's playlist
- Allowing the user to get recommendation based on one song that they like
- Centering on one Artist and getting that artist's favorite song based on the user's preference of feature

## 1.4 Scope

Define the boundaries of the project, including what is and isn't covered. **Included**

- Top songs over the years
- Recommendation based on playlists
- Recommendation based on one song given
- Top songs based on artist

# Chapter 2

# Technical Approaches and Code UML

## 2.1 Development Environment

- Conda environment: SpotifyUnWrapped
- Python: 3.13.0
- Streamlit
- Pandas
- NumPy
- SpotiPy
- PlotLy
- urllib
- Scikit-learn

Most of the tools used revolved around the data that the program gets from Spotify API, these data are organized using Pandas when it comes to the Perceptron. The environment setup was made using Conda and it is exported in the project as requirements.txt as well as the environment.yml file for each of setup.

## 2.2 Data Collection and Preparation

For the Explore page, the data came from Kaggle[1] in order to display the graphs and visuals using Plotly. The data that came from kaggle was completely clean and did not need to be cleaned at all. The data in the Playlist

Recommendation, SpotiPy Recommendation, and Artist Analysis page came from the Spotify API[2]. The data that was retrieved from the Spotify API had to be organized by giving each songs a column for their features as well as other information that the API had. This way all of the data was available for display as well as perceptron training.

## 2.3 Implementation Details

Provide a step-by-step description of the coding process, algorithms implemented, and any specific methodologies followed.

1. Start with the home page of the project, made with Streamlit according to the docs[7].
    - Write the links that are related to the project
    - Write the description, features, tech stack, as well as the usage of the pages in Streamlit.
2. Kaggle data[1] was found and used for plotting using Plotly and Streamlit onto the Explore page of the application.
    - Write the dataframe that includes all the data onto the page
    - Use Plotly to draw the chart.
    - Use Plotly to draw the treemap
    - Use Plotly to display on streamlit the top 100 songs over time as the user drags it
3. The playlist recommendation used the Spotify API[2], and a Perceptron algorithm to find the most similar song in the 100 most listened playlist made by Spotify.
    - Connects to the Spotify API.
    - Implementing a Perceptron class that deals with the creation of a Perceptron object that is used to predict the most similar song in the top 100 playlist that is to the playlist that the user provided.
    - Use that class and clean the data then display it in front of the user.
4. The SpotiPy page uses the SpotiPy library and authenticating using Spotify API to get the most accurate recommendation used on a song that the user inputs.
    - Connects to the Spotify API.
    - Implements a cooldown period, so that the user can only get recommendations every 60 seconds.
    - Recommendations based on the user is saved to json locally for use if they give the same search.

5. The Artist Analysis page uses SpotiPy as well for finding the top songs made by the artist filtered by their feature according to the user's choice.
    - Allows for input from the user for their favorite artist.
    - Connects to the Spotify API and extracts the top songs based on the feature from the artist's Spotify page.
    - Displays a list of the top songs based on that feature as feature scores visible.

# Chapter 3

# Project Demonstration

## 3.1 Screenshots and Code Snippets

Include visual evidence of the project's outputs here. Add screenshots and code snippets as needed.

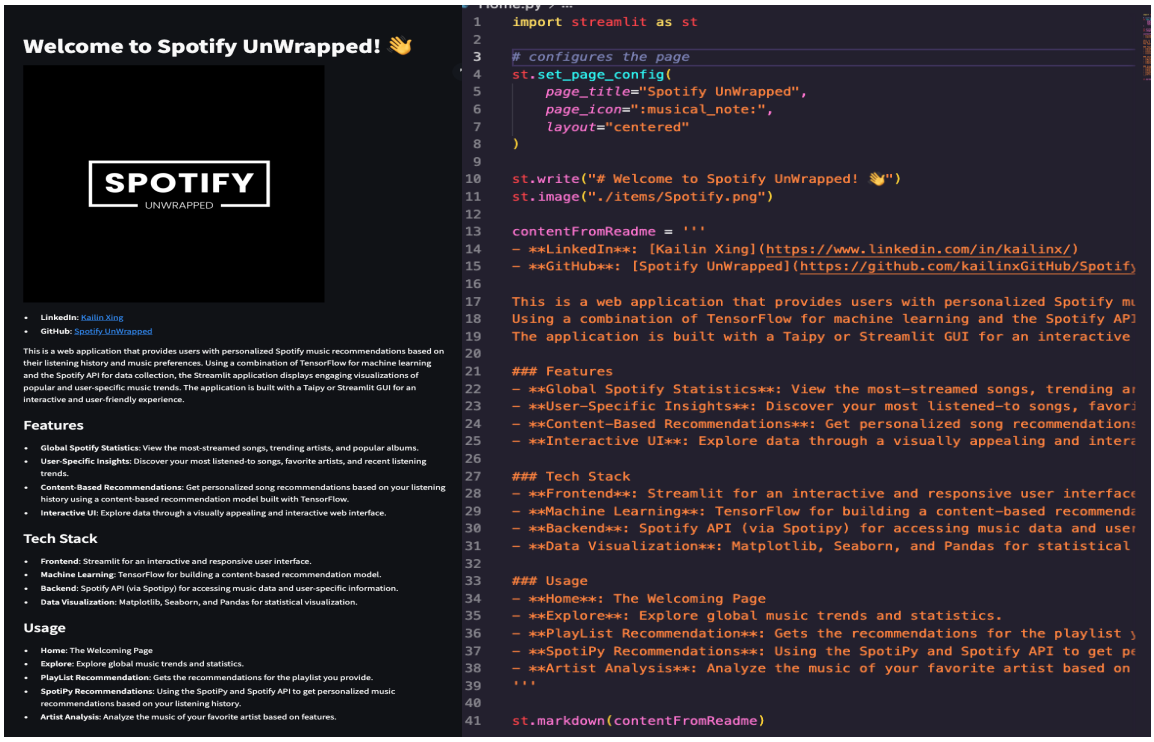


Figure 3.1: Home page

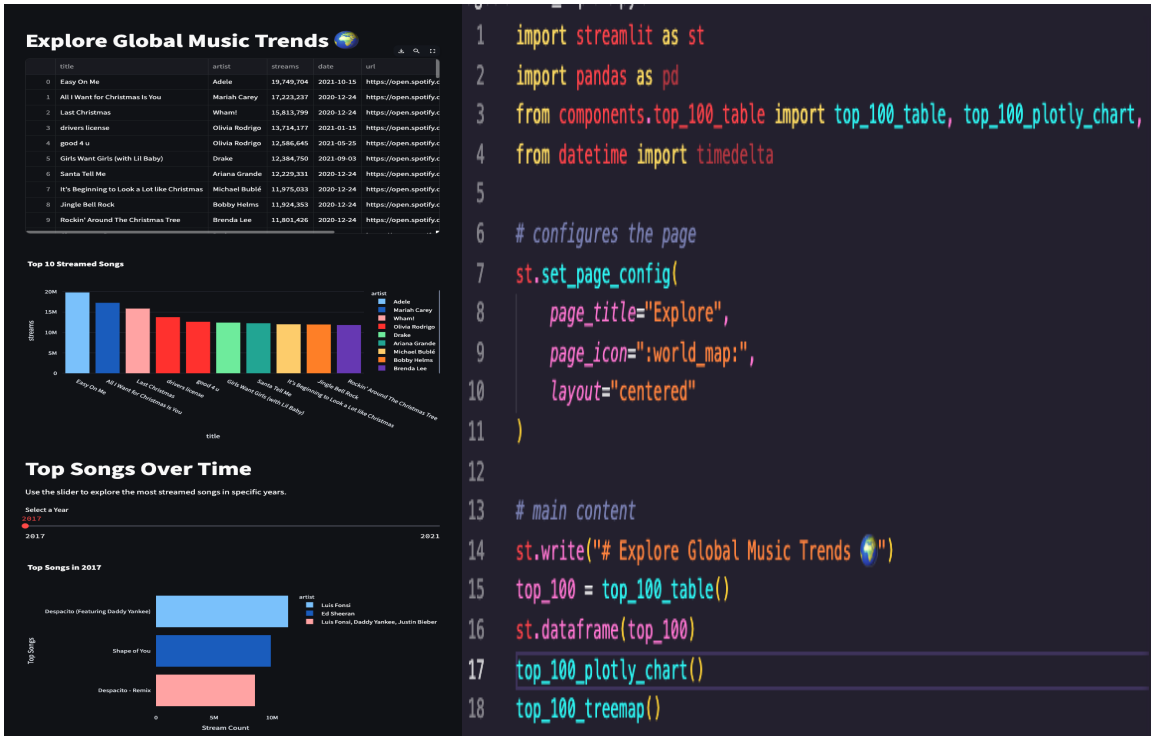


Figure 3.2: Explore page

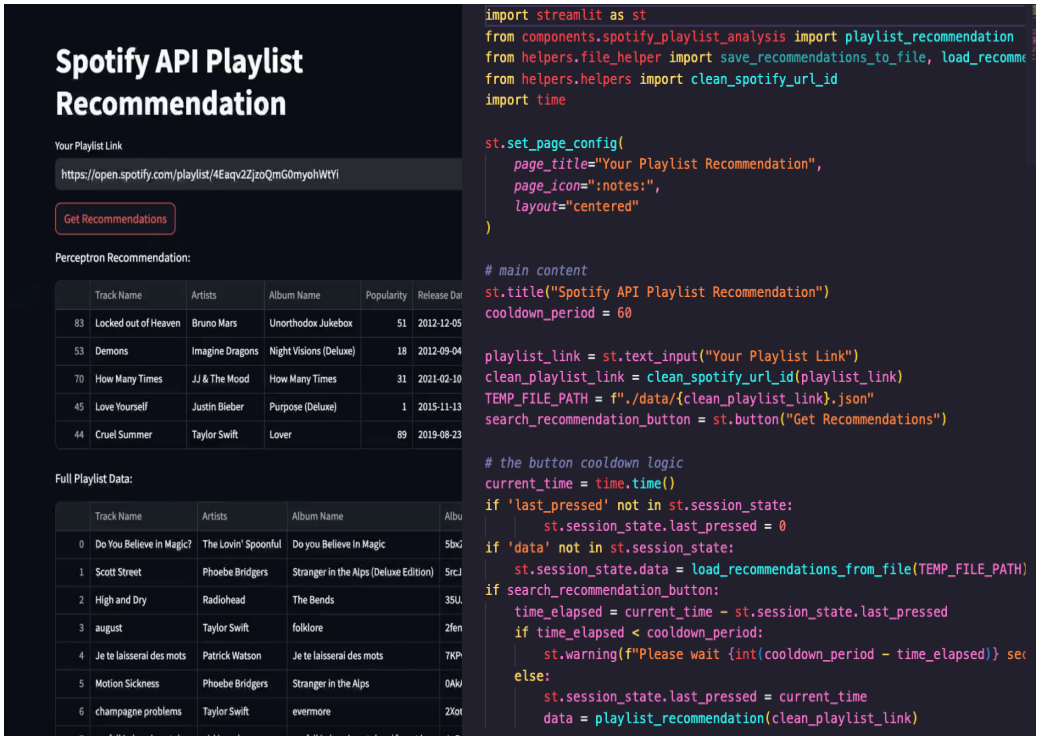


Figure 3.3: Recommendation Based on Playlist page

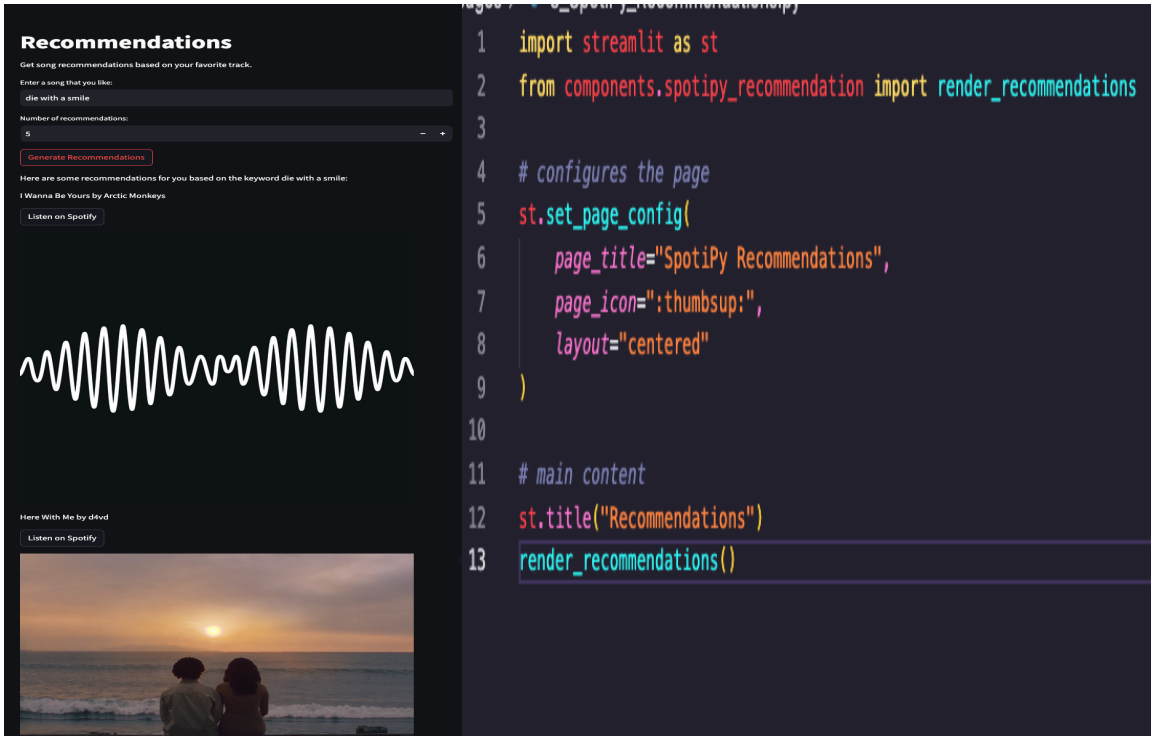


Figure 3.4: Recommendation with Song page.

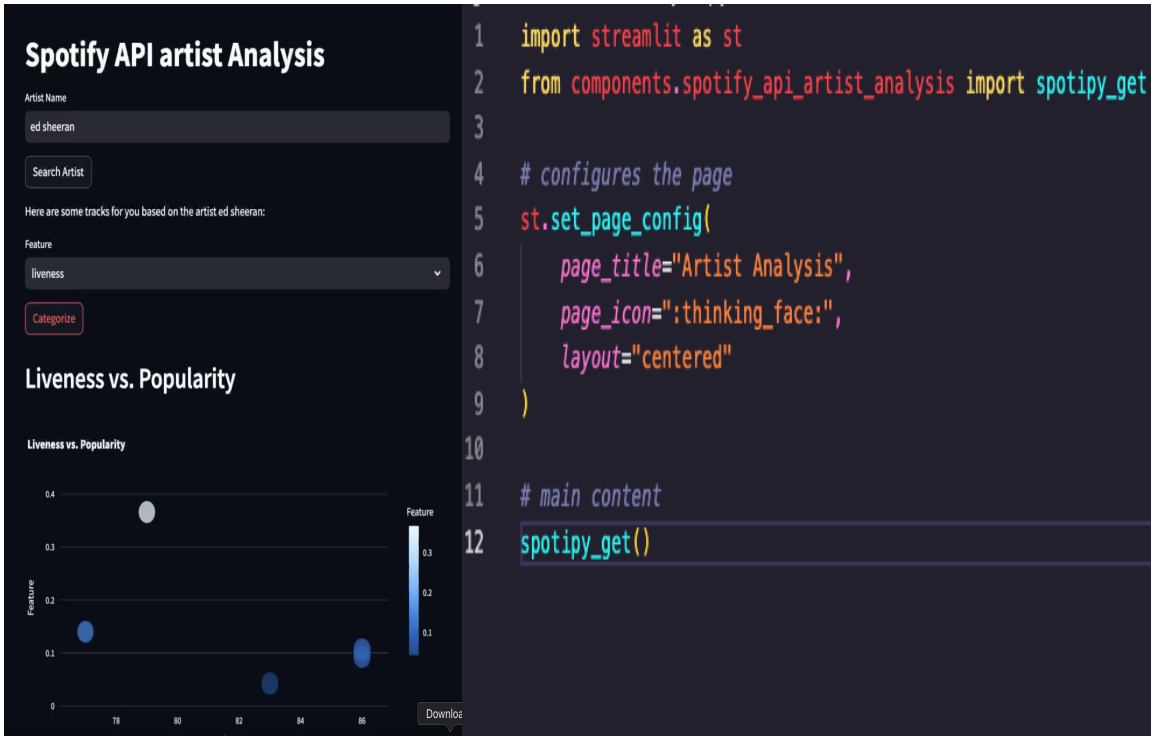


Figure 3.5: Artist Analysis page.

# Chapter 4

# Discussion and Future Work

The application shows a way of using the Spotify API to make magic happen, alongside Kaggle datasets, and machine learning algorithms. These implementations revolving around playlist-based recommendations, SpotiPy based song suggestiona dn artist specific song type suggestions all showcases how data was used effectively.

However some limitations were identified as long as future works that can be implemented:

- More advanced Models: Although KNN and SVD were tried to be implemented, their performances and depth was not ideal thus was ignored. In the future maybe more sophisticated machine learning algorithms such as GNN can be used.
- Recommendation system comparison: There isn't an ability for the user to compare multiple recommendation systems at the time, which would be a great addition to the existing features.
- Spotify API: Even though a lot could be gained from the API, the API's restriction on its requests makes it hard to call at times.

# Chapter 5

# Conclusion

Here's the link for a: Demo Video.
A lot of key findings were found alongside objectives were achieved. These are the things that emphasizes on the significance of the project:

- SpotifyAPI was used well and the object was achieved request limitations should be kept in mind the next time it is used.
- Perceptron algorithm classified well but was not the most accurate recommendation system out there.
- Displaying the world's most famous songs to the users achieved. Explore page's use of Plotly is a great skill to have learned and object to have achieved, more could done in Matplotlib for more customization.
- Analysis based on the user's playlist achieved, using the perceptron model that I made to obtain recommendations for the user based on their playlist.
- Recommendation based on one song achieved, a short one shot recommendation system that allows for a not-so-accurate but fast recommendation.
- Favorite artist songs recommendation based on style achieved.

# Bibliography


[1] Varun Saikanuri. *Top 100 Spotify Songs History*. Kaggle. Available at: https://www.kaggle.com/code/varunsaikanuri/spotify-data-visualization/input.

[2] Spotify Developers. *Spotify API*. Spotify. Available at: https://developer.spotify.com/.

[3] TensorFlow Team. *TensorFlow*. TensorFlow. Available at: https://www.tensorflow.org/.

[4] Shruti Somankar. *Get Playlist Data*. Medium. Available at: https://medium.com/@shruti.somankar/building-a-music-recommendation-system-using-spotify-api-and-python-f7418a21fa41.

[5] Beatrix LaVera. *For the SpotiPy Page*. GitConnected. Available at: https://levelup.gitconnected.com/how-to-build-a-music-recommendation-system-with-python-and-spotify-api-using-streamlit

[6] Leon Palafox. *For the SpotifyAPI Class*. Towards Data Science. Available at: https://towardsdatascience.com/how-to-utilize-spotifys-api-and-create-a-user-interface-in-streamlit-5d8820db95d5.

[7] Streamlit *Streamlit*. Streamlit documentation. Available at: https://docs.streamlit.io/.