Deven Panchal
Kevin Wang

DSCI 551 Final Report: Spotify Mood Recommender

Team Members:

Kevin Wang
Applied Data Science
Undergraduate Major: Statistics & Machine Learning
Skills: Python, SQL, R, data analysis, machine learning

Deven Panchal
Applied Data Science
Undergraduate Major: Economics
Skills: Data analysis, Python, web scraping

Background:
Our goal with this project was to create an application that would recommend songs on Spotify to users based on more finely tuned attributes than are usually available through Spotify. While Spotify usually creates song recommendations based on genre and artists, we wished to create a recommendation system that would match the finer details of each song such as mood and energy. The hope would be that such a system would allow users to find songs they might not usually find on their own but would nonetheless still enjoy.

Description:

Dataset:

For our dataset we originally intended on using the Spotify API to retrieve songs and song attributes. However, we discovered that the Spotify API does not provide direct access to songs on a song-by-song basis that would be compatible with our idea for a recommendation system. As such, we searched for an alternative data source and found a dataset containing over a million songs from Spotify. Additionally, this dataset contains the twenty-four song attributes that Spotify assigns to every song (i.e. danceability, energy, acousticness, etc.). While not the all-

encompassing library of songs we hoped to query, it was enough to provide the foundation for our recommendation system moving forward. Screenshots from this dataset are shown below.

tracks_features.csv (345.74 MB)

Detail | Compact | Column — 24 of 24

id	name	album	album_id	artists	artist_ids	track_num...
7lmeHLHBe4nmXzuXc0HDjk	Testify	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	1
1wsRitfRRtWyEap10q22o8	Guerrilla Radio	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	2
1hR0fIFK2qRG3f3RF70pb7	Calm Like a Bomb	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	3
2lbASgTSoDO7MTuLAXlTW0	Mic Check	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	4
1MQTmpYOZ6fcMQc56Hdo7T	Sleep Now In the Fire	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	5
2LXPNLSMAauNJfnC58lSqY	Born of a Broken Man	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	6
3moeHk8eIajvUEzVocXukf	Born As Ghosts	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	7
4llunZfVXv3NvUzXVB3VVL	Maria	The Battle Of Los Angeles	2eia0myWFgoHuttJytCxgX	['Rage Against The Machine']	['2d0hyoQ5ynDBnkvAbJKORj']	8

disc_numb...	explicit	danceability	energy	key	loudness	mode
1	False	0.47	0.978	7	-5.399	1
1	True	0.599	0.957000000000001	11	-5.763999999999999	1
1	False	0.315	0.97	7	-5.4239999999999995	1
1	True	0.44	0.967000000000001	11	-5.83	0
1	False	0.426	0.929	2	-6.729	1
1	False	0.298	0.848	2	-5.947	1
1	False	0.41700000000000004	0.976	9	-6.032	1
1	False	0.27699999999999997	0.873	11	-6.571000000000015	0
1	False	0.441	0.882	7	-7.362999999999999	1
1	False	0.44799999999999995	0.861	9	-6.12	1

Deven Panchal
Kevin Wang

Frontend (UI):

We utilized Django for an interactive User Interface. The app will allow the user to explore the database and song attributes. They will be able to not only see the raw data itself but also be able to enter a song name and find its attributes. They can also search for specific values of features such as danceability, energy, and liveness, in addition to the other attributes. In addition, the user will be displayed the metadata information on a separate tab, if they choose to interact with it. They will be given an option to upload the dataset as well. Lastly, the UI will have a “Song Recommender” tab, which will ask a user for 2 attributes and a song name, then input the top 10 closest songs. Lastly, the user will need the credentials for S3, so we are assuming the user is given the credentials to access S3, download the dataset, and then upload it to the UI using the “File Upload” feature.

Backend:

For the backend, the goal was to find a way to allow the user to input a song title and a pair of attributes they wish to search on and to return a list of songs that are the most similar based on those attributes. First we stored the dataset in Amazon S3 for processing. We then needed to create a machine learning model to form recommendations. We used Spark to handle this machine learning aspect of the recommendation system. Using Spark’s built-in machine learning library, we created a k-nearest neighbor model on the dataset that would allow us to find any given song’s nearest neighbors based on and two selected attributes. These neighbors would then be outputted as the recommendations for the user.

Conclusion:

Over the course of this project we learned many new and useful skills, primarily in the areas of machine learning and distributed systems. The practical use of a Spark application was very helpful as it allowed us to understand how Spark can be used in projects in the real world. Additionally, the process of creating an application from initial brainstorming and conception to final completion was a very educational experience and helped us gain a better understanding of the whole process.

We also had many obstacles along the way. The greatest challenge for us was understanding the web development process. Neither of us had extensive experience in web development, so it was a challenge to learn HTML, CSS, and Javascript in such a short amount of time in order to create a usable interface. Additionally, the Spotify API, which we initially had sought to use for data querying purposes, turned out to be much more limited in scope than we had anticipated, forcing us to abandon it altogether. One consequence of this is that our original plan to use the Spotify API for album art or audio samples had to be abandoned since the API was structured in such a way that made it impossible to query the Spotify database on a song-by-song basis.

Overall, the use of Spark and the machine learning aspect of the project also gave us great insight into practical uses of the topics we cover during lectures and this project was a very illuminating experience into how web applications go from idea to execution.