DS 501

Case Study 4 - Report

Songs Recommendation System

Team Noise Cancellation

Sree Likhith Dasari

Snehith Varma Datla

Ayush Shinde

Chinmaya Shivaswamy

Our Business Proposition

Oftentimes, one finds themselves wanting a soundtrack to a particular activity they are doing. Maybe you want a playlist for a nice cosy dinner on Christmas eve. Perhaps you want some tunes for when you're driving to and back from work. Whatever it is, having the 'right' music at the 'right' time can make all the difference. Song streaming platforms such as Spotify and Apple Music all have their own song recommendation algorithms. They make use of the massive amounts of data that users provide, in addition to various data and information from the songs themselves in order to recommend songs to their uers. There are however, many users who feel that there is a lack of playlists curated for specific 'moods' and 'activities'. They may feel that there are not enough options when it comes down to specific moods/activities. They may feel that the songs being recommended to them aren't exactly to their tastes.

Such users would want the perfect song recommendations for various activities and moods that they might do or feel on any given day, and that is the problem we have tried to address and come up with a solution for. On the one hand, Spotify recommends songs to the user based on the following- lyrical content and language, song features, and past listening habits. This does not leverage the fact that the user would want a particular type of music to listen to in a specific situation/scenario. Our recommendation system is mood/activity specific, enabling us to provide users with the perfect playlist at the perfect time. We view it as different from Spotify due to its different recommendation parameters. By curating the perfect playlist/recommending the right songs at the right time, we hypothesize that we would be able to increase user engagement with the app. This increases the utility of our product.

The general idea would be as follows -

The user provides our company with a list of songs that they like listening to for different specific activities.We allow for them to create an activity name, and then input a sample list of songs that they would like for that particular activity. We generate numerous playlists for them, all of which are activity specific. The songs for these playlists are drawn from a larger repository of songs. As the user uses and rates these playlists, we would generate more and more specific and accurate playlists for each activity. Over time, we envision that the user will be able to open the app and have available to them a plethora of individualized, activity specific playlists all tuned to their lifestyle and musical sensibilities.

Motivation and Reason for Interest in Topic

Being consumers of music ourselves, we as do most other people, have a vested interest in the type and specificity of music being recommended to us. This topic interested us because we have often found ourselves wanting music that is more curated our tastes. In addition, this project is something that is very relatable to us. Spotify is on everyone's phone's nowadays, as are various other music streaming platforms. The fact that our work and understanding could be applied to something that we all interact with on a daily basis fueled our interest and work on this case study.

In doing this project, we have gained a better understanding of how song recommendation systems work, and the different types of models that can be used for this purpose, and has given us a vision of how we would tackle this problem if we were in the industry.

The idea of these activity specific playlists that are generated on user given data intrigues us because we feel this type of feature is currently not available to users in the market. Could it be something that could break trends? Is it something that would add a degree of convenience and utility to our lives? These are all questions that arose while trying to complete this project.

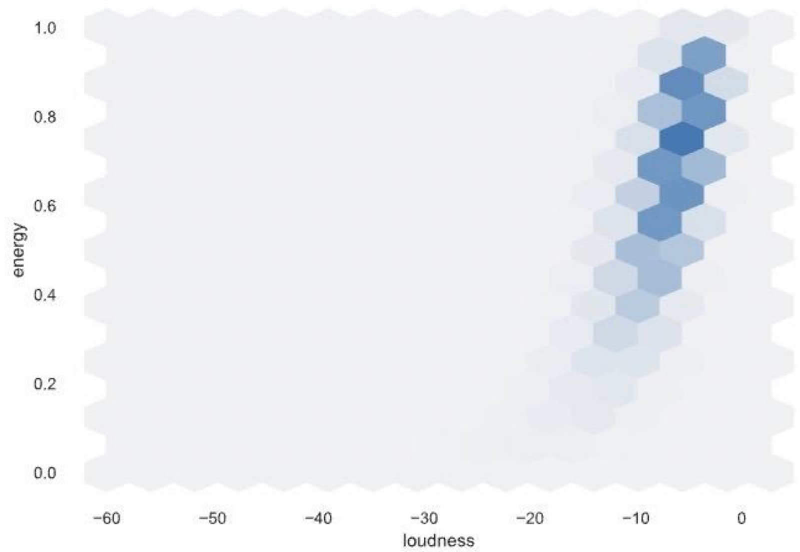
Analysis Details and Conjectures

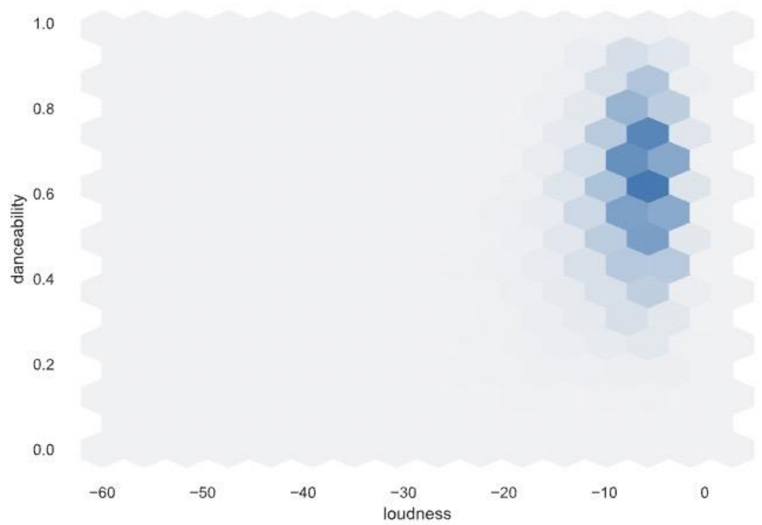
We first created a profile report for the dataset using pandas profiling. It gave us lot of information about the data that we were dealing with, including correlation heatmaps between all the variables, an insight into variables with high levels of correlation to each other and various pieces of information on individual features like the number of distinct values, the maximum value, the minimum value, the mean, the percentage of negative values and zeros, etc.

As we know, Spotify provides APIs to developers which allows us to access their data regarding users, artists and playlists.The data fetched using the Spotipy API describes various features and qualities such as the "acousticness', "energy", "liveliness" in addition to 13 other such features describing each song. This was greatly beneficial in the EDA stage of our case study. We were able to detect trends, and prove and disprove the various conjectures we came up with.

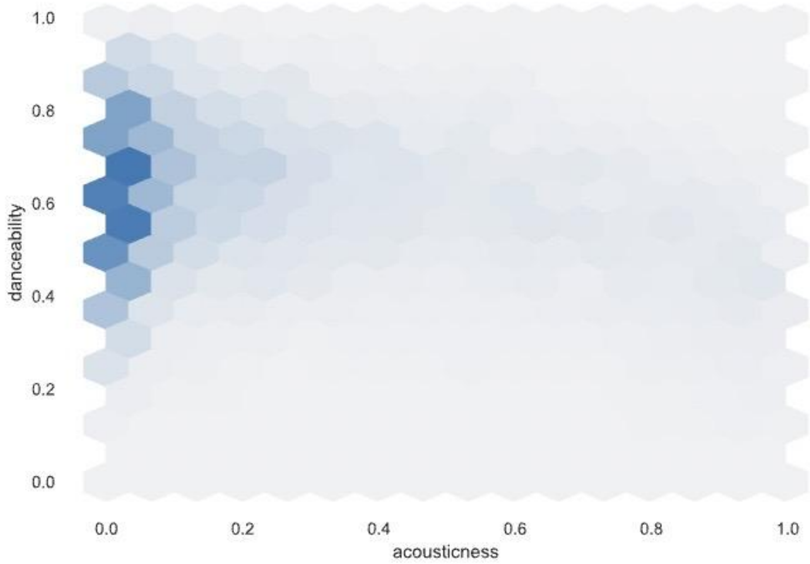
We came up with 5 conjectures that we wished to research and test.

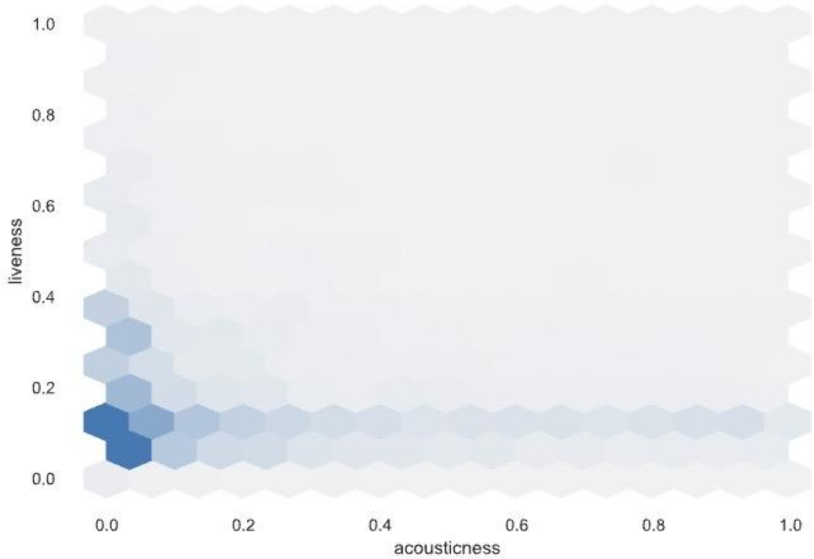
1. The first conjecture we came up with is that 'Loudness has better relation with danceability than Energy', where we plotted a couple of graphs through which we came to a conclusion that our first conjecture is true.



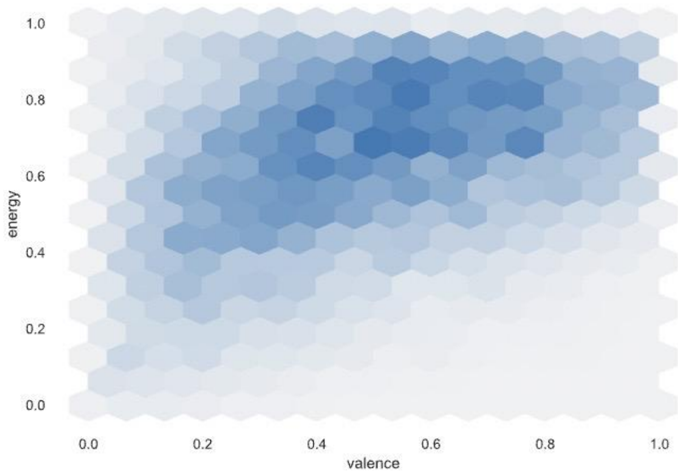


2. We made a similar conjecture that ‘Acousticness has a poorer relation with liveness than with danceability’ which is false based on the below graphs we plotted among those 3 variables.



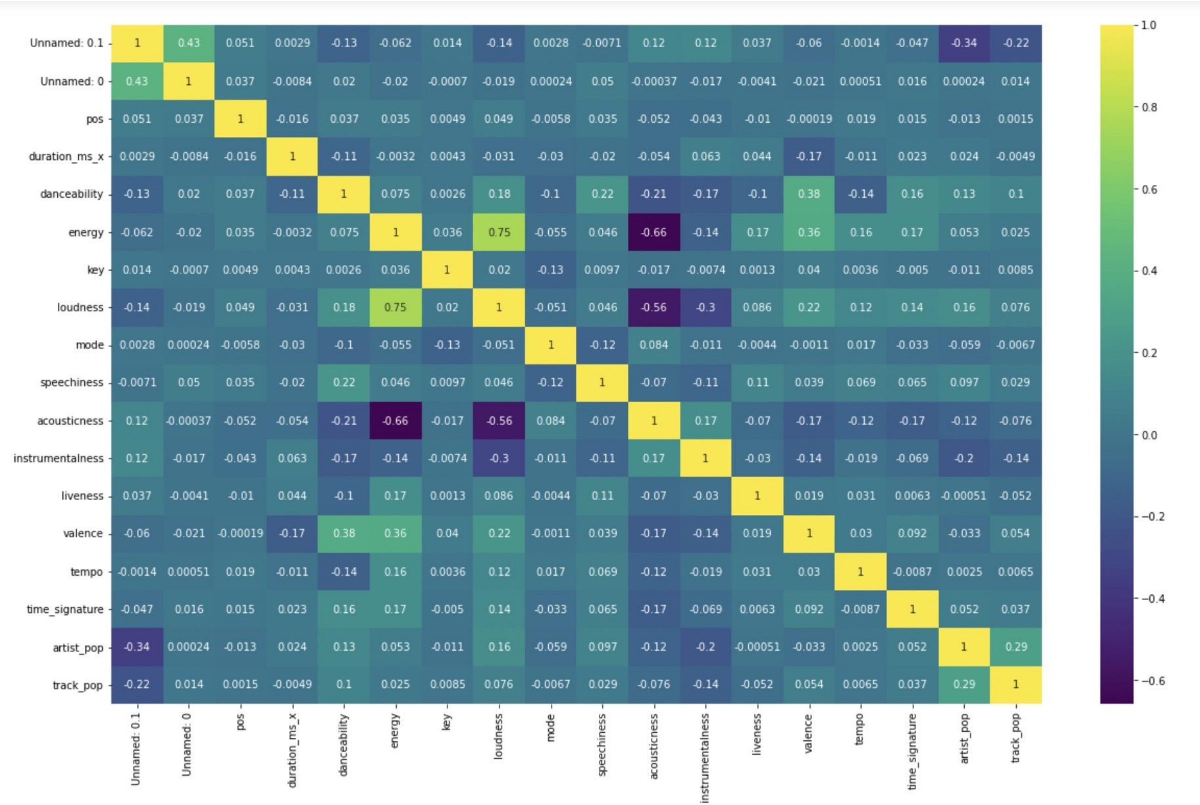


3. The third conjecture we made is ‘Valance doesn't add Energy to a song.’ where we made a scatter plot to see if there is any relation between valence and energy and surprisingly, we found our conjecture to be false.



4. The fourth conjecture is ‘Energy is the most important feature’ which is true based on the correlation values.

5. The last conjecture we made is ‘Acousticness is the least important feature’ which turned out to be false.



It is to be noted that we did not base our business proposition/product directly off of this EDA of the Spotipy API-generated data. Instead, this analysis served to provide us with an intuition of the how the music was broken down and articulated for the data scientist. We saw that it was possible to recommend songs to users based off of a whole plethora of different features. This lead us to the idea of trying to train a model for a song recommendationn system that focused on an area we thought had lacking infrastructure and developement, that is, user preference based, activity-specific playlists.

This sort of research was necessary for us to connect the dots and realize the potential of what we were trying to do.

Connection Between the Model and the Business Proposition

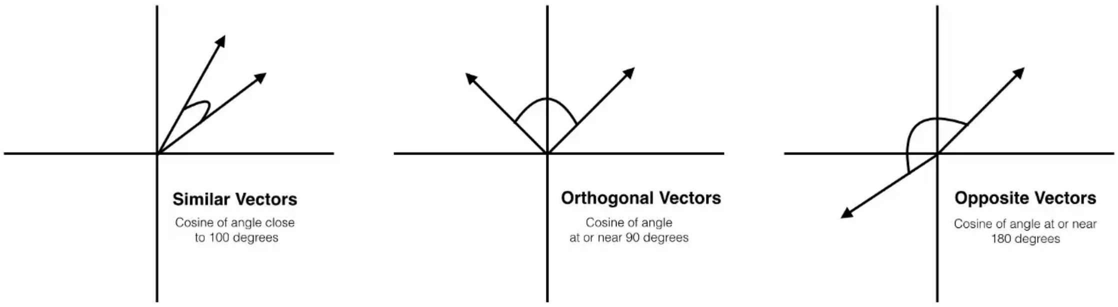
The goal behind this project was to design a model that recommends or predicts songs based on the preferences of a user from the playlist created by them. A cluster-based method may be used to predict the songs, but it is limited in its ability to incorporate additional information, such as a classification predictor. There are two popular types of models used when it comes to the development of a recommendation system - content-based filtering and collaborative filtering. A cluster-based approach is less flexible than these two forms of recommendation systems.

For this project, we have used the Content-Based Recommendation System. In this, the attributes of each item are used to identify comparable items. We may propose an item based on how similar it is to all other items in the dataset by giving each item a score that represents how similar it is to all other items. In our case, we used characteristics corresponding to each song (like acousticness, loudness etc.)

We now implement the content-based filtering algorithm after extracting all the data from playlists. The algorithm was implemented in two phases.

- Playlist Summarization:
    - This stage involves compiling a playlist of songs into a vector that can be compared to every other song in the dataset to identify commonalities.
    - Then, using the dataset we already created in the previous step, we identify the attributes of those songs.
    - As a result, it's critical that our dataset have a large number of songs to reduce the likelihood that the playlist at this stage will contain no matching songs.

    - Finally, we add all the feature values of each song in the playlist together as a summarization vector.
- Similarity and Recommendation:
    - We can determine the degree to which each song in the database and the playlist are comparable by getting the playlist summary vector and the non-playlist songs.
    - Cosine similarity was used as the similarity metric - a mathematical value that measures the similarities between vectors.



Mathematical representation is as follows:

$$\text{Cosine Sim}(A,B) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

How does your analysis support your business proposition? (please include figures or tables in the report, but no source code)

Though we did exploratory data analysis to find the relation between variables, there was not much to analyze and come up with a business solution. So, we started learning how other music streaming platforms are recommending songs. We got to know that most of them are recommending songs based on language, user pattern behaviour and their recent played songs. However, we observed something which is lacking in those recommendation systems. None of them had an activity based recommendation system. So, we decide to provide a recommendation system which suggests songs/playlist by taking user’s playlist as input. So for example, if a user is going on a trip and wants to add more songs to this travel playlists based on their interest, they can upload their playlist into the model. This model will display few songs which are very closely associated with different songs that he or she uploaded. This way, the user can get more similar songs of their taste with minimum effort and maximize their time spent on songs. Here, the distinction is that the power is in the hands of the user. They get to define the activity that they would like a playlist for, followed by giving the model a sample set of songs that they deem appropriate for such a playlist. By doing so, we hypothesized that we could provide users with highly relevant and compatible playlists, something we believe is not being done to a high level of quality in the current market/industry.

How did your team work together as a group from ideation to implementation? Write on one page.

We began our project by brainstorming ideas for a topic suitable, one which was parallel to our needs and requirements, and something that we could relate to and get behind. After a lot of deliberation, we decided to look into the area of song recommendation systems. It matched our interests and a consumer-centric business issue that we aimed to solve. We carried out some data analysis on the data that we were able to collect using the Spotipy API. This gave us an insight and intuition for some trends in music. Various features pertaining to the songs were explored and discussed amongst the group. As the idea evolved and the group carried out some brainstorming sessions, we realized that there was a business application in the vein of the topic we'd settled on.

Our idea was simple. To draw from the various song recommendation models already being used on the various song streaming platforms available online but to address an area we felt hadn't really been looked into. Giving the user the power to define activities that they would like music for, provide data on what sort of music fits their idea of a good playlist for that particular activity, and then enjoy the benefits of a customized playlist for activities that they enjoyed doing. Doing the data analysis on the API generated data allowed us to 'look behind the tinted window' and get a feel for how specific song recommendations could be. We realized that the power this sort of articulation of features could give developers.

As discussed above, we built a recommendation system that predicts songs according to the playlist of a user as per their liking and taste in music. We tested this model with various datasets. Some Spotify generated playlists, some of our own playlists, playlists given to us by our peers, all saw our model work on them. A master dataset was used in form of "training" dataset, which was trained through all the process addressed earlier. The output was computed on a "test" dataset, which gives the list of song recommendation to the user based on their playlist.

Below, we have shared an example output-

	artist_name	track_name
4	Missy Elliott	Lose Control (feat. Ciara & Fat Man Scoop)
21822	Marc Anthony	Aguanile
23545	Drake	Hold On, We're Going Home
23613	Kanye West	Black Skinhead
31758	Rihanna	Where Have You Been
32029	Drake	Signs
45222	Chris Brown	Questions
45225	Debbie Deb	Lookout Weekend
45226	Lisa Lisa & Cult Jam	Can You Feel the Beat
45227	Mase	What You Want (feat. Total)
45229	Faith Evans	Love Like This
45230	Johnny O.	Fantasy Girl
45231	Nelly Furtado	Maneater
45233	Kaskade	Beneath with Me (feat. Skylar Grey) - Kaskade'...

User Playlist

artist_name	track_name
Chris Brown	Run It!
T-Pain	F.B.G.M.
T-Pain	Drankin' Patna
Fabolous	You Be Killin Em
Drake	The Ride
T-Pain	Booty Wurk (One Cheek At a Time)
Fabolous	My Time
Drake	Take Care
Missy Elliott	Work It
Chris Brown	Back To Sleep
Missy Elliott	Get Ur Freak On
Chris Brown	Kiss Kiss
T-Pain	Dan Bilzerian
Fabolous	Can't Deny It - feat. Nate Dogg
Missy Elliott	Gossip Folks (feat. Ludacris)
Chris Brown	Beg For It
Chris Brown	Party

Recommended Songs