Project Summary – A Streaming Odyssey

Danish James

Project Design

The intent of this project was to design a model that could accurately predict what intrinsic factors of a song made it popular. Is there an inherent bias towards upbeat songs, or songs in the major key verses minor key? Perhaps there are other factors at play, like the ratio of vocals to instrumentals or how loud the song has been mixed. If there is a correlation, would it then be possible to design and optimize songs to maximize their popularity and outreach?

In order to determine this, I utilized Spotify as my primary source. Spotify, as the prominent music streaming service on the planet, also provides streaming data via spotifycharts.com. The website provides the top 200 streamed songs per region on a daily basis, as well as a weekly basis. For this particular case, I utilized the top 200 streamed songs globally, and took 365 days worth of data over the past year. In addition, I also utilized the Spotify API to pull various audio features about each song that could be utilized, as well as the follower counts for the artists. I initially wanted to implement more features that were intended to mitigate additional factors such as artist popularity and the age of a song, but ultimately I had to stick with the factors that were pulled from the API. Many of these factors were numerical floats on a scale from 0 to 1, but there were certain factors that were larger numbers such as follower counts which easily reached a few million for the most popular artists.

Modeling for this project was performed utilizing a Lasso regression model. Initially I tried out a simple OLS regression, however it was clear from the results already that there would be issues. Primarily, as I had been unable to formulate a good method for calculating the age of a song, I had tried to mitigate the issue by combining stream counts and removing song age entirely. This did mollify it to a certain extent, however it was clear that there were still issues with the model. I utilized a StandardScaler to try and bring all of my features in line with each other and minimize the impact of the different weights of each variable. Then I tested out both linear regression and polynomial regression, and found more success with a linear model. Thus I utilized KFolds with the Lasso Regression for my final model. I ended up with a resulting R^2 of 0.0239 which indicated that the model was not effective at explaining the resultant stream counts. While there were certain aspects that seemed to have some influence, such as the loudness and the tempo of the song, fundamentally it was not a significant enough influence to be meaningful.

Tools

- Python
    - Pandas, Numpy, Spotipy
    - Sklearn, Jupyter Notebook, matplotlib

- Microsoft Word, Microsoft Powerpoint

Data

The bulk of the dataset was obtained by scraping spotifycharts.com. The site did not appear to have any built-in blocks against scraping, however to be safe I chose to scrape with a cooldown period of 5 seconds after every 5 pages had been scraped. This did lead to a very long period of scraping as I was obtaining 365 pages worth of information, however it ended up being relatively short when all was said and done. This was then exported into a .csv file for storage. The file contained 73,000 rows worth of data, consisting of every song in the top 200 streams globally for the past year.

In addition, I also needed specific data features about the song and about the artist. I initially tried to pull this via the Spotify API, but it quickly became apparent that pulling 73,000 songs worth of data is an extremely time-consuming process and would take at least a few hours just for the initial data set. In order to cut down on the time spent pulling data, I instead created a separate song dataframe consisting only of the unique songs in the original dataframe, and a separate artist dataframe consisting only of the unique artists from the original dataframe. This cut down a search for 73,000 songs/artists, to a search for about 2,000 songs and 500 artists, cutting down the run time by several hours.

All the features utilized for this model can be found in Appendix I. Almost every feature was already prepared for usage as Spotify output them as floats between 0 to 1. There were a few exceptions, but they were handled by the StandardScaler. The scraped data was fairly easy to extract as the tables and divisions were very well-labeled. However there did seem to be an issue where some rows were not available, and those were dropped. I also noticed that some song titles did not come out correctly in the final dataframe, however as song titles were not utilized for the model it was safe to ignore them.

What I Would Do Differently Next Time

I felt as though I made a mistake in diving directly into scraping and API pulling, without thinking about optimizing my process first. If I had realized that I didn't need to pull all 73,000 rows on the first run through I would have saved many hours of unnecessary scraping. In addition, I would have been more thorough about analyzing my initial scraped dataframe. I did not catch several issues with the scraped data, such as missing information, until it caused an error later on down the line and forced me to redo most of the process to correct it.

In addition, I also think I did not spend as much time working on the feature engineering as I would have liked. While part of the issue was due to the problem stated above, I also spent too much time trying to make the difficult regression models work, instead of starting with the simple ones and working up to the more complex models. Had I conducted a different approach, I would have caught other issues with my approach much sooner. I think with a

refined approach, as well as a cleaner data scraping process, I might have been able to alleviate a lot of the problems that came up with my model.

Appendix I: Features

VARIABLE	TYPE	DESCRIPTION
DAILY STREAMS	Int	Amount of plays a song receives every day
SONG NAME	String	Song Identifier
ARTIST NAME	String	Artist Identifier
DURATION	Int	Total playtime of the song, converted to seconds
ARTIST POPULARITY	Int	A measure of artist popularity based on their follower count on Spotify, indicating how many people actively receive updates about this artist
MODALITY	Boolean	Determines whether the song is in Major or Minor key
ENERGY	Float	A value that tries to measure how intense the song is
TEMPO	Int	Measures how fast the song is in Beats Per Minute (BPM)
VALENCE	Float	Measure of the positivity of the song
INSTRUMENTALNESS	Float	A value for how much of the song involves vocals, and how much is more purely instrumental sounds
LOUDNESS	Float	Measures how loud the song is in decibels (dB)
SPEECHINESS	Float	Measures how much of the song is spoken word as opposed to singing