Capstone Final Report

Spotify is one of the top streaming platforms with contents such as podcasts, audiobooks, and, mainly, music. The platform requires users to have a premium subscription in order to access and download millions of songs on multiple devices, without Wi-Fi. The main source of revenue for Spotify is the user subscriptions, so it is very important to maintain and increase user engagement and longevity for the business to grow in profit. It would benefit Spotify to know about upcoming hit songs in order to market to its users before the songs are released. This can help maintain high user engagement as they are constantly aware of new and popular music present on the platform. By using characteristics of past popular songs, Spotify can predict popularity for future songs and use them for efficient marketing in order to maintain and increase user engagement.

To begin, we need to collect quality and sufficient data, which can be found in this CSV file provided by Kaggle: https://www.kaggle.com/datasets/zeesolver/spotfy . Additionally, we need to identify which parts of the data will be useful to the problem we are trying to solve. This can be done through data wrangling. Once this step is complete, we can continue to the next step of exploratory data analysis where we can identify relationships between characteristics and see which of these characteristics affects the popularity of a song. Finally, we can develop a machine learning model that predicts the popularity of future songs using a scale that will measure popularity from 1 to 10.

The deliverables in this project include a GitHub repository containing all the Jupyter notebooks I will use for each data analysis step of the project, a slide deck explaining the important findings from the data set, and a project report explaining the overall project.

First, I went through the data set and cleaned it up a bit. I identified any missing values, got rid of any unnecessary columns or rows, and developed histograms of each numerical column which showed the general distribution of the data values. A few important columns that I have identified in this step were “streams”, “in_spotify_playlists”, “bpm”, “key”, “mode”, “danceability”, “valence”, “energy”, “acousticness”, “instrumentalness”, “liveness”, and “speechiness”.

Second, I took the clean data set and did further analysis to see any reliable trends. I tried to identify any correlation between characteristic columns by using a pairplot and a heatmap. Here I was able to identify a slight correlation between “valence” and “danceability”. Using these characteristic columns, I then created training and testing data sets to create a linear regression model using these data sets. The regression result showed a R-squared value of 0.172 and an adjusted R-squared value of 0.167. This low R-squared value indicates that the independent variables in the regression model are not effectively explaining the variation in the dependent variable.

Third, I moved on to using the random rainforest classifier model and I also combined another dataframe into the current dataframe that I was using. This new data frame contained songs that were unpopular on spotify. I merged this new data frame into the data frame I was using originally and I added a column called “popularity”. The songs that were unpopular were given a value “unpopular” and the songs that were popular were given a value “popular”. This allowed for the data to fall into one of the two categories. After merging these two data frames together, I then proceeded to prepare the data to be used in the random rainforest classifier model. As a result, I was able to get an accuracy of 0.940 and f1-score of 0.922. The features I used included “bpm”, “energy”, and “danceability”.

For future research I would try to use data from other music platforms as well, such as Apple Music or Shazam. I was not able to find enough data regarding other platforms, so I wasn’t able to further my research in this direction. Additionally, I would test out different models using additional features and see if I could get a stronger result.