Project Summary: Predicting Artists Success

The music industry over the years has evolved along with consumer tastes. Both the predominant genres of popular music and music consumption patterns have evolved over time. Recording and listening technology has progressed significantly in recent years. As such, a musical artist's popularity in the modern age is most often quantified from streaming service data. This makes it more difficult for the music industry to determine success for individual artists in the modern age. Thus, the goal of this project was to point to certain indicators of popularity and success that might allow for better promotion of artists. Additionally, some of these indicators might make it easier for consumers to discover new artists using.

This big data project was conceived with Spotify data taken from a publicly available Kaggle dataset. Almost all of the individual CSV files contained the user id of each artist, their name, whether they are done with their career and their spotify id. To make sure all factors affecting an artist’s popularity were included, several files were analyzed, each containing different information. Each file had one or two columns in common that allowed for SQL querying. The two major metrics that were analyzed were Spotify’s popularity index, a value between 0 and 100 that compares artist popularity relative to the rest of Spotify, and their Spotify follower count. The key metrics were then compared to several factors affecting artist popularity. The factors included the artists’ total number of releases, the amount of time between first and last releases, and chart performance through the average peak ranks and average weeks on board for songs. Each comparison provided potential predictive tools for determining artist success in the streaming age.

The large dataset was pared down using Amazon Web Services (AWS). The CSV files of interest were uploaded into an AWS S3 bucket called artistrankings. Each file was stored in an individual folder within the bucket so that AWS would properly read them. The data from these files would be converted into a database using AWS Glue. In order for the process to properly occur, a database called artistrankings was created. Then, a developed AWS Glue crawler used the data from the artist rankings bucket to create tables for each CSV file that populated the artistrankings database. Each table was modified to ensure that the schema containing primary keys, foreign keys, and attributes would have column names and data types that allowed for proper queries to occur. Additionally, an Amazon Redshift cluster was created and the schemas were copied into the dev database. These schemas were populated with the tables from AWS Glue using another crawler. Once the tables were created, SQL queries on the AWS Athena and AWS Redshift platforms could run effectively.

On AWS Athena, four queries were run to create four tables that were used for data visualization. The tables from the artistrankings database that were queried were final, data\_w\_spotify, feature\_core\_100, and charts. The final table contained the primary key id for each artist, followed by the name of each artist, a boolean column called isdone, and their unique spotifyid. The isdone data identified whether or not an artist was still actively making music. The data\_w\_spotify table contained the same four columns as the final table, as well as the popularity metric and the number of followers. The feature\_core\_100 table contained the same columns as both final and data\_w\_spotify, but also contained each artist’s number of releases and their dates of first and last release, as well as additional statistics related to popularity, followers, and releases. Finally, the charts table could be queried with the previously mentioned tables because it contained the artist’s name, but also contained weekly chart information for each individual song. This included the date, rank, song title, rank from the previous week, the peak rank in that given week, and the number of weeks on board the charts. For both the data\_w\_spotify and feature\_core\_100 tables, extraneous columns not necessary for analysis were removed to prevent inadequate table creation on AWS Glue.

The first query called the popularity and followers columns from data\_w\_spotify and the id and name columns from final and ordered by ascending id. The second query began with a subquery which called the id and name columns from final, the popularity and followers columns from data\_w\_spotify, the song column from charts, and performed an aggregation of the minimum values for peak-rank. This aggregation was performed to find the highest chart positions for each song. The subquery was then queried to produce the average peak rank values for each artist in a new column called avg-peak-rank and ordered by ascending id. The third query called the id and name columns from final, the popularity and followers columns from data\_w\_spotify, and the num\_release column from feature\_core\_100. An aggregate column was also created in this query by finding the difference between the first and last release dates into a column called years\_active. The data was then ordered by ascending id. The final query called almost the same columns as the second query, except this subquery found the maximum values for the weeks-on-board column of charts and was then queried for the average value of the maximum weeks on board for each artist and ordered by ascending id. For all of these queries, filtering was performed so that all rows had popularity values greater than or equal to 50 and followers values greater than or equal to 1,000,000. The queries were run on AWS Athena at first, but the third query required an aggregate function that only worked on AWS Redshift. As a result, the queries were then repeated on AWS Redshift with the necessary modification being made to the third query to get desired results.

To see whether there is a trend or some sort of relationship between average peak chart ranks and popularity, a double stacked bar chart was created using Tableau. The pink zone represented the average popularity, and the blue zone represented the average peak ranks. Since the dataset was large, and not all datasets showed on the bar chart, artists with less than 20 million followers and a popularity rate below 75 percent were filtered out. The double stacked bar chart did not show a trend between popularity and peak rank. The average peak rank was arranged in a descending order, but the popularity rate followed a random pattern. In addition, a bar chart of average peak chart rankings was created. Calvin Harris came first with an average peak ranking of 14, followed by Bruno Mars. This is not surprising since these artists are currently popular.

Furthermore, a bar chart was created to show the average amount of weeks that an artist’s song spent on board. Imagine Dragons topped the list with 30 weeks, followed by Dua lipa with 27 weeks. Although weeks spent on board shows how popular an artist's songs are, it doesn’t necessarily indicate that they are popular or that the public likes them. To see whether average weeks on board and popularity for artists were related, a double stacked bar chart was created. The orange zone represents the average popularity of an artist and the pink zone represents the average weeks spent on board.While the popularity trend was mostly flat, the average weeks on board data were in a descending order. This shows that there is no relationship between the two because they are neither moving in the same direction nor in the opposite direction. Although it is not true in every case, an artist's popularity and their songs' popularity are not related according to our dataset.

Further comparison occurred between the artist's popularity and their followers. Although there is not an immensely strong relationship between the followers and popularity based on the scatter plot, there is a slight exponential-looking trend. The trend showed that followers and popularity have some relationship. Based on this, we can conclude that popularity does lead to more followers, although having more followers does not mean more people will like an artist’s song or that their popularity will rise.

Moreover, a double stacked bar chart was created to look at the relationship between the number of releases and an artist's popularity. The number of releases is set in a descending order and it has a pattern, but an artist’s popularity does not. Since they do not move in the same direction or in any pattern related to each other, the number of releases does not affect an artists’ popularity. Last, but not least, a bar chart was created to show the amount of years an artist has been releasing material with a color gradient that showed their popularity. Dark blue represented high popularity, while lighter blue represented low popularity. Since the years releasing values were displayed in a descending order, if there was a pattern between the two, the colors would go from dark to lighter or lighter to darker blue. But since this did not occur, the conclusion was that there was no relationship between the two factors.

Based on all the graphs and charts, the only factor that has a relationship with an artist's popularity is followers. There was a slight positive relation between followers and popularity. Other than that, the rest of the variables did not have any relationship with popularity. Therefore, only followers affected popularity in this dataset. Further analysis of streaming data with more indicators could find trends that are more compatible with popularity. However, this study proved inconclusive on the subject.