README File

About the Dataset

This dataset is a csv file from Kaggle that contains information about the most streamed Spotify songs on the music streaming platform Spotify for the year of 2023 (Elgiriyewithana, 2023). It has 953 entries (943 unique values), a large number that we would prefer for analytics purposes, and 24 columns that provide insights into each song's attributes, popularity, and presence on various music platforms. For each row entry, it contains information about the song with the following columns:

- track_name: Name of the song
- artist(s)_name: Name of the artist(s) of the song
- artist_count: Number of artists contributing to the song
- released_year: Year when the song was released
- released_month: Month when the song was released
- released_day: Day of the month when the song was released
- in_spotify_playlists: Number of Spotify playlists the song is included in (count)
- in_spotify_charts: Presence and rank of the song on Spotify charts
- streams: Total number of streams on Spotify
- in_apple_playlists: Number of Apple Music playlists the song is included in
- in_apple_charts: Presence and rank of the song on Apple Music charts
- in_deezer_playlists: Number of Deezer playlists the song is included in
- in_deezer_charts: Presence and rank of the song on Deezer charts
- in_shazam_charts: Presence and rank of the song on Shazam charts
- bpm: Beats per minute, a measure of song tempo
- key: Key of the song
- mode: Mode of the song (major or minor)
- danceability_%: Percentage indicating how suitable the song is for dancing
- valence_%: Positivity of the song's musical content
- energy_%: Perceived energy level of the song
- acousticness_%: Amount of acoustic sound in the song
- instrumentalness_%: Amount of instrumental content in the song
- liveness_%: Presence of live performance elements
- speechiness_%: Amount of spoken words in the song

Data Cleaning

In this section, we performed basic data cleaning tasks. We loaded the data from our Google Drive cloud, performed all of the necessary imports, investigated the dataset via head() and info() built-in functions, checked for NaN and 0 values for each column and replaced them accordingly on a case-by-case basis. We checked to see if any columns are detected as incorrect data types and made changes accordingly. We also performed some feature engineering by: 1) combining the month, week, day columns to create a new release date column, 2) creating a new column for the average monthly streams, and 3) converting artist count to a categorical value.

Descriptive Statistics

We first visualized the distribution of our total streams and average monthly streams data using histograms. To remove any outliers, we log transformed our total streams from our dataset into a normal distribution, then visualized its fixed distribution.

Then, we moved on to the song features columns in our data, such as numerical song features (bpm, danceability%, valuence%, energy%, etc.), artist count, song key, and mode (major or minor). Using histogram, we visualized the distribution of our numerical song features. We also visualized the relationship between the total streams and each numerical song feature using scatterplots. In addition, we graphed the bar plots of mean of total streams by artist count, and the mean of average monthly streams by artist count. Last but not least, we visualized box plots of the total streams by artist count, and the average monthly streams by artist count.

We were also curious to see the current statistics of our song key and mode attribute, so we created a bar plot of the song key's count and the song mode's count. Then, we displayed the box plots of the average monthly streams by song key, and the average monthly streams by song mode. We ended our analysis on the song key and mode attribute by creating a pivot table of the mean of our artistic song features (acousticness%, danceability%, etc.) by the song mode.

Next, we focused on the playlist attribute of the two most well-known music platforms: Spotify and Apple Music. Using scatterplots, we visualized the relationship between the total streams and the number of Spotify playlists, and the average monthly streams and the number of Apple Music playlists.

Finally, we ended our descriptive analysis with a box plot of the average monthly streams by release month and a histogram of the number of songs by release year.

Diagnostic Statistics

We initiated our diagnostic analysis with ANOVA. We wanted to see if there's any statistically significant differences in group means of total streams and average monthly streams by artist count. The results show significant evidence of difference in means of average monthly streams by artist count, so we conducted a Dunn test and Tukey's test for our post-hoc test to see which exact group pairings differ significantly.

Then, we conducted another ANOVA analysis to see if there's any statistically significant differences in group means of total streams and average monthly streams by release month. The results show significant evidence of difference in means of both total streams and average monthly streams by release month, so we conducted another Dunn test and Tukey's test to see which exact group pairings differ significantly.

Our last approach with ANOVA is to see if there's any statistically significant differences in group means of total streams and average monthly streams by song key. However, the results didn't show any significant evidence of difference in means.

Next, we visualized a correlation matrix of all of our variables. We also created a separate data frame of the correlation of our numerical variables (total streams, avg monthly streams, number of spotify playlists, number of apple playlists, and number of deezer playlists) with the rest of our variables.

Finally, we performed a Natural Language Processing using the nltk package for word frequency analysis of song titles.

Predictive Statistics

We utilized three different models for our predictive analysis: 1) multiple linear regression, 2) logistic regression, and 3) random forest regression. Before feeding our data to the models, we one-hot encoded our categorical variables to get the data ready.

For our multiple linear regression, we split our dataset into 70% training data and 30% testing data. After fitting a multi-linear regression model on the training data, we performed hyperparameter tuning through k-fold validation to determine model accuracy using R-squared value. We also performed feature selection by dropping statistically insignificant variables and fitting a new multi-linear regression model, which gave us an updated set of coefficients.

Next, we implemented logistic regression for classification, specifically focusing on predicting whether a song is a "top hit" based on the top 25% of streams on Spotify. A binary variable named "TopHits" was created as the target variable, with value as1 if the number of streams is above the top 25%, and 0 otherwise. Again, we split our dataset into 70% training data and 30% testing data and performed hyperparameter tuning through k-fold validation to determine model accuracy using accuracy score calculated using the confusion matrix.

Finally, To explore non-linear relationships, a random forest regression model is implemented. Feature engineering is performed, including label encoding and percentage conversion. The model is trained, validated, and tested, with an emphasis on evaluating the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. After, we evaluated the performance of the model based on MSE, MAE, and Rsquared, we created two graphs, where one compares the actual vs predicted values and the other compares residuals vs predicted streams.

Prescriptive Statistics

Although we are unable to draw a cause-and-effect conclusion from our observational dataset, we could still infer insights from our results. Based on our overall analysis and industry research, we closed off our analysis by proposing several recommendations for musicians looking to compose songs that could achieve high streams.

Requirements:

- This program requires Python environment.
- Required libraries: pandas, numpy, matplotlib, seaborn, statsmodels, scikit-learn

How to Run the Program:

- Ensure the required libraries are installed. Install by using pip install directly within Google Colab or use the corresponding command in your terminal or command prompt.
- Run the Python script containing the provided code. Make sure the codes

Sample Input and Output:

- The code takes in the dataset “spotify-2023.csv” containing music streaming data as the input.
- Output includes ANOVA tables, post-hoc test results, correlation matrices, machine learning regression model summaries, accuracy scores, and evaluation metrics for each model. Additionally, visualizations such as paired plots and scatter plots are generated for better understanding.