Data Science Project Report

1) Project Details and Goals

Kaggle: Most Streamed Spotify Songs 2024 Dataset

This project analyzes trends and patterns in music streaming using the "Most Streamed Spotify Songs 2024" dataset, aiming to identify factors contributing to high streaming counts, popular genres, and potential correlations between song attributes and their popularity. While the dataset explicitly focuses on Spotify, it also includes data on how songs are performed across different platforms, offering broader insights into the music industry. The analysis incorporates predictive modeling techniques, such as KNN, Linear Regression, and Decision Tree, to predict the all-time rank of future songs using regression or classification models. By comparing the performance of these models, the most accurate prediction method will be selected. Additionally, the project aims to provide actionable business insights for artists and producers to optimize their music's performance on streaming platforms.

2) Data Analysis

Dataset Description

The dataset contains comprehensive information about the most streamed songs on Spotify in 2024. Each row represents a unique song, capturing its attributes and streaming metrics. The dataset is suitable for supervised and unsupervised learning tasks.

What should we expect the data format to be?

- **Rows**: Each row corresponds to a unique song.
- **Columns**: Each column represents specific attributes related to the song or its streaming performance.
- **Data Types**:
  - Numeric data for features like total streams and duration.
  - Categorical data for attributes like genre or artist.

What are we predicting?

The primary goal is to explore patterns in streaming data and predict the **All Time Rank** based on input features through the following prediction models: KNN, Linear Regression, and Decision Tree. Note that, the only features that will be used for training and testing are those with integer values.

Columns/Key Features

- **Track Name**: Name of the song.
- **Album Name**: Name of the album the song belongs to.

- **Artist**: Name of the artist(s) of the song.
- **Release Date**: Date when the song was released.
- **ISRC**: International Standard Recording Code for the song.
- **Track Score**: Score assigned to the track based on various factors.
- **Spotify Streams**: Total number of streams on Spotify.
- **Spotify Playlist Count**: Number of Spotify playlists the song is included in.
- **Spotify Playlist Reach**: Reach of the song across Spotify playlists.
- **Spotify Popularity**: Popularity score of the song on Spotify.
- **YouTube Views**: Total views of the song's official video on YouTube.
- **YouTube Likes**: Total likes on the song's official video on YouTube.
- **TikTok Posts**: Number of TikTok posts featuring the song.
- **TikTok Likes**: Total likes on TikTok posts featuring the song.
- **TikTok Views**: Total views on TikTok posts featuring the song.
- **YouTube Playlist Reach**: Reach of the song across YouTube playlists.
- **Apple Music Playlist Count**: Number of Apple Music playlists the song is included in.
- **AirPlay Spins**: Number of times the song has been played on radio stations.
- **SiriusXM Spins**: Number of times the song has been played on SiriusXM.
- **Deezer Playlist Count**: Number of Deezer playlists the song is included in.
- **Deezer Playlist Reach**: Reach of the song across Deezer playlists.
- **Amazon Playlist Count**: Number of Amazon Music playlists the song is included in.
- **Pandora Streams**: Total number of streams on Pandora.
- **Pandora Track Stations**: Number of Pandora stations featuring the song.
- **Soundcloud Streams**: Total number of streams on Soundcloud.
- **Shazam Counts**: Total number of times the song has been Shazamed.
- **TIDAL Popularity**: Popularity score of the song on TIDAL.
- **Explicit Track**: Indicates whether the song contains explicit content.
- **All Time Rank**: Ranking of the song based on its all-time popularity.

3) Algorithms & Tools

Algorithms

- **Decision Tree Classifier**: A supervised learning algorithm used for classification. This algorithm was a good first choice as this classifier works well for both categorical and numerical data, which closely aligned with the dataset. This classifier outputted an accuracy of 99%
- **Logistic Regression**: A linear model for classification tasks. Also commonly used to predict probabilities by using the sigmoid function. This algorithm was highly effective resulting in an accuracy of 96%
- **K-Nearest Neighbors**: A basic but yet essential classification algorithm. Like Decision Tree, it can handle both numerical and categorical data. However, the algorithm didn’t yield the highest accuracy resulting in a score of around 69%.

Tools

- pandas
- numpy
- StandardScaler
- train_test_split
- DecisionTreeClassifier
- KNeighborsClassifier
- LogisticRegression
- mean_squared_error
- accuracy_score
- metrics
- matplotlib.pyplot as plt

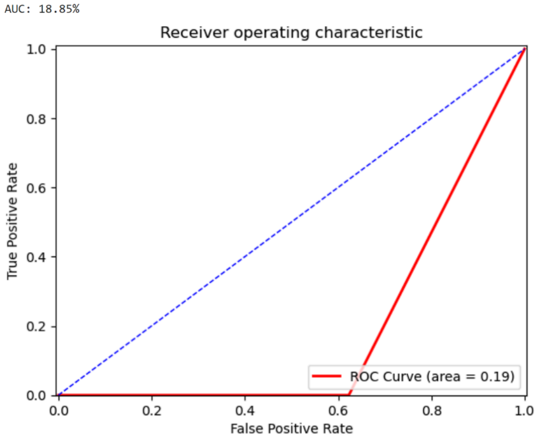
4) Final Results

Accuracies

- **DT Classifier (1st)**: 99.6 %
- **LR Classifier (2nd)**: 96.6 %
- **KNN Classifier (3rd)**: 68.8 %

AUC & ROC Graphs

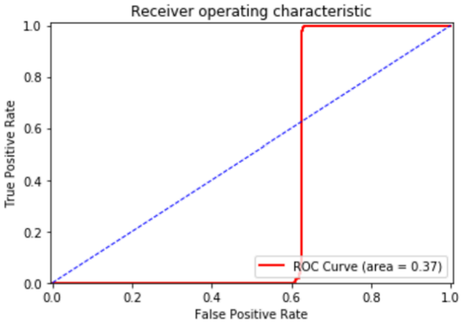
- **DT Classifier**



```
AUC: 18.85%
```

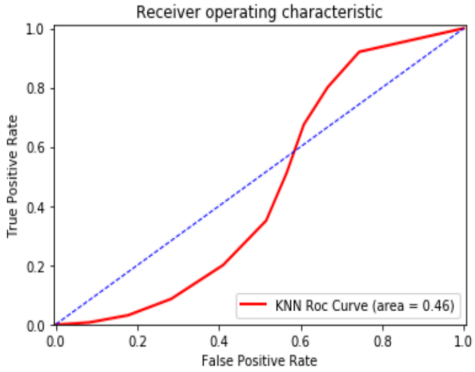
- **LR Classifier**

```
AUC: 37.385%
```



- **KNN Classifier**

```
AUC: 45.792%
```



5) Reflection

- In terms of accuracies, the Decision Tree Classifier had the highest accuracy of 99.6%, but the KNN Classifier had the highest AUC of 45.8% and ROC Area of 0.46!

6) Team Responsibility

Project Lead: Keyvan M. Kani

- **Adrian Flores Aquino**
  - KNN Classification
    - Sample K’s → (K=10)
- **Md Islam**
  - Linear Regression
- **Keyvan M. Kani**
  - Decision Tree Classifier

- **Project presentation slides**: The tasks to create the slides were split up between all three members