

Spotify Logo: Image Courtesy of Spotify

**Predicting Song Popularity using Spotify's Music Dataset:**

**A Machine Learning Approach**

**Introduction to Data Science**

**Denisha Saviela**

**April 10, 2023**

**Abstract:**

This report explores the prediction of Spotify songs' popularity based on objective features. The dataset consists of various features such as danceability, mood, and liveness generated by Spotify. The popularity index of each song, which is calculated relative to all other songs on the platform, has a significant financial impact on the creator's success. The goal of this project was to predict hit songs in future years based on the given dataset. Three machine learning models, namely linear regression, random forest, and decision tree, were used for this purpose. After preprocessing the dataset and cleaning the data, the models were trained and evaluated. The results show that the decision tree model provides the most accurate predictions, with the random forest model providing the closest predictions to the ground truth. However, the performance of all three models seems to be low. The report concludes that further improvements to the models are required to increase prediction accuracy.

**Overview:**

The Spotify music dataset contains numeric metrics generated by Spotify which measure the songs' danceability, mood, liveness, etc. The data also contains the songs' title and artist. From further research, Spotify uses machine learning to “help listeners discover content via recommendations and search, generate playlists, extract audio content-rich signals for cataloging and other content-based applications, understanding voice commands, serve ads, develop business metrics and optimization algorithms, create music with AI-assisted tools, and more” [1]. This piqued my interest to use this dataset and explore with the data more to embark on a rather interesting goal — predicting Spotify’s songs’ popularity score based on their objective features. While Spotify provides a popularity index for each and every score, which is calculated relative to all other songs in the platform. This rank has a tremendous financial impact on the creator’s success, since it affects the extent to which their songs would be recommended to other users and generate revenues. The goal of this project was to predict the hit songs in future years from multiple artists based on the dataset given. Further data will be created to increase prediction accuracy.

**Data Acquisition:**

This is a dataset consisting of features for tracks fetched using Spotify's Web API [2]. The Spotify dataset contain will contain the SpotifyFeatures.csv. The data collected for the set are all featured related to the presence of predicting hit songs. There are no limitations in sharing this data set because it was collected and distributed by a public institution.

The features in this dataset are listed below:

1. popularity
2. acousticness
3. danceability
4. duration\_ms
5. energy
6. instramentalness
7. liveness
8. loudness
9. speechiness
10. tempo
11. valence

This dataset can be used to make a classification model that predicts whether a track would be a “Hit” or not. Most of the features above will likely be the most useful in helping create a model to predict the next most popular song.

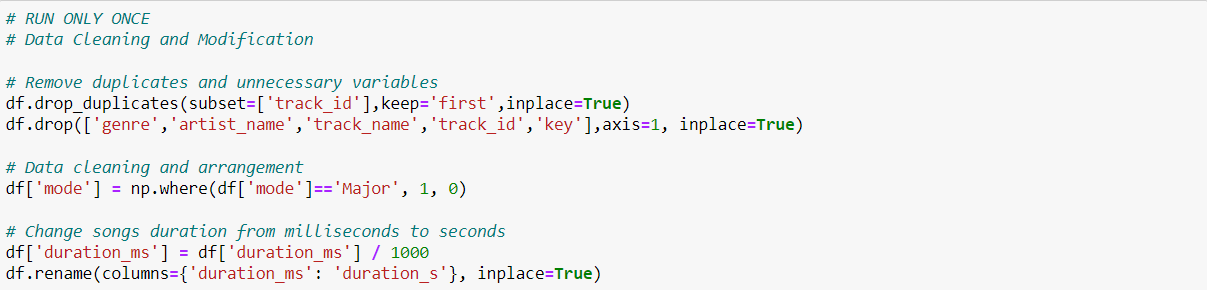
**Preprocessing:**

From Kaggle, the excel file was reloaded into a data frame and ran with some descriptive statistic. The database contains more than 232,000 songs with various levels of acousticness, danceability, energy, loudness and most importantly popularity.

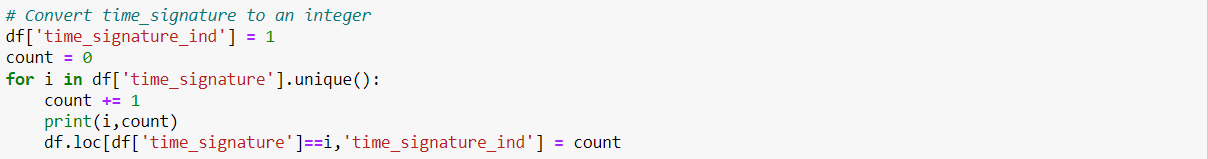
The descriptive statistics was helpful in detecting potential anomalies and missing data. I also encountered some modifications I was interested to conduct, like standardizing the songs duration from milliseconds to seconds and ignoring unnecessary variables. In addition, I checked whether there are multiple instances of some songs.

Data Cleaning and Modification:

Following the data cleaning process, the number of songs was reduced to 176,000. This modification hopefully reduced a potential bias which could result when the same song is appear both in the training and testing sets.



The removal of track\_id will keep the first duplicate and drop the rest. The second line of removing duplicates is set to true so the source data frame is changed and none is returned. The time\_signature column has a data type of object. This needed to be converted to an integer to have a successful data modeling.



After converting time\_signature, we can drop the original time\_signature and work with the new variable time\_signature\_ind.

**Model Selection:**

In order to model the data, there were two variables that needed to be declared — x which is a table containing all the songs' features, except popularity, and y which is a vector that contains the songs’ popularity score. Preparing the data for learning was the first step. Then, was to separate the database into training and testing sets, choosing to use 80% of the test for training and the rest for testing. The algorithm used for this is three shelf models that were selected to predict the songs’ popularity, that is linear regression, random forest, and decision tree [4].

The longest step was training the models. The estimated the models’ success rate in predicting the songs is shown below. Using the score() function calculates the coefficient of determination. A score that is closer to 1 means the regressor is more accurate.

Output:

Linear regression:

training score: 0.202

testing score: 0.199

Random forest:

training score: 0.906

testing score: 0.341

Decision tree:

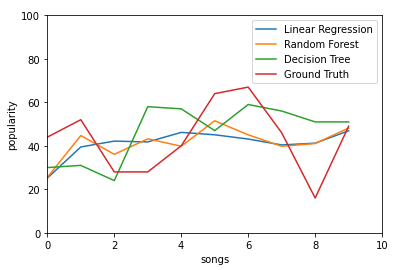
training score: 0.998

testing score: -0.350

From the results above, the three models’ performances seem to be low, especially the linear regression. According to the models’ scores, the decision tree model provides the most accurate predictions. Addressing the testing score for the decision tree model, the negative score implies that this model’s performance doesn’t follow the trend of the data, leading to a bad fit no matter what song is being tested [3].

**Results and Evaluation:**

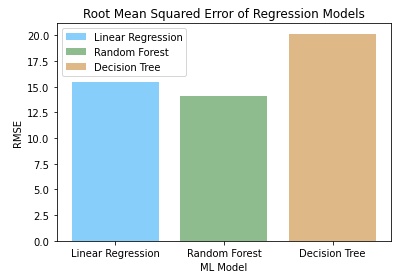
In this part a visualization of the model’s predictions was used to compare their accuracy using 10 songs taken from the testing data.



Regarding the above-mentioned graph, the red line represents the songs’ popularity score, while the other three lines represent the models’ prediction. Higher distance from the red line represents a lower accuracy. It can be observed that all three models didn’t perfectly fit the red line, and that the random forest provided the closest predictions to the ground truth.

In addition, it can also be observed the three models have different fluctuations. The linear regression seems to be more “conservative” and less prone to dramatic changes in predictions, while the decision tree predictions much more “sensitive” to changes hence the extreme spikes.

Prior to implementing any further improvements, to evaluate the current models’ prediction accuracy was done by calculating the Root Mean Squared Error (RMSE), which computes the difference between the model’s estimated prediction and the actual popularity score. A lower RMSE, represents model’s prediction accuracy for song’s popularity score.

****

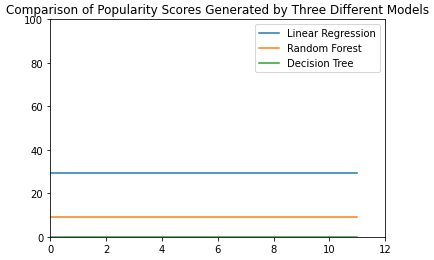
The graph below depicts the models’ accuracy before implementing any changes. The Random Forest model demonstrates the lowest prediction error.

A particularly helpful method that can be used to evaluate a model's prediction of a dataset by extracting important features from the dataset and transforming them into a format that can be easily understood by the model is called a feature extractor. By using a feature mixer, the model can be trained on a set of features that are more representative of the underlying patterns in the data, rather than the raw data itself. This can help to improve the accuracy of the model's predictions, as it can more effectively learn the relationships between the features and the target variable. Additionally, a feature mixer can be used to identify which features are most important in predicting the target variable. This can be useful in feature selection, where certain features can be excluded from the model to reduce complexity and improve performance.

In order to implement the feature extraction explanation, the features range values should be collected. Most features ranged between 0 to 1, while others, like tempo, ranged between 0 to 250 beats per minute (bpm). For every feature, the implementation of “faders” was implemented.

****

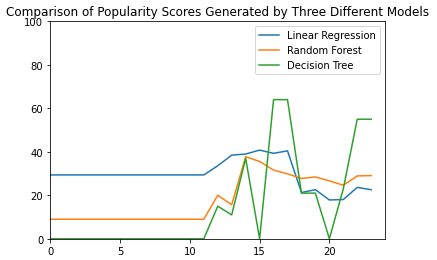
The features can be manipulated which will change the predication popularity score. The graph below is a representation of the current features above.

****

For the first set of features, all three models have predicted a different popularity score. Based on the given input, the linear regression gave a ranking of 35/100 for the first song, the random forest gave a ranking of 15/100, and the decision tree gave a ranking of 0/100.

Changing the feature extractor will give a different graph. This visualization also support multiple testing and comparison. As seen below, increasing acousticness, energy liveness, loudness, tempo, and valence and rerunning the script again results in a different graph and resulted in an increased popularity ranking.

****

****

The graph enables it to easily compare between the three models’ prediction before and after the changes made. Both the decision tree and the random forest models predicted the first changes had an increased popularity, while the linear regression prediction was lower.

**Conclusion:**

The Spotify music dataset, which contains objective features such as danceability, mood, liveness, and others that are generated by Spotify for each song helped predict the popularity score of Spotify's songs using these features. The popularity rank of a song has a significant financial impact on the creator's success and can influence the extent to which the song is recommended to users and generates revenue. The data acquisition process and preprocessing of the data, including data cleaning and modification, followed by model selection using linear regression, random forest, and decision tree models was done to predict song popularity. The decision tree model provides the most accurate predictions, but the scores for all three models are low. Additionally, visualization included of the model's predictions suggests that the random forest model provides the closest predictions to the ground truth. It was also highlighted that the need for further improvements to enhance the accuracy of the models should be done.

**References**

[1] Machine Learning – Spotify Research

<https://research.atspotify.com/machine-learning/#:~:text=It%20is%20used%20to%20help,music%20with%20AI%2Dassisted%20tools%2C>

[2] Web API References | Spotify for Developers

<https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features>

[3] Random Forest

<https://towardsdatascience.com/random-forest-in-python-24d0893d51c0>

[4] sklearn.dummy.DummyRegressor

<https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyRegressor.html#sklearn.dummy.DummyRegressor.score>