Predicting Spotify Songs’ Popularity Using Machine Learning Algorithms

Anh Nguyen, Hanna Song

December 16th, 2022

Abstract

Spotify is a digital music and podcast provider that has access to millions of songs and other content from all over the world. To internally classify songs, Spotify assigns each song 13 key audio attributes (can be numerical or categorical) and a popularity score (measured by the number of streams a song has). This study aims to look at the current top trending songs from Spotify database and analyze the correlation between their audio attributes and popularity scores, therefore predicting the upcoming hit songs from a dataset of random new releases. In order to reach an accurate prediction, we will be testing 4 different models on our datasets–Linear Regression, Decision Tree, Random Forest Classifiers, and K Nearest Neighbor.

1. Introduction

Spotify is the world’s biggest music streaming platform by number of subscribers. Users of the service simply need to register to have access to one of the largest collections of music in history, plus podcasts and other audio content. In Q1 2022, there are 422 million people who use Spotify once a month, of which 182 million are subscribers of the service.[1] Understanding this, we want to look at data from the streaming service itself to analyze the key characteristics of top trending songs to predict the next waves of hits. Our project aims to build a model to best predict future song success, using 4 mentioned machine learning methods.

2. Description of the Dataset

- **Data sets:** Both of the raw datasets were obtained through the Spotify API website. The first dataset contains the most trending 50 songs on Spotify at the moment, and the second dataset is newly released songs. Each song from both sets is assigned 13 different audio features and a popularity score. We scraped the descriptions of all key attributes from Spotify's API website[2].

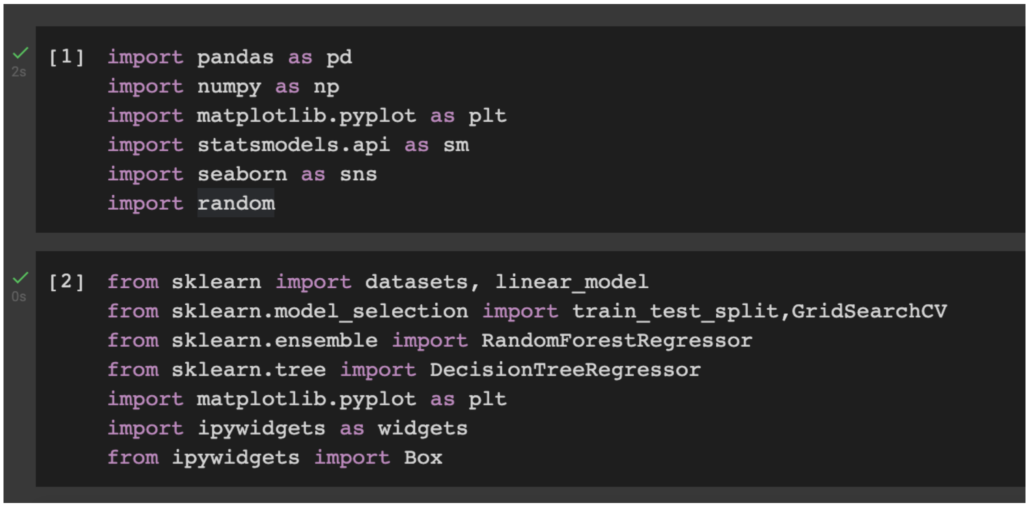
[1] Iqbal, M. (2022). Spotify Revenue and Usage Statistics (2022). <https://www.businessofapps.com/data/spotify-statistics/>

[2] (2022) Get Tracks' Audio Features. *Spotify for Developers.* <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features>

- title (object): Name of the track/song.
- artist (object): Name of the artist.
- top genre (object): Genre of the track/song.
- year (float): Year the song is released.
- bpm (float): tempo of the song, measured by beats per minute.
- nrgy (float): Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
- dnce (float): Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- dB: (float): The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
- live (float): Detects the presence of an audience in the recording (liveness). Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
- val (float): A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track (valence). Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
- dur (int): The duration of the track in seconds.
- acoust (float): A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- speech (float): Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
- pop (float): Overall popularity score (based on # of clicks) The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity of a track is a value between 0 and 100, with 100 being the most popular.

  The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past.

- **Data Preparation**: We used the following Python libraries and SKLearn algorithms to conduct this research:

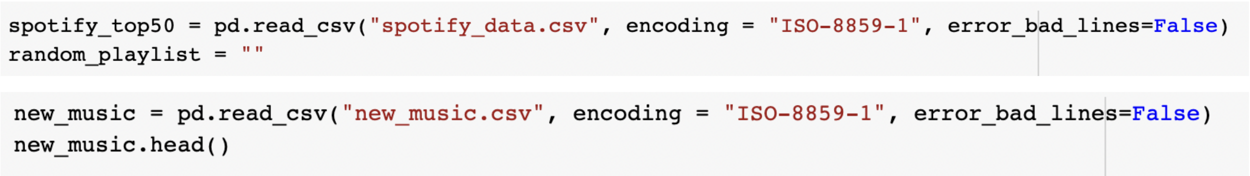


```
[1] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import statsmodels.api as sm
    import seaborn as sns
    import random
```

```
[2] from sklearn import datasets, linear_model
    from sklearn.model_selection import train_test_split,GridSearchCV
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.tree import DecisionTreeRegressor
    import matplotlib.pyplot as plt
    import ipywidgets as widgets
    from ipywidgets import Box
```

figure 1.

We loaded two csv files into spotify_top50 and new_music dataframes:



```
spotify_top50 = pd.read_csv("spotify_data.csv", encoding = "ISO-8859-1", error_bad_lines=False)
random_playlist = ""
```

```
new_music = pd.read_csv("new_music.csv", encoding = "ISO-8859-1", error_bad_lines=False)
new_music.head()
```

figure 2.

3. Exploratory Data Analysis (EDA)

Each row of the datasets represents a song with its key sound attributes and its popularity. After obtaining the 2 base datasets, we cleaned both sets up to drop all null values that could cause confusion to the model. Prior to building a prediction model, we performed exploratory data analysis on both the Top50 Spotify songs and the new music datasets. We focused on finding out the most popular artists, genres, and songs (with the highest popularity scores) and looking for a high correlation between relevant features like energy, danceability, acousticity, loudness, etc… with their corresponding popularity scores. As a result, we found out that the pairs of features that have the strongest correlations were energy and popularity scores with a correlation of 0.46. Unfortunately, some outstanding features also returned negative correlations with popularity such as liveness, speechiness, and valence, which was not something we expected. The data visualization of our EDA process could be found below:

- Top 5 most popular artists:

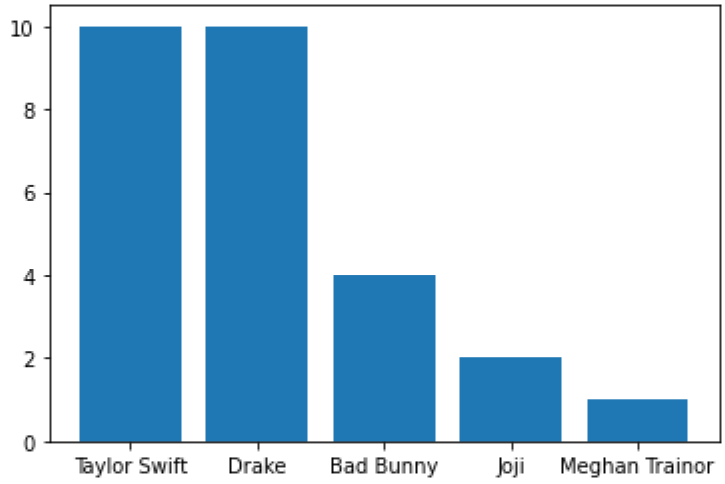


figure 3.

- Top 5 most popular genres:

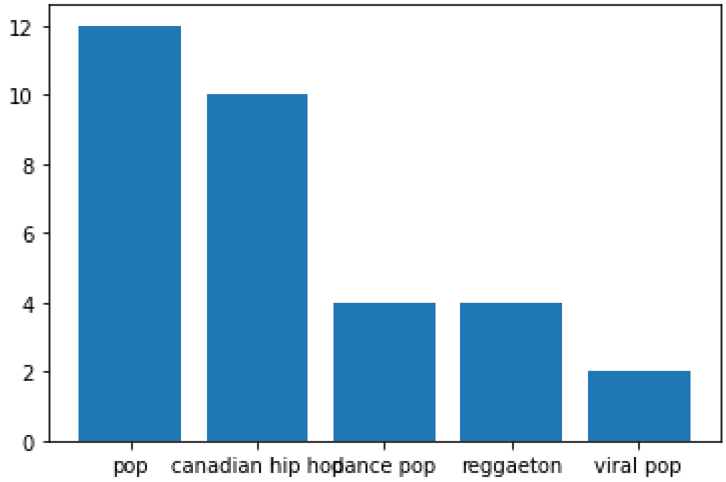
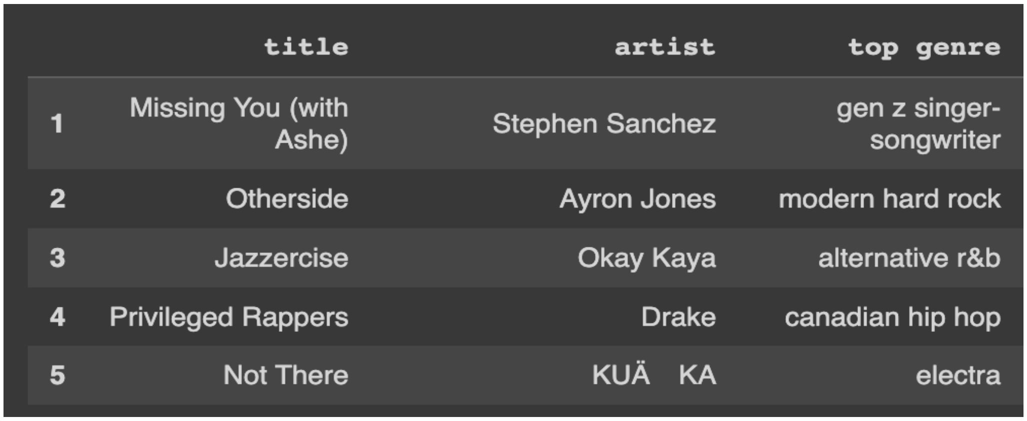


figure 4.

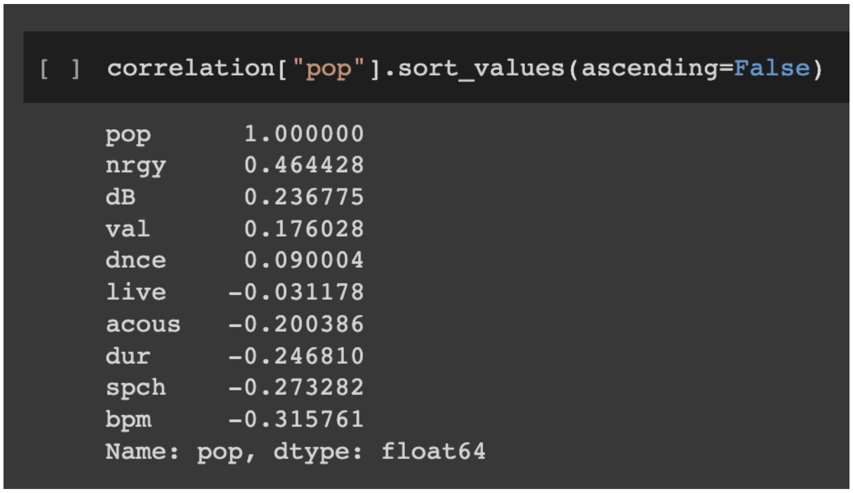
- Top 5 songs from the new releases:



	title	artist	top genre
1	Missing You (with Ashe)	Stephen Sanchez	gen z singer-songwriter
2	Otherside	Ayron Jones	modern hard rock
3	Jazzercise	Okay Kaya	alternative r&b
4	Privileged Rappers	Drake	canadian hip hop
5	Not There	KUÄ KA	electra

figure 5.

Correlation for Top50 dataset (figure 6 and 7):



```
[ ]  correlation["pop"].sort_values(ascending=False)

     pop       1.000000
     nrgy      0.464428
     dB        0.236775
     val       0.176028
     dnce      0.090004
     live     -0.031178
     acous    -0.200386
     dur      -0.246810
     spch     -0.273282
     bpm      -0.315761
     Name: pop, dtype: float64
```

figure 6.

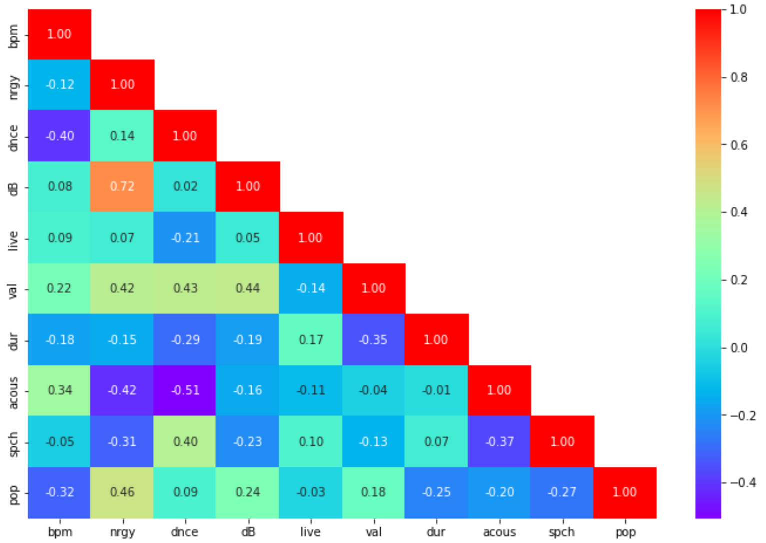
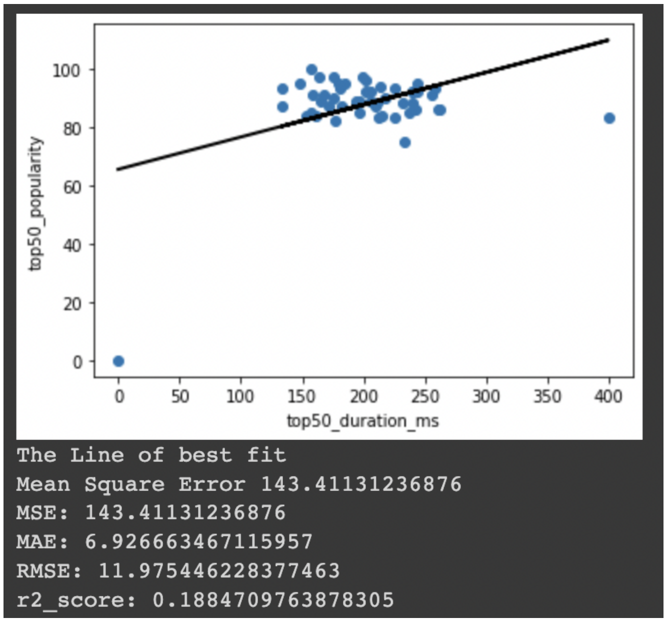


figure 7.

According to BBC, “Duration and tempo affects the pace and feel of any music you listen to”[3], hence why we predicted that duration would significantly affect the popularity score of a song. Therefore, we have found the line of best fit and Mean square Error for both dataset with their duration and popularity.

Duration vs. Popularity:

[3] (2022). What are duration and tempo? <https://www.bbc.co.uk/bitesize/topics/zcbkcj6/articles/z3yfng8#:~:text=Duration%20and%20tempo%20affects%20the,the%20speed%20of%20the%20music>



```
The Line of best fit
Mean Square Error 143.41131236876
MSE: 143.41131236876
MAE: 6.926663467115957
RMSE: 11.975446228377463
r2_score: 0.1884709763878305
```

figure 8.

MSE (Mean Square Error) computes the difference between the model’s estimated prediction and the actual popularity score. A lower MSE represents the model's prediction accuracy for a song's popularity score. By the formula:

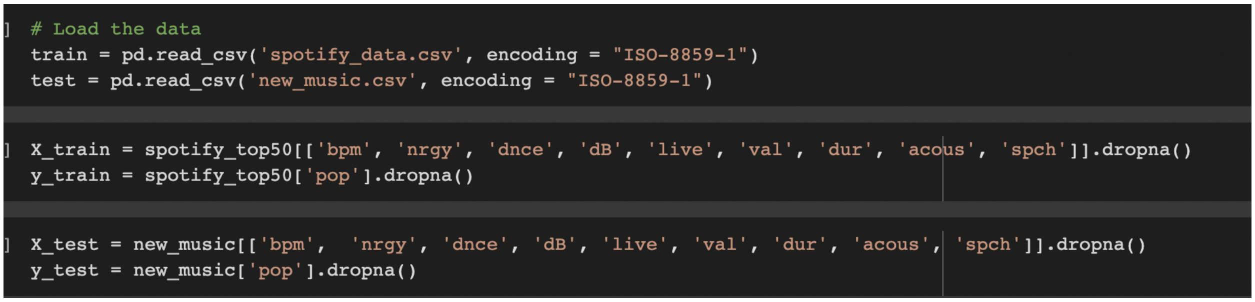
$$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(Predicted_i-Actual_i)^2}{N}}$$

This will also be used for modeling of Linear Regression, Decision Tree, Random Forest and K Nearest Model modeling. Figure 8 shows the line of best fit and MSE for Top 50 dataset which indicates duration is positively correlated with popularity of the song, as expected.

4. Modeling Approach

We analyzed the traits of the song by filtering out features that were measured by numerical values (float and integer) and calculating

Our modeling approaches for prediction are Linear Regression, Decision Tree, Random Forest, and K Nearest. Along these modeling techniques, we are aiming which would have a better insight of prediction. We had two datasets, which are split into train and test sets. As seen in figure 9, we trained the Spotify top 50 data set and tested on the newly released dataset.



```
# Load the data
train = pd.read_csv('spotify_data.csv', encoding = "ISO-8859-1")
test = pd.read_csv('new_music.csv', encoding = "ISO-8859-1")
```

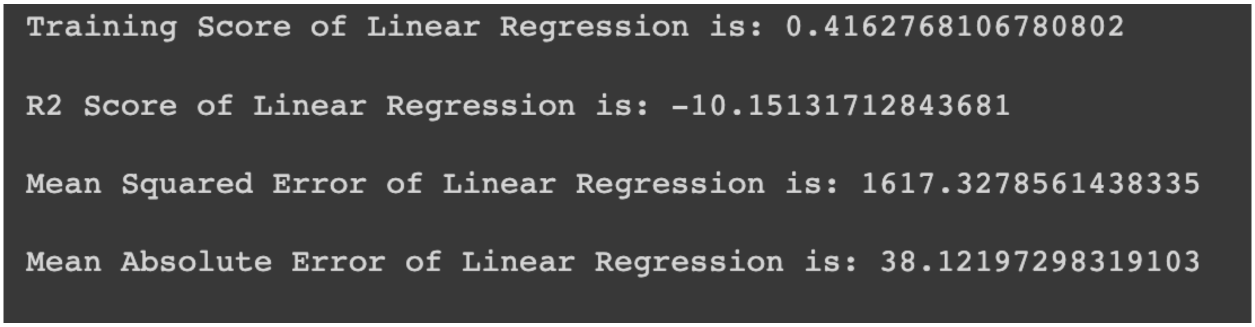
```
X_train = spotify_top50[['bpm', 'nrgy', 'dnce', 'dB', 'live', 'val', 'dur', 'acous', 'spch']].dropna()
y_train = spotify_top50['pop'].dropna()
```

```
X_test = new_music[['bpm',  'nrgy', 'dnce', 'dB', 'live', 'val', 'dur', 'acous', 'spch']].dropna()
y_test = new_music['pop'].dropna()
```

figure 9.

4.1. Linear Regression:

Using the SKLearn library for Linear Regression, we acquired the results below:

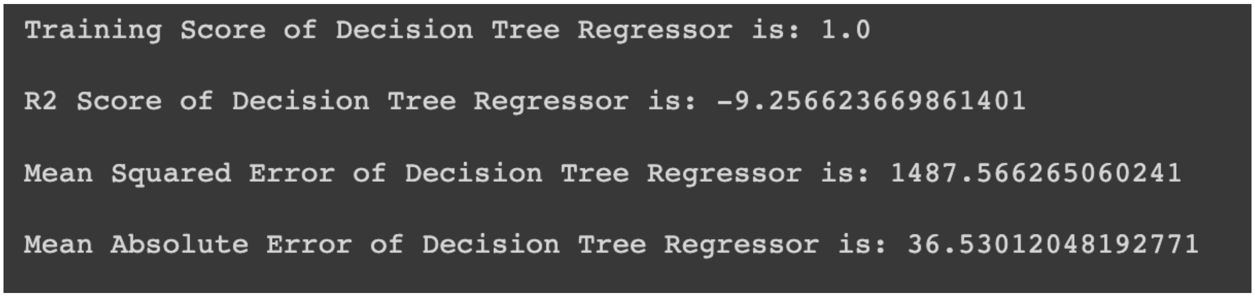


```
Training Score of Linear Regression is: 0.4162768106780802

R2 Score of Linear Regression is: -10.15131712843681

Mean Squared Error of Linear Regression is: 1617.3278561438335

Mean Absolute Error of Linear Regression is: 38.12197298319103
```

figure 10.

4.2. Decision Trees:

Using the SKLearn library for Decision Tree Classifier, we acquired the results below:



```
Training Score of Decision Tree Regressor is: 1.0

R2 Score of Decision Tree Regressor is: -9.256623669861401

Mean Squared Error of Decision Tree Regressor is: 1487.566265060241

Mean Absolute Error of Decision Tree Regressor is: 36.53012048192771
```

figure 11.

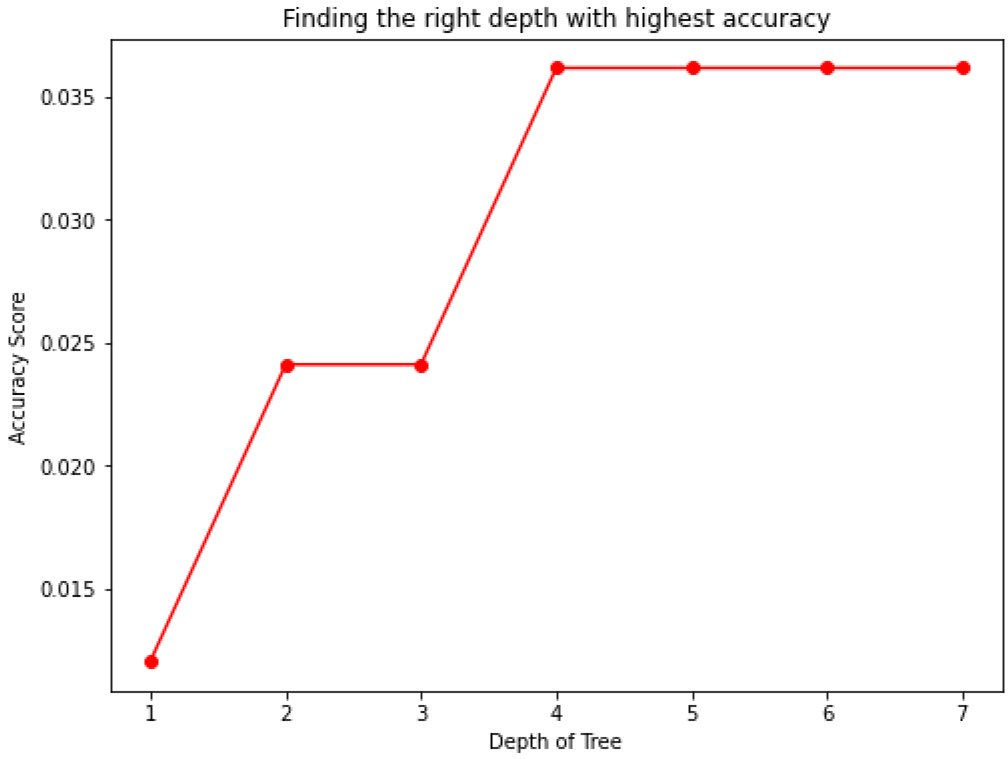
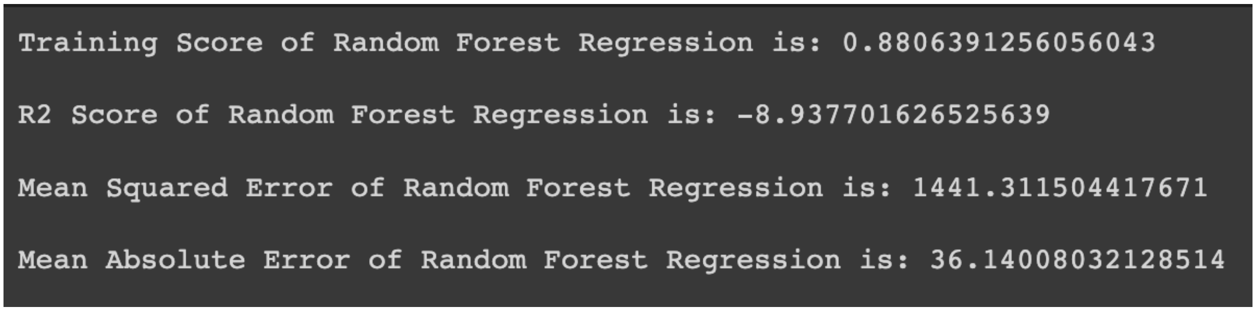


figure 12.

For Decision Trees modeling, we first find the best number of K with the highest accuracy. We have found the number of 4 is the best fit for finding decision tree regression. Our training percentage of the decision tree is 1.0, and R2 score of -9.26.

4.3. Random Forest:

Using the SKLearn library for Random Forest Classifier, we acquired the results below:



```
Training Score of Random Forest Regression is: 0.8806391256056043

R2 Score of Random Forest Regression is: -8.937701626525639

Mean Squared Error of Random Forest Regression is: 1441.311504417671

Mean Absolute Error of Random Forest Regression is: 36.14008032128514
```

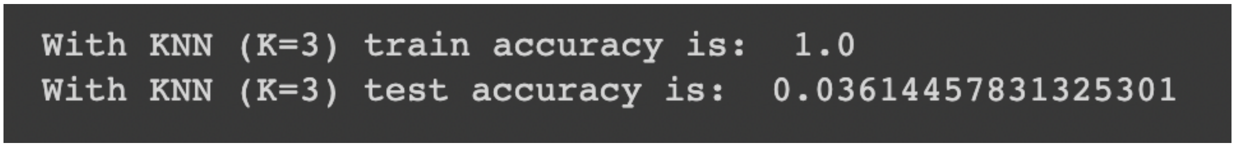
figure 13.

Random forest modeling is one of the most popular ensemble machine learning methods which use both classification and regression problems. We tried to avoid overfitting to select hundreds of different sub-samples randomly and reduce the variance. As a result, the training score of random forest regression is about 89% accuracy, and the R Square is -8.94.

The graph above depicts the models' accuracy before implementing any changes. The ***Random Forest*** model demonstrates the lowest prediction error.

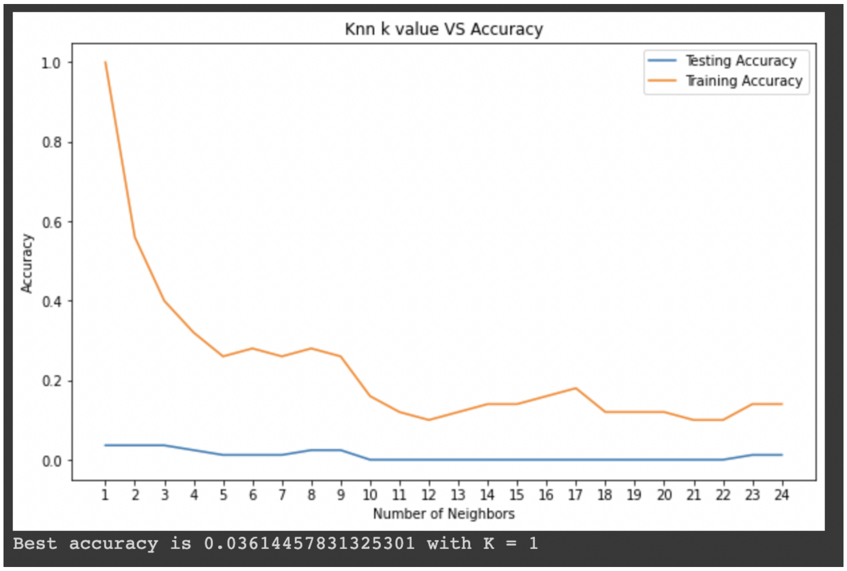
4.4. K Nearest Neighbor:

Using the SKLearn library for K Nearest Neighbor (KNN), we acquired the results below:



```
With KNN (K=3) train accuracy is:  1.0
With KNN (K=3) test accuracy is:  0.03614457831325301
```

figure 14.



```
Best accuracy is 0.03614457831325301 with K = 1
```

figure 15.

The accuracy score for the train data set of the KNN method is 1.0, implying a potential overfitting of our model. Therefore, the results for this model will be discarded.

5. Results and Conclusion

Throughout our modeling approach with four different techniques, we received the following results:

- Linear Regression: -10.15
- Decision Trees: -9.26
- Random Forest: -8.94
- K Nearest Neighbor: 0.04

As observed, all of our R Square values are smaller than 0, indicating that all of our models' predictions are worse than a constant function that always predicts the mean of the data. However, our best model seems to be Random Forest, so we will continue to explore this method further.

All our codes could be found in the accompanying Colab Notebook (shared to your email).

6. Limitations and Future Work

Since we did not reach desirable results for our study, we drew some conclusions on the limitations we have in our method. First of all, our data sets are too small, especially the train data sets, which should include a bigger number of songs for a more balanced model. As a result, we should conduct the study further after acquiring a full Spotify data set to train. Moreover, preferably, we should scrape for data for release dates of the songs as well, since popularity scores are calculated by Spotify in a way that songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past, which creates a temporal aspect for our data set. We will attempt to incorporate this variable in our future work.

References

1. Iqbal, M. (2022). Spotify Revenue and Usage Statistics (2022). <https://www.businessofapps.com/data/spotify-statistics/>
2. (2022) Get Tracks' Audio Features. Spotify for Developers. <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features>
3. (2022). What are duration and tempo? <https://www.bbc.co.uk/bitesize/topics/zcbkcj6/articles/z3yfng8#:~:text=Duration%20and%20tempo%20affects%20the,the%20speed%20of%20the%20music>

We also referred to the following sites for our codes:

4. Pierre, S. (2019). Analysis of Top 50 Spotify Songs using Python. <https://towardsdatascience.com/analysis-of-top-50-spotify-songs-using-python-5a278dee980c>
5. Hipolito, A. (2017). Spotify: Analyzing and Predicting Songs. <https://blog.mlreview.com/spotify-analyzing-and-predicting-songs-58827a0fa42b>
6. Rakshmitha. (2020). Spotify got Scikit Learn! <https://rakshmitha17.medium.com/spotify-got-scikit-learn-c9454b81fcbb>