A comparative analysis on music recommendation system using different ML models for predicting song popularity

CSE422 Final Project

Members:

Monthasir Delwar Afnan (20101247)

Rakib Hasan Rahad (20101010)

Shahidul Haque Chowdhury (19201058)

Section: 04

Table of Contents

Introduction

In recent years, music recommendation systems have gained immense popularity due to the rise of music streaming services. These systems are designed to provide personalised music recommendations to users based on their music preferences and listening history. One of the key components of a music recommendation system is predicting the popularity of songs. Predicting the popularity of songs can help music streaming services in recommending songs that are likely to be enjoyed by a user and increase user engagement on their platform.

Machine learning models are used to predict song popularity by analysing various features of the song, such as tempo, genre, artist, and lyrics. In this project, we aim to compare the performance of different machine learning models for predicting song popularity. We will explore how these models can be used to build a recommendation system that can provide users with personalised song recommendations. This will involve collecting and processing large datasets of song features and popularity ratings. We will then use various machine learning algorithms, such as logistic regression, decision trees, and SVM. Finally, we will evaluate the performance of these models using different metrics, such as accuracy, precision, and recall, to determine which model is best suited for music recommendation systems.

Motivation

The motivation for building a music recommendation system using different ML models for predicting song popularity lies in the potential to improve user experience and engagement with music streaming services. Accurately predicting song popularity can help music streaming services make better recommendations to users, leading to increased user satisfaction and retention. Moreover, understanding user preferences can help music streaming services tailor their recommendations to each individual user, creating a more personalised experience for the user.

Dataset Description

The dataset used in this report is taken from the official Kaggle repository. The data contains about 15 columns, each describing the track and its qualities. Songs released from 1956 to 2019 are included from some notable and famous artists like Queen, The Beatles, Guns N' Roses, etc. The dataset contains categorical attributes as well as numerical attributes that are combined and used for the prediction. This data contains audio features like Danceability, BPM, Liveness, Valence(Positivity) and many more.

Source:

Link:
https://www.kaggle.com/datasets/iamsumat/spotify-top-2000s-mega-dataset

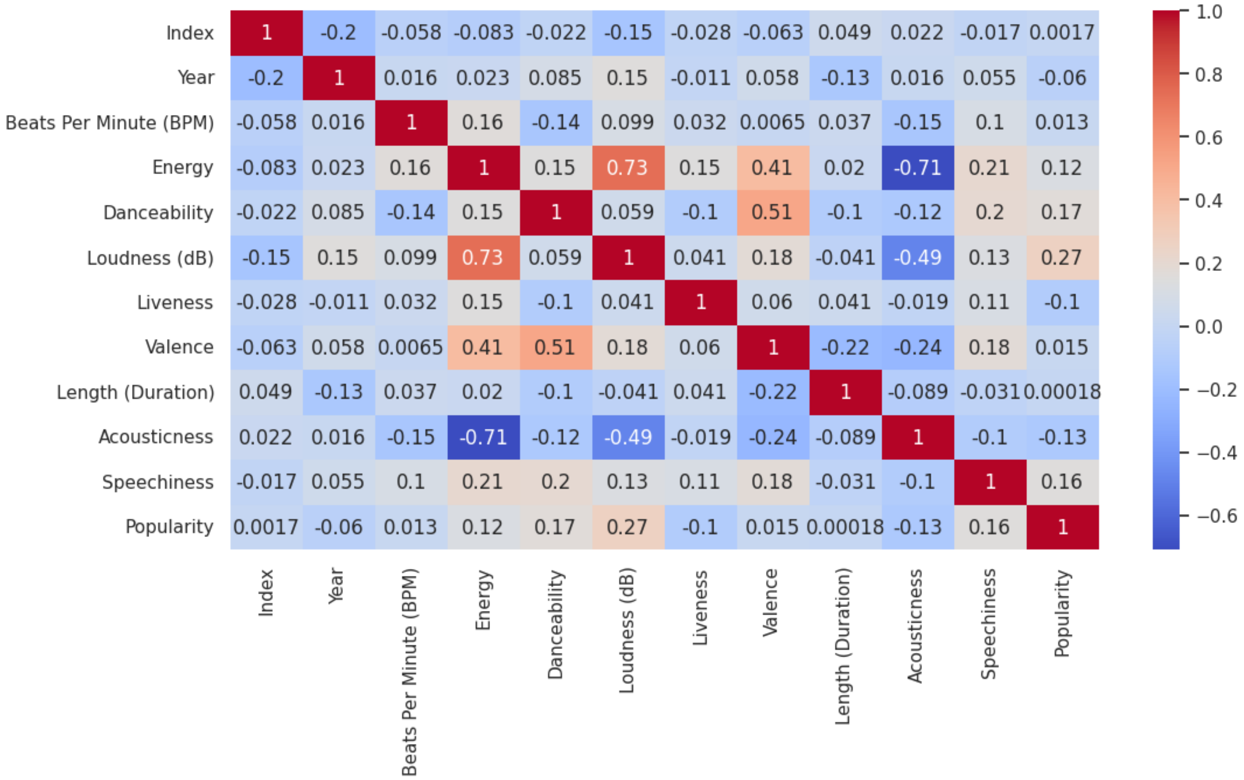
Reference:
This data is extracted from the Spotify playlist - Top 2000s on PlaylistMachinery(@plamere) using Selenium with Python. More specifically, it was scraped from http://sortyourmusic.playlistmachinery.com.

Column headers/Features, Label, Number of instances/Rows

- There are a total of 15 columns of headers or features in the dataset.

- These 15 features include 3 data types: int, float and object.

- This is the correlation between all the numerical features in our dataset.



	Index	Year	Beats Per Minute (BPM)	Energy	Danceability	Loudness (dB)	Liveness	Valence	Length (Duration)	Acousticness	Speechiness	Popularity
Index	1	-0.2	-0.058	-0.083	-0.022	-0.15	-0.028	-0.063	0.049	0.022	-0.017	0.0017
Year	-0.2	1	0.016	0.023	0.085	0.15	-0.011	0.058	-0.13	0.016	0.055	-0.06
Beats Per Minute (BPM)	-0.058	0.016	1	0.16	-0.14	0.099	0.032	0.0065	0.037	-0.15	0.1	0.013
Energy	-0.083	0.023	0.16	1	0.15	0.73	0.15	0.41	0.02	-0.71	0.21	0.12
Danceability	-0.022	0.085	-0.14	0.15	1	0.059	-0.1	0.51	-0.1	-0.12	0.2	0.17
Loudness (dB)	-0.15	0.15	0.099	0.73	0.059	1	0.041	0.18	-0.041	-0.49	0.13	0.27
Liveness	-0.028	-0.011	0.032	0.15	-0.1	0.041	1	0.06	0.041	-0.019	0.11	-0.1
Valence	-0.063	0.058	0.0065	0.41	0.51	0.18	0.06	1	-0.22	-0.24	0.18	0.015
Length (Duration)	0.049	-0.13	0.037	0.02	-0.1	-0.041	0.041	-0.22	1	-0.089	-0.031	0.00018
Acousticness	0.022	0.016	-0.15	-0.71	-0.12	-0.49	-0.019	-0.24	-0.089	1	-0.1	-0.13
Speechiness	-0.017	0.055	0.1	0.21	0.2	0.13	0.11	0.18	-0.031	-0.1	1	0.16
Popularity	0.0017	-0.06	0.013	0.12	0.17	0.27	-0.1	0.015	0.00018	-0.13	0.16	1

- As our target feature is “Popularity”, the correlation between popularity and loudness is the highest [0.27].

Biasness/Balanced

- All unique classes have equal number of instances in our dataset.

Dataset pre-processing

Faults:

NULL Values:

To check if any null values are present in our dataset, we used the isnull() function from the pandas library. This function returns boolean values, True or False. In our dataset, 8 columns had null values.

Solutions:

There were 8 columns of features in that dataset that had null values: Beats Per Minute, Energy, Danceability, Loudness, Liveness, Valence, Length and Acousticness. To deal with the null values, we used the dropna() function of the pandas library to drop the entire row that had the null value.

Encoding:

We used one hot encoding as machine learning algorithms cannot directly handle categorical data and require numerical data as input. Each category is assigned a unique binary code, where only one bit is set to 1, and all other bits are set to 0. After we apply one hot encoding, it will create n no. of columns for n no. of classes.

Feature Scaling

We applied scaling in our target feature, “Popularity.” We used the Standard Scaler method to scale our target feature.

$$(X-\text{mean})/\sigma \ [\ \sigma=\text{standard deviation}\ ]$$

We used StandardScalar function from sklearn.preprocessing library.

Dataset splitting

- Train set 80%
- Test set 20%

We used train_test_split function from sklearn.model_selection library. Among 1039 instances, we used 80% of the data for the train set and the rest 20% for the test set. X_train used 831 instances, and X_test used 208 instances.

Model training

For model training, we used four types of models.

- KNN
- Descison Tree
- SVM
- Logistic Regression

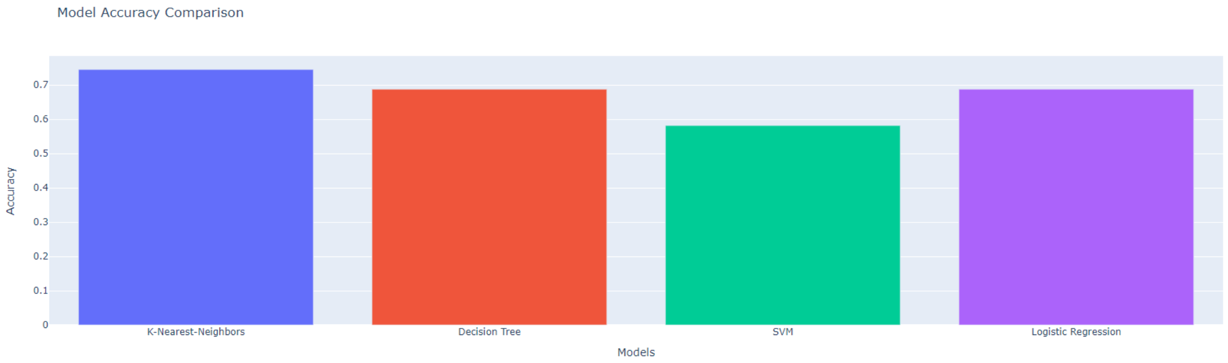
After applying standard scaling, we used all the above models to predict the accuracy. After applying the accuracy scores are:

K-Nearest-Neighbors Accuracy: 0.6875
Decision Tree Accuracy: 0.5817307692307693
SVM Classifier Accuracy: 0.6875
Logistic Regression Accuracy: 0.7451923076923077

We also used the MinMax Scalar method, but the accuracy score was slightly lower than Standard Scaler. In Logistic Regression, the highest accuracy was found 0.71 using MinMax Scalar method and 0.745 using Standard Scaler.

Comparison analysis

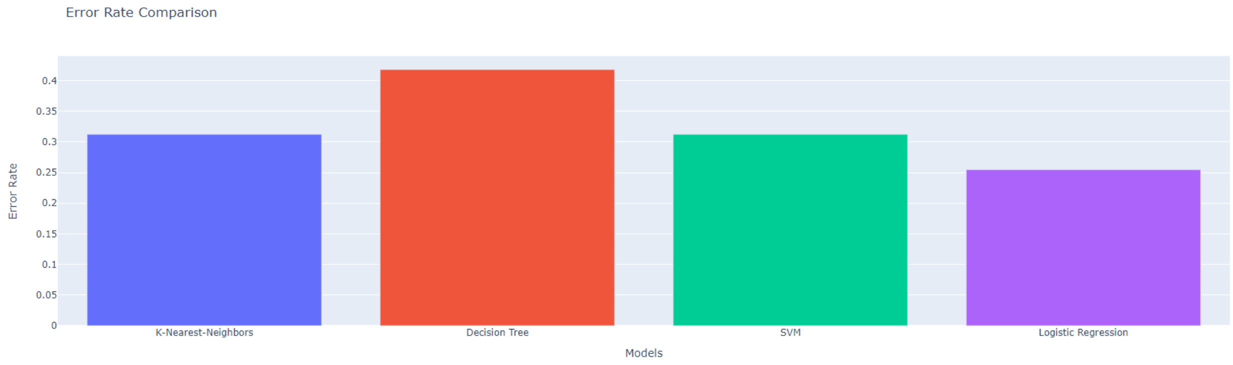
Accuracy Bar Chart:



Accuracy Score:

K-Nearest-Neighbors Accuracy: 0.6875
Decision Tree Accuracy: 0.5817307692307693
SVM Classifier Accuracy: 0.6875
Logistic Regression Accuracy: 0.7451923076923077

Error Rate Bar Chart:



Error Score:

KNN error: 0.3125
Decision Tree error: 0.4182692307692307
SVM error: 0.3125
Logistic Regression error: 0.2548076923076923

Conclusion:

Music recommendation systems have become essential to music streaming services in recent years. These systems use different machine learning models to analyse user behaviour data and song features to provide personalised music recommendations to users. Building a music recommendation system using different ML models to predict song popularity can transform how users discover and enjoy music, leading to a more satisfying and personalised music streaming experience. By improving the accuracy of song recommendations, music streaming services can increase user engagement and retention, benefiting users and music streaming services.

Future work/Extension

Using advanced machine learning algorithms, such as deep learning models, for predicting song popularity, these models can analyse more complex features of the song, such as the melody and chord progression, and lead to more accurate recommendations. Predicting song popularity includes exploring more advanced machine learning algorithms, incorporating real-time data and user context, and integrating social network analysis. These efforts can lead to better music recommendations, improving the overall user experience of music streaming services.