PREDICTING SPOTIFY'S POPULAR SONGS USING THE STRUCTURE OF POPULAR SONGS IN THE PAST TWO DECADES

ABSTRACT

Spotify is the fastest growing digital music service platform that gives access to millions of songs. Spotify rewards music artists financially when their songs are streamed repeatedly. Music distributors and record labels may also have a share in these financial rewards depending on the contract between music artists and Spotify. This project therefore seeks to help artists, music distributors and record label managers maximize financial benefits from their association with Spotify by building a model that will help predict songs that will be popular based on structing of the songs. With the help of data wrangling tools like NumPy and Pandas, datasets from Kaggle Spotify 1 million tracks and annual GDP of the United States from 1929 to 2023 were cleaned by dropping irrelevant columns and columns with missing values, replacing missing values with 0, converting duration column from milliseconds to minutes and merging cleaned datasets into a single dataset. The final features of the merged dataset were genre, year, popularity, danceability, liveness, instrumentalness, tempo, acousticness, duration, GDP year and GDP. Exploratory data analysis followed and relationships between features were highlighted and visualized with matplotlib and seaborn line, scatter and histogram plots. The distributions of each feature were shown and the dependent feature, popularity was found to have a left skewed distribution with a long tail. This project also found that highly instrumental and acoustic songs were not loud songs, songs with longer durations were not popular and the most popular song genre was pop music. Generating a heatmap revealed that year was strongly correlated to popularity whereas tempo was not. Liveness, acousticness, instrumentalness, GDP year, GDP and duration were negatively correlated to popularity. Some independent features were also found to exhibit multicollinearity. These features were instrumentalness versus loudness, acousticness versus loudness and GDP_year versus GDP. Numerical features were scaled, and categorical features converted to numerical using One Hot Encoder. Four predictive machine models were built and assessed. Three of which were linear models, namely, Ridge, Elastic Net, Decision Tree Regressor and one was non-linear model: Random Forest Classifier. Assessing the performance of Random Forest model showed accuracy and f1 score was 0.15 and 0.11 respectively. A ROC graph also showed an inverted curve with the area under the curve of 0.21. These values indicate that the non-linear model was performing poorly. The performance of the linear models was also evaluated and compared. The Ridge model performed best with the highest R-squared score of 0.54 and lowest Mean Absolute Error and Mean Squared Error 8.20 and 10.69 respectively. The all features in the dataset were found to be equally important to the model and is indicative of a model requiring more data to improve. This project was limited by the number of features in that dataset.

INTRODUCTION

Music is a well-known food for the soul. Its ability to stimulate the brain to improve memory, relief stress and pain and create social cohesion has been long studied and documented. Spotify, in their course of expanding the reach of all types of music have played a very instrumental role in capitalizing on the current global digitization to provide music that can be easily streamed on their digital platform. Spotify maintains several playlists that feature the most streamed songs across different time frames. Spotify also offers financial rewards to music artists who have highly streamed songs. These rewards may extend to music producers and record labels and music distributors depending on contracts signed between Spotify and music artists. As of August 2024, Spotify’s market capitalization is valued at approximately $64.25 billion. The driving force of Spotify’s revenue generation is the sales of a range of subscriptions the allows Spotify customers to use the digital platform to stream music of their choice. This marketing strategy has resulted in attracting new customers and retaining new ones. It has also encouraged customers to switch from Ad-Supported Services to more expensive Premium Services. The overall modus operandi of Spotify is very attractive to music artists and producers who are willing to use the platform to their financial advantage and to boost their fame across the globe.

This project seeks to help artists, music distributors and record label managers maximize financial benefits from their association with Spotify by building a model that will help predict songs that will generate global appeal and cause them to be streamed repeatedly based on the structure of the song alone. This model will be able to predict marketable songs even before they are released and uploaded onto Spotify.

The model will operate with the assumption that successful songs will end up with high popularity values on Spotify's most popular song list in future.

Problem Statement

What predicted song structure can influence the rating of unreleased songs on Spotify's most popular song list based on the Kaggle Spotify 1 million tracks dataset from 2000 to 2023?

METHODS

DATA COLLECTION

The key data source for this project was extracted from Kaggle. The Kaggle dataset was extracted from Spotify platform using the Python library ”Spotipy". Spotipy allows users to access music data provided via API’s. The dataset included about 1 million tracks with 19 features between 2000 and 2023.

The popularity column ranks songs from 0-100 in the Kaggle Spotify 1 million tracks dataset. Songs with high popularity ranks could be assumed to have garnered significant streams on the platform. Other features in the dataset includes genre, danceability, instrumentalness, year, track_id, artist name, track_name, tempo, mode, key, valence, energy, loudness, liveness, acousticness, duration(milliseconds), speechiness and time signature.

A supplementary dataset downloaded from usafacts.org on the annual GDP of the United States from 1929 to 2023 was also used. This data had 5 rows and 96 columns. The columns were years from 1929 to 2023 and the rows were Gross Domestic Product, By type, Private Industries, By industry and Agriculture, forestry and fishing row.

DATA WRANGLING

Handling missing columns

The main and supplementary datasets were cleaned with the help of tools like pandas and numpy. In the main dataset 15 values were missing in artist_name column and 1 value was missing in track_name column. These columns were deleted.

The track_name , energy , speechiness, mode, key and time signature columns were also deleted because the track_name was irrelevant to the project and the other columns did not have variable values.

Missing values in the supplementary dataset were found in all columns and all rows except the Gross Domestic Product row. With respect to this project the rows with the missing columns did not have any relevance so they were deleted.

The dataset was then transposed, and the resulting shape was 1 row and 96 columns. Data from the years prior to 2000 was deleted and the resulting columns were renamed GDP_year and GDP.

Unit conversion

The duration column which was originally in milliseconds was converted to minutes and renamed duration_min.

Merging of datasets

The cleaned datasets were merged into one dataset using a left join. The resulting shape of the dataset was 1159764 rows and 12 columns. The resulting features were popularity, genre, year, danceability, liveness, instrumentalness, tempo, acousticness, loudness, duration_min, GDP_year and GDP.

EXPLORATORY DATA ANALYSIS

The distribution of each feature in this new data frame was visualized using data visualization tools like matplotlib and seaborn to create histograms. The dependent feature(popularity) had a left skewed and long tail distribution. The independent features had different distributions shapes.

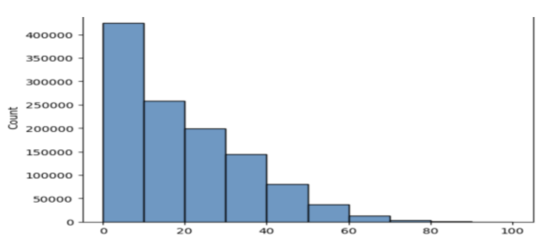

Fig 1 A histogram showing the distribution of Popularity

Relationships between features were explored using line, scatter, bar plots and histograms. Observations from Exploratory analysis revealed that:

1. The song feature, instrumentalness, was not related to loudness and this is evident from the graph below.

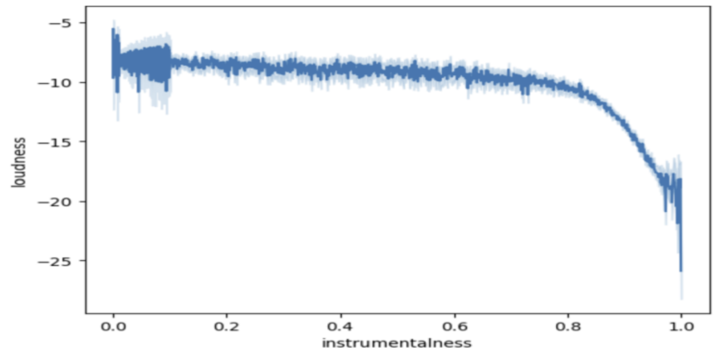

Fig 2 A line plot of loudness versus instrumentalness

2. Songs with longer durations are not popular songs and this is shown in the graph below.

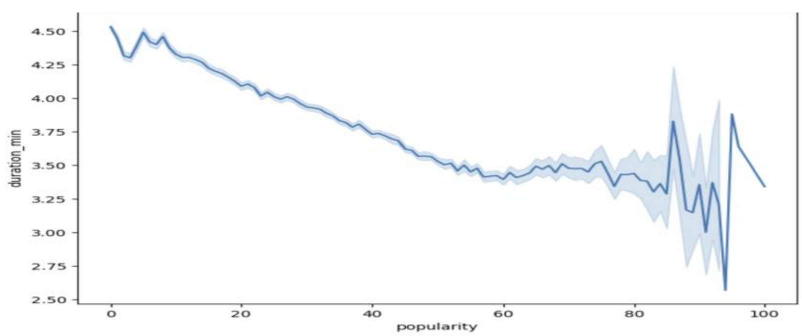

Fig 3 A line plot of duration_min versus popularity

3. EDA reveled that popular songs have low acousticness

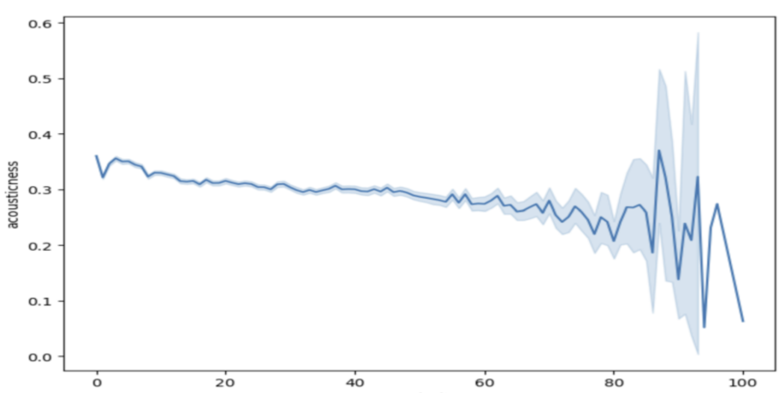

Fig 4 A line plot of acousticness versus popularity

4. The most popular song genre was found to be pop.

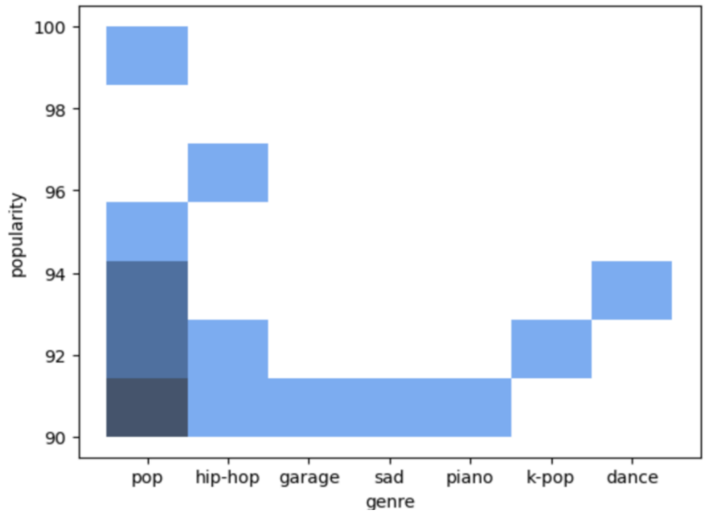

Fig 5 Histogram genres of the first twenty most popular songs

A heatmap, which further revealed the relationships between the dependent feature and each numerical independent feature as well as the relations within numerical independent variables was created with seaborn.

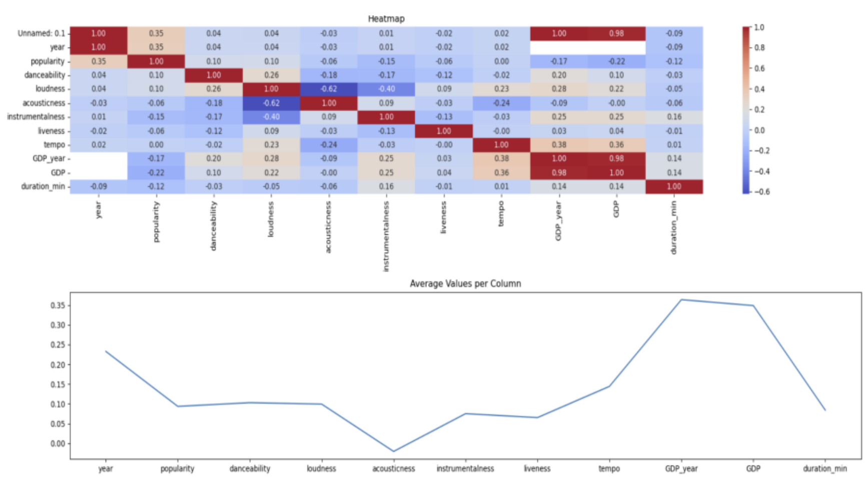

Fig 6 Heatmap showing correlation between numerical variables

The finding from the heatmap graphs above showed that, popularity was:

1.not correlated to tempo

2.weakly correlated to danceability and loudness.

3.negatively corelated to acousticness, liveness, instrumentalness, duration_min, GDP_year, GDP

4.strongly correlated to year

The heatmap also exposed multicollinearity which is the strong association between some of the independent variables and this is shown in the graphs below.

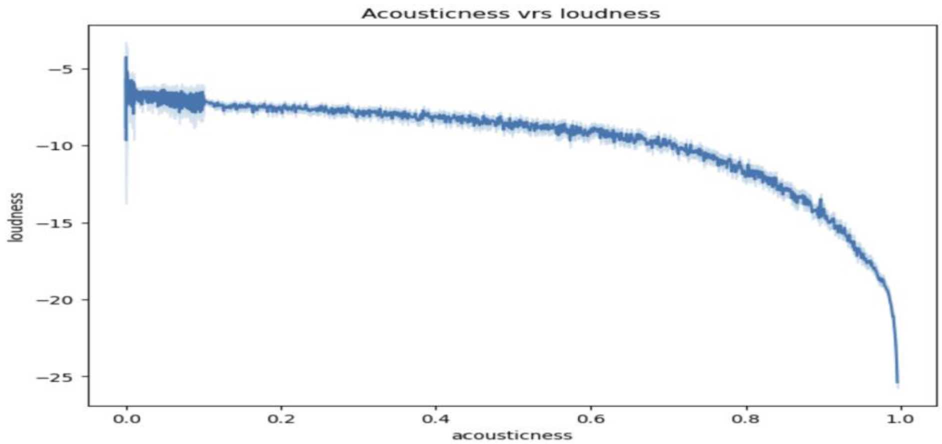

Fig 7 A line plot of loudness versus acoustiness

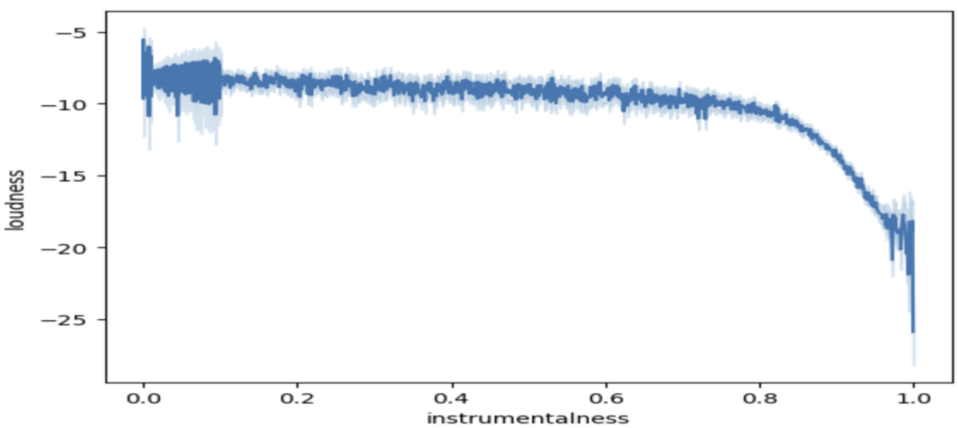

Fig 8 A line plot of instrumentalness versus loudness

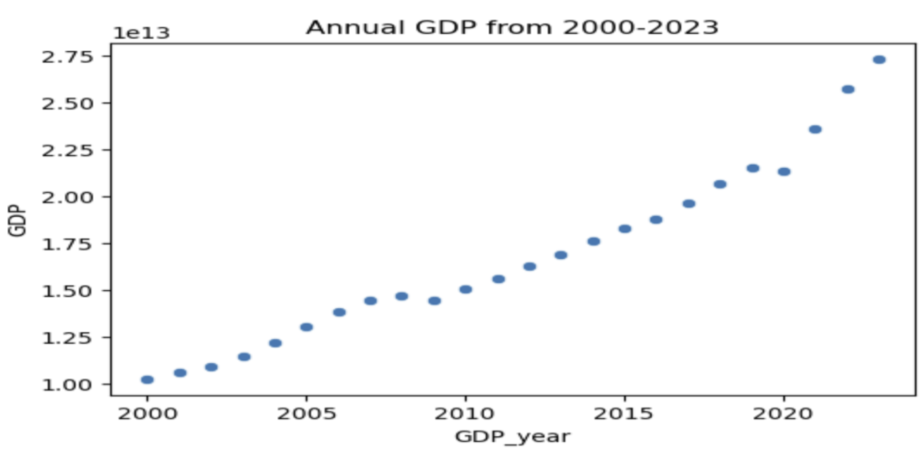

Fig 9 A scatter plot of GD versus GDP_ year

PREPROCESSING

Data was split into 75% training sets and 25% test sets using the train_test_split from sklearns model selection. The split resulted in the creation of four sets namely, X_train,y _train, X_test, y_test sets. The shape of data in the train sets was X_train had 869823 rows and 12 columns, and y_train had 869823 rows and no columns. The test sets also had X_test having the shape 289941 rows and 12 columns and y_test had 289941 rows and no columns.

```
genre                 object
year                   int64
danceability         float64
loudness             float64
acousticness         float64
instrumentalness     float64
liveness             float64
tempo                float64
duration_ms            int64
GDP_year               int32
GDP                  float64
duration_min         float64
dtype: object
```

Fig 10 A list of the data types in both the training and test sets.

The features in the the data sets were transformed into numerical forms that makes it easy for machine learning algorithms to work with. The categorical feature (genre) was transformed into numerical column using OneHotEncoder and the numerical columns were scaled with StandardScaler. These two data transformations were concatenated to form a single feature space with ColumnTransformer.

Using these insights into the features of the data, the preprocessing stage was commenced with defining X and y variables and splitting data into 75% training set and 25% test sets. Categorical features were converted into numerical features using OneHotEncoder and numerical features were scaled with StandardScaler. Both transformations were combined into a single feature using ColumnTransformer and used to fit the training sets.

MODELLING

At the end of project, four machine learning models were built. Three of these models used the linear regression algorithm. These were the Ridge, Elastic Net and DecisionTreeRegressor models. The only non-linear model built was the Random Forest Classifier.

Linear models.

The Ridge , ElasticNet and DecisionTree Regressor algorithms were each used to build a predictive models for the project. Each of these models were first defined together with the ColumnTransformer using a make_pipeline from sklearn.pipeline. Each of the three models pipelines were used to fit training data (X_train, y_train) and predict the test set(X_test).

The Ridge Regression model had its alpha parameter set to 0.1 and the Decision Tree Regressor had its max depth parameter set to 2.

Cross_ validate from sklearn's model_selection was used to fit and assess the training set. This step was necessary because cross validate offered a more robust and comprehensive evaluation of the model's performance.

The R -squared scores, Mean abosolute error (MAE) and Mean squared error (MSE) values calculated for each of the linear models built.

Non-linear model (Random Forest Model)

A non- linear model was built using the Random Forest Classsifier algorithm. The number of estimators was set to 8. The algorithm was defined together with the ColumnTransformer using a make_pipeline from sklearn.pipeline. The models pipelines were used to fit training data (X_train, y_train) and predict the test set(X_test).

The accuracy and the F1 score were calculated as a way of assessing the performance of the model.

A receiver operating characteristics curve was constructed to further evaluate model performance.

RESULTS AND DISCUSSION

Linear Models

The assessment each of the built linear model's performance comparing the R-square score, Mean absolute error and the Mean squared error scores showed that the Ridge model performed the best because it had the highest R-squared score and the lowest MAE and MSE values.

Table 1 Results of assessment of linear models

Model Assesment	Ridge Model	ElasticNet	DecisionTreeRegressor
R-square	0.54	0.13	0.15
Mean Absolute Error	8.02	12.10	11.79
Mean Squared Error	10.69	14.81	14.61

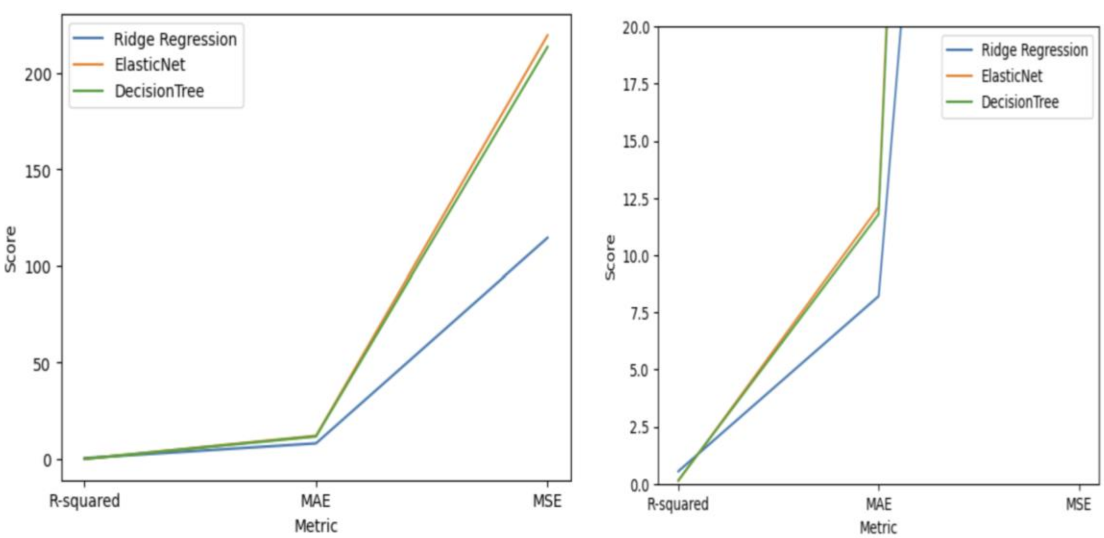

Fig 11 A line plot showing the performance of all three linear models.

Non-Linear Model

The Random Forest Classifier was assessed by the accuracy and f1scores.

```
# Predicting test values with RF model

y_pred = model_res.predict(X_test_loaded)
y_pred_prob = model_res.predict_proba(X_test_loaded)
lr_probs = y_pred_prob[:,1]

# Assessing RF model

ac = accuracy_score(y_test_loaded, y_pred)

f1 = f1_score(y_test_loaded, y_pred, average='weighted')
cm = confusion_matrix(y_test_loaded, y_pred)

print('Random Forest: Accuracy=%.3f' % (ac))

print('Random Forest: f1-score=%.3f' % (f1))
```
```
Random Forest: Accuracy=0.146
Random Forest: f1-score=0.118
```

Fig 12 The calculation and results of the accuracy and f1 score.

The results show that the model is not performing well. The accuracy and the f1 score are very low.

The results of the ROC graph further highlight the poor performance of the model.

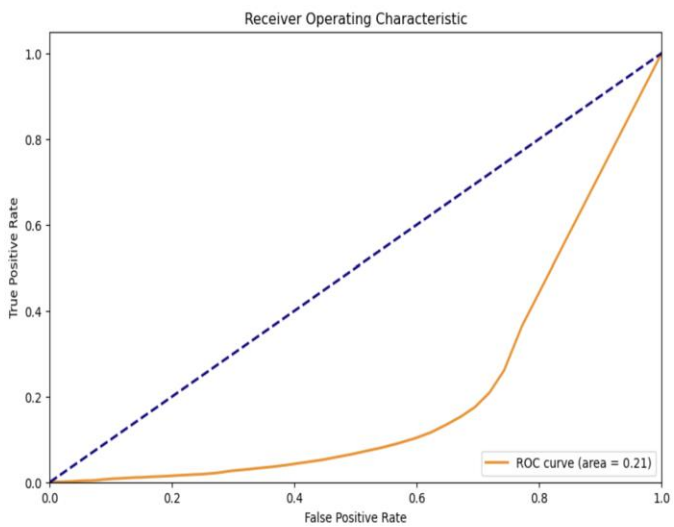

Fig 13 ROC curve showing the performance of the Random Forest Model.

The inverted ROC curve above shows that the models' predictions are worse than random guessing. This may be because of incorrect labeling of positive and negative predictions, or the negative predictions outnumber the positive predictions of the model. It may also be because the model is not suitable for the data.

For a good model, the Area under the curve value must be one or close to one. In the case of this Random Forest model the area under the curve is 0.21 which is close to 0. This means that the model has the worse measure of separability and is therefore making wrong predictions.

Choosing the best model

After assessing all four predictive models built for the project, the Ridge models work the best and will be considered as the working model for the project.

To further understand how the model operates, the important features of the model will be calculated using the ability model's to predict training columns accurately using the models coefficients.

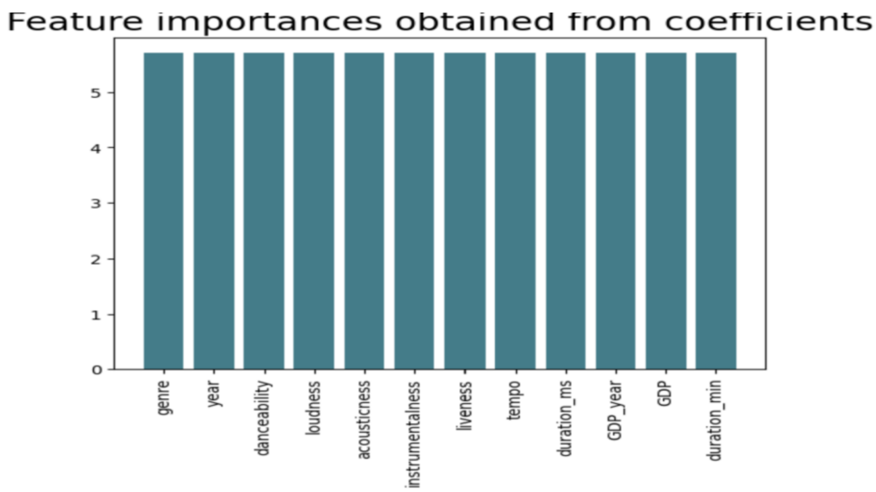

Fig 14 A bar chart showing feature importance of Ridge model

The graph shows that all the features in the data are equally important to the model.

<u>RIDGE MODEL IMPROVEMENT</u>

The current performance of the Ridge Model could be improved by acquiring more data. Acquiring more data will increase the number of features and enhance generalizations. More data will also improve the robustness of the model and lead to better feature engineering with will offer deeper insights to the predictions of the model.

LIMITATIONS

The constraints with the solution space are that the proposed model was built based on the Kaggle dataset trends only and this model may not be able to account for all songs that make it to the list based on other factors not considered in this project.

CONCLUSION

The Ridge Regression model will be able to predict the popularity of unreleased songs on Spotify, based on the genre, year, danceability, loudness, acousticness, instrumentalness, liveness, tempo, duration of song in minutes and the GDP of the year of intended song release on the Spotify platform.