UNIVERSITY OF NORTH TEXAS
INFO 5502 (Principals and Techniques for Data Science)

Spotify Song Popularity Prediction

Abstract:

Title of the project
Team members
Abstract
Dataset specifications
Data processing techniques
Flowchart of design
Data processing techniques
Models & algorithms
Model evaluation
Project results
Project milestones
Repository / archive
Project requirements
References
Conclusion & future work
Source Code

UNIVERSITY OF NORTH TEXAS
INFO 5502 (Principals and Techniques for Data Science)

PROJECT REPORT

TITLE OF THE PROJECT: Spotify Song Popularity Prediction

GROUP NAME: Group 07

TEAM MEMBERS:

STUDENT NAME	STUDENT ID	ROLES AND RESPONSIBILITES
Veepura chari Dharmavarapu	VeepuraChariDharmavarapu@my.unt.edu	Model Preparation and Validation
Lavanya Pobbathi	LavanyaPobbathi@my.unt.edu	Data Preparation and modelling
Sowmya Ushake	SowmyaUshake@my.unt.edu	Data Pre-processing
Chong Zhang	ChongZhong@my.unt.edu	Data Analysis and Visualization
Chintureddy Baireddy	ChintureddyBaireddy@my.unt.edu	Feature Analysis and Validation

ABSTRACT:

Spotify is one of the most popular music streaming services offering over 50 million songs and 700,000 podcasts. About 40,000 new songs are added to Spotify every day! So how does a song become popular on Spotify? Do the most popular songs share any common characteristics?

The aim of our project is to build a machine learning model to predict the popularity of a song based on some features like loudness, danceability, energy, tempo etc. The report outlines the data collection process, data cleaning, and the development of the model. Results from the model are presented, along with an analysis of the accuracy of the predictions, validation of the model and a

discussion of the implications for future projects. We test four models on our dataset. Our best model was random forest, which was able to predict popularity song success with 84% accuracy.

This popularity prediction model can provide valuable insights into the customer preferences and trends in music industry and helps to make educated decisions about music production, marketing and revenue generation.

DATASET SPECIFICATIONS:

We have used two datasets from Kaggle site which, in turn, was originally collected from the Spotify API. Each observation represents an individual song and these datasets called as 'song_data' which contains 6398 unique songs, it used for creating the model and another one is 'song_validate' which contains 5876 song records which is used for evaluating the model. Each dataset has 19 attributes, 16 of them numerical and 3 of them categorial values. Actually, our dataset is cleaned one without having null values. As per our target, we have changed the dependent variable of Popularity Score - a value between 0 and 100 based on total streams into 0 or 1 value as a target.

DATA FEATURES:

Each record contains 19 features categorized by track information, artist information, album information, and audio analysis features. We describe these feature in detail below:

➢ Categorial Values:

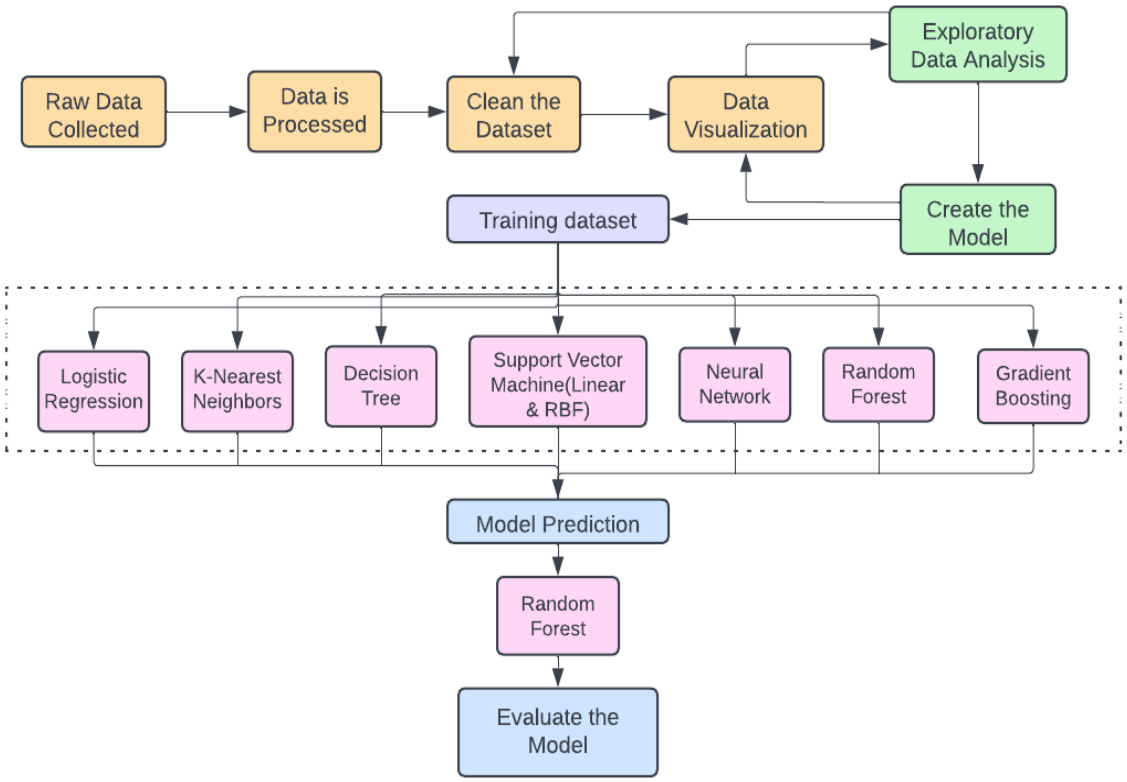
- *Track* – the song's unique Spotify track title
- *Artist_title* – the artist's title of the song
- *Uri* – song url

➢ Numerical Values:

- *Danceability* – It 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 the most danceable.
- *Energy* – a value measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
- *Key* – The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. Ex: 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
- *Loudness* – The overall loudness of a track in decibels (dB). Values typical range between -60 and 0 dB.
- *Mode* – It indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
- *Speechiness* – a value from 0.0 to 1.0 describing the amount of spoken words present in the track. Values close to 1.0 indicate exclusively speech-like tracks (e.g. podcast, audio book, poetry).
- *Acoustiness* – a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- *Instrumentalness* – Predicts whether a track contains no vocals. The closer value is to 1.0, the greater likelihood the track contains no vocal content.

- *Liveness* – a value from 0.0 to 1.0 that describes the presence of an audience in the track. Values closer to 1.0 represent tracks that were performed live.
- *Tempo* – The overall estimated tempo of a track in beats per minute (BPM).
- *Valence* – A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive.
- *duration_ms* – the duration of the track in milliseconds.
- *time_signature* – an estimated overall time signature of a track.
- *chorus_hit* – a value from 0 to 100 measures the hit range
- *sections* – a value from 0 to 200 indicates section of the song
- *popularity* –The popularity of the track. The value will be between 0 and 100, with 100 being the most popular.

FLOWCHART OF DESIGN:



DATA PROCESSING TECHNIQUES:

Data Cleaning:

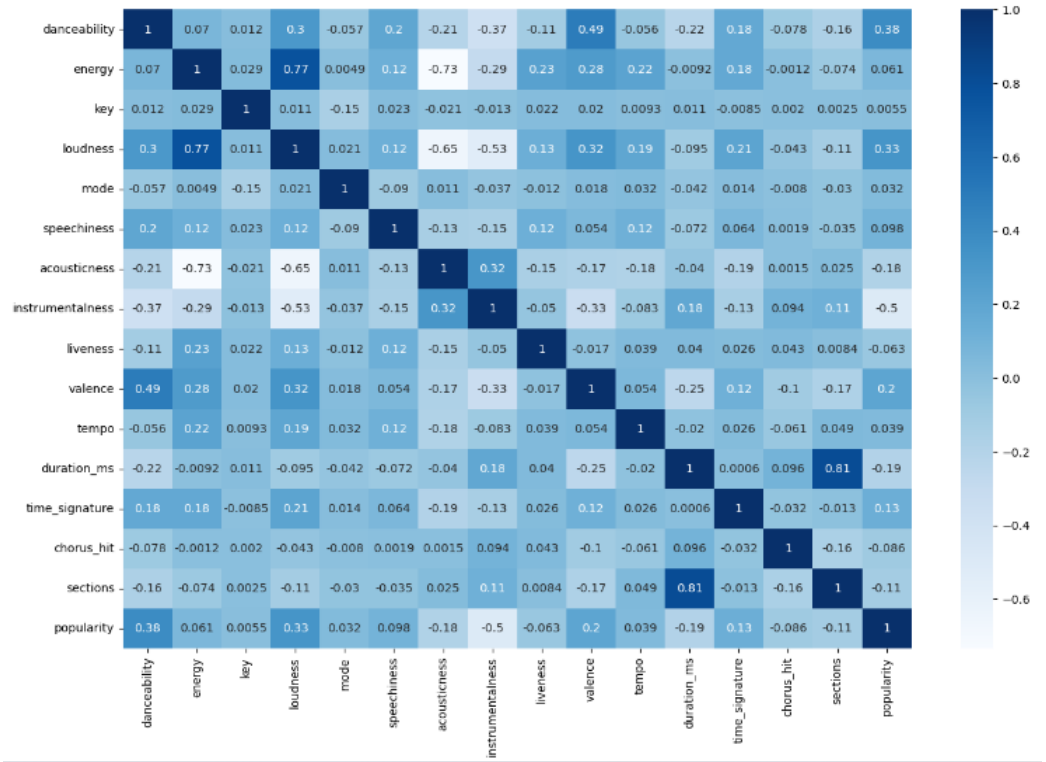
As our dataset is cleaned, we are still checked the duplicate and null value records in the dataset.

```
track                0
artist               0
uri                  0
danceability         0
energy               0
key                  0
loudness             0
mode                 0
speechiness          0
acousticness         0
instrumentalness     0
liveness             0
valence              0
tempo                0
duration_ms          0
time_signature       0
chorus_hit           0
sections             0
target               0
dtype: int64
```

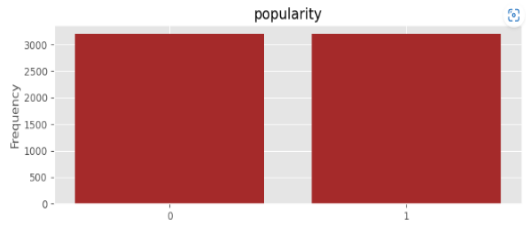
```
track               6398
artist              6398
uri                 6398
danceability        6398
energy              6398
key                 6398
loudness            6398
mode                6398
speechiness         6398
acousticness        6398
instrumentalness    6398
liveness            6398
valence             6398
tempo               6398
duration_ms         6398
time_signature      6398
chorus_hit          6398
sections            6398
popularity          6398
dtype: int64
```

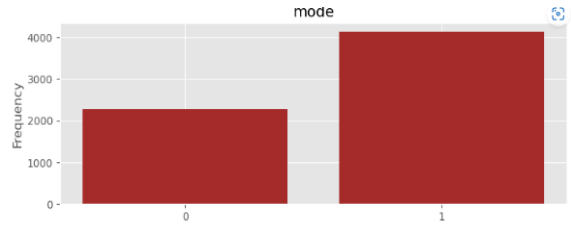
Exploratory Data Analysis:

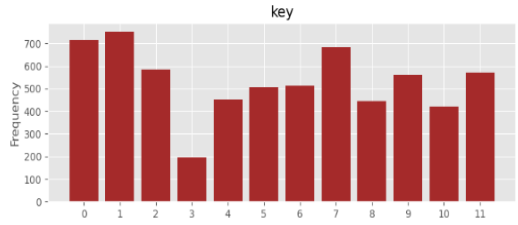
We took a look at how the popularity scores were distributed, and one thing we have noticed right away is that songs are equally between popular and unpopular data by indicating 0 or 1. In fact, it appears that this dataset is quite balanced. Below every section explains about the features:

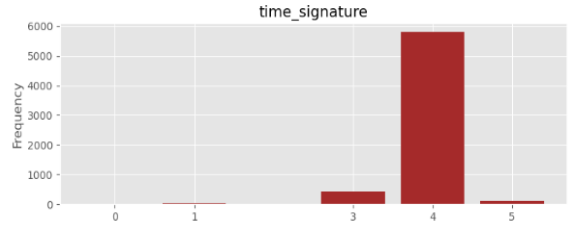


Then, we have realized that this was probably going to be a difficult problem for linear regression to solve, simply since many of the features appear to have much correlation with the target variable as per above correlation diagram. The above correlation heatmap helps us to understand that the correlation among the independent variables and target variable. We can be able to see that danceability is highly correlated with valence and tempo is highly correlated with energy. It indicates that if the tempo is higher, then the energy will be higher.







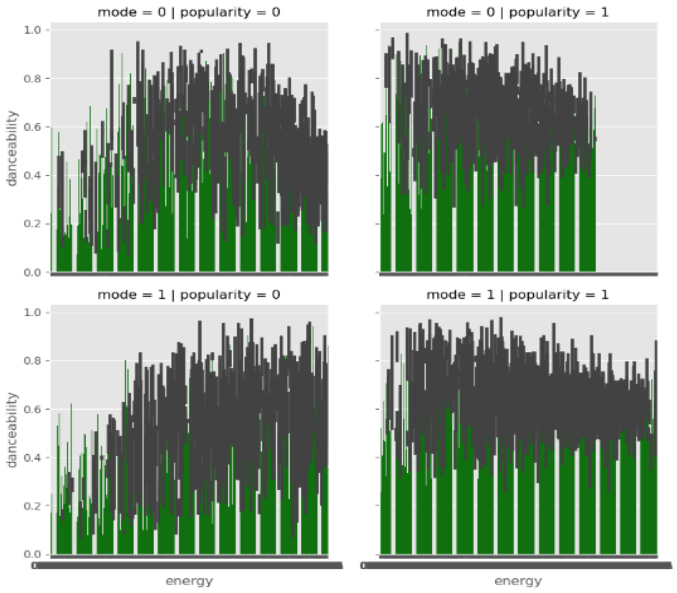


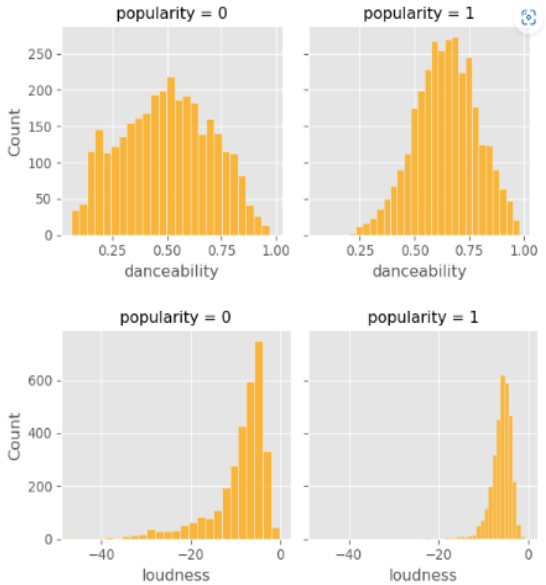
While analysing the categorial data, we have noticed that we collected more songs which are having high mode value. The graph depicts that we have collected equal amount of hit and non-hit songs which will help us to train the model more effectively. The key is evenly distributed in the dataset as we can be able to observe that data with all key values are evenly distributed. The above graphs helped us to understand the data distributed among the features which will be helpful to identify the features which are highly correlated with the target variable.

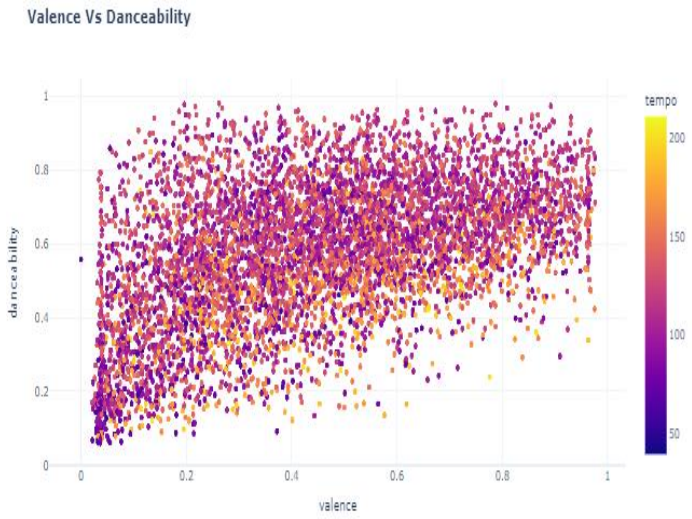
The below graphs are visualized based on mode and popularity with danceability and energy in four cases like mode and popularity (0,0), (0,1), (1,0), (1,1). The data visualization helps us to understand that the songs are popular when the energy and danceability are higher which indicates that the songs with higher energy and danceability are having high chances of becoming hit.

The Histographs below indicates that the songs with low loudness are having good chances of becoming hit.

The seaborn graph between valency and danceability indicates that the valency is directly proportional to danceability as valency is an indicator of happy and sad songs in which higher valency is a happy and the song with lower valency is sad song.







We have figured the numerical data based on above barplot map that the feature distribution within the dataset. After performing the feature validation and feature engineering that we started doing data splitting in training and testing which is one of the famous techniques to evaluate the model without any external data involvement. The splitting can be done in many ways as we are able to split the data into certain ratios which helps us to figure out the best model training and validation.

Splitting and scaling the data:

Preparing the dataset by dropping the track, artist, uri & target then defining the x and y for creating the model and then splitting the dataset into training set into 0.7 and testing set into 0.3 by taking random state is 42 and shuffle and our data is too good to get scaled in order to fit in the model as we got well cleaned and processed data.

MODELS & ALGORITHMS:

We have used the below seven algorithms for data analysis and trained them for getting the score of each model:

- Logistic Regression: It model predicts a dependent data variable by analysing the relationship between one or more existing independent variables. For e.g. if it could be used whether a political candidate will win or lose an election.
- Decision Tree: The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The deeper the tree, the more complex the decision rules and the fitter the model.
- K-Nearest Neighbors Classifier: It is computed from a simple majority vote of the nearest neighbors of each point a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.
- Support Vector Machine (Linear & RBF Kernel): The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.
- Neural Network: It is a find the most optimal hyper-plane that separates the data into two distinct classes.

- Random Forest Classifier: It is a meta estimator that fits a number of decision tree classifiers on various sub samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- Adaboost Classifier: Builds a strong classifier by combining multiple poorly performing classifier so that you will get high accuracy strong classifier.

```
Linear Regression trained.
Logistic Regression trained.
K-Nearest Neighbors trained.
Decision Tree trained.
Support Vector Machine (Linear Kernel) trained.
Support Vector Machine (RBF Kernel) trained.
Neural Network trained.
Random Forest trained.
Gradient Boosting trained.
```

```
Linear Regression: 34.64%
Logistic Regression: 64.11%
K-Nearest Neighbors: 63.07%
Decision Tree: 77.24%
Support Vector Machine (Linear Kernel): 50.31%
Support Vector Machine (RBF Kernel): 64.79%
Neural Network: 49.69%
Random Forest: 84.17%
Gradient Boosting: 83.39%
```

	Linear Regression	Logistic Regression	K-Nearest Neighbors	Decision Tree	Support Vector Machine (Linear Kernel)	Support Vector Machine (RBF Kernel)	Neural Network	Random Forest	Gradient Boosting
MAE	0.357	0.359	0.369	0.215	0.497	0.352	0.503	0.163	0.166
RMSE	0.404	0.599	0.608	0.464	0.705	0.593	0.709	0.404	0.407

Through the results of module selection, Random Forest has high score percentage. So, we went with Random Forest Classifier. Even without tuning the model, we can extract best accuracy among other models. Here, we tried to hyperparameter tuning in which we considered required number decision trees and gave entropy as criterion.

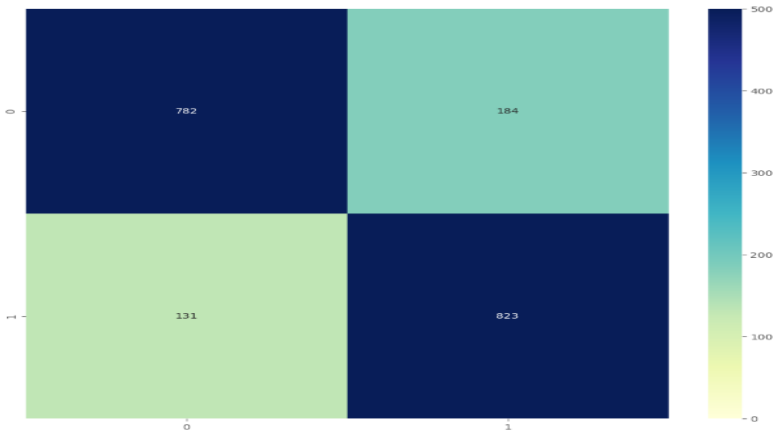
MODEL EVALUATION:

We focused mainly on the accuracy of results, based on below values of metrics like Confusion matrix and Classification Report then decided that the precision and recall were acceptable at 87% and 89%. The confusion matrix on the validation set shows that there are some false positives and false negatives, and we are successfully extracted the true positives and true negatives into the confusion matrix without any deviations which depicts that our model is well trained and modelled for predictions.

```
Confusion matrix:
 [[782 184]
 [131 823]]
Classification report:
              precision    recall  f1-score   support

           0       0.86      0.81      0.83       966
           1       0.82      0.86      0.84       954

    accuracy                           0.84      1920
   macro avg       0.84      0.84      0.84      1920
weighted avg       0.84      0.84      0.84      1920
```



Based on above table, we can see that the confusion matrix which depicts that the data from actual and predicted are evenly distributed among false negatives and true positives which indicates that the model is well prepared and evaluated which is ready for further prediction analysis.

PROJECT RESULTS:

The RF model yielded 84% accuracy on the validation data and 87.7% accuracy on the test data, with similar result on the training data indicating no over-fitting and after hyperparameter tuning, we got much better accuracy.

PROJECT MILESTONES:

Data collection and cleaning: Gathering and cleaning the project's essential data, including historical streaming information, artist and track information, and user behavior information.

Feature Engineering: Finding pertinent elements that may affect a song's success, such as tempo, energy, danceability, and singer popularity.

Model selection and training: Selecting appropriate machine learning algorithms for the project, such as random forests or neural networks, and training them on the collected data.
Model evaluation: Evaluating the performance of the trained models using appropriate metrics, such as accuracy or area under the ROC curve.

Deployment: Integrating the trained model into the Spotify platform and deploying it to make predictions on new songs.

Continuous improvement: Monitoring the performance of the model in production and continuously improving it based on new data

REPOSITORY / ARCHIVE:

Github Link:

https://github.com/veepura123/Song-popularity

PROJECT REQUIREMENTS:

- Operating Systems: Windows.
- Used LUCID chart for Flow Chart diagrams.
- Uses of Applications: Microsoft Word, Microsoft Excel, and Microsoft PowerPoint.
- Libraries: Pandas, Numpy, seaborn, Matplotlib, Sckit-Learn, Pickle, Gradio for UI development.

REFERENCES:

- https://dorazaria.github.io/machinelearning/spotify-popularity-prediction/#52-nearest-neighbors-classifier
- https://www.kaggle.com/code/pelinsoylu/spotify-popularity-prediction-ml-practice
- https://medium.com/@shgo5289/spotify-data-analysis-through-power-bi-7cb3099ac922
- https://blog.devgenius.io/mini-ml-project-predicting-spotify-songs-popularity-part-1-ec1c906b8ff8

From above references, we have improvised and changed the dataset. Mainly, changed the popularity target into 0 or 1 based on the percentage. Besides that, we have used classification instead of regression because the target data is encoded which supports for classification over regression.

CONCLUSION & FUTURE WORK:

As per our evaluation, we have seen that Adaboost and Random Forest outperform LR and NN regarding accuracy. The most robust model is the RF. We have faced some challenges like which model should we use, converting the popularity, model selection and loads of errors. As this the Learning process, we learnt a lot from the project itself. This project based on classification model which helps to understand how the regression and classification are differ and how to act accordingly.

In future experiments we would like to investigate label influence and social media presence with respect to song success. And we would be thinking to have a tool in the recording studio that would give artists the ability to gauge the potential popularity of their new material by creating a feedback loop.

Source Code:

https://colab.research.google.com/drive/1-KZ-6nEFde3O2eB86W48IxP0jFkaabWz

*****The End*****