CSE343/ECE343 — Machine Learning Monsoon 2021
Track-Hit : Music Hit Predictor

Akhilesh Reddy
2019230
IIIT-Delhi
akhilesh19230@iiitd.ac.in

Gitansh Raj Satija
2019241
IIIT-Delhi
gitansh19241@iiitd.ac.in

Vinay Pandey
2019288
IIIT-Delhi
vinay19288@iiitd.ac.in

Yatharth Taneja
2019346
IIIT-Delhi
yatharth19346@iiitd.ac.in

Abstract

Music is a great stress buster, helps in relaxation, and elevates our mood. Songs have different lyrics and sentiments attached, unique audio features which leave an impact on the audience. While some songs become a sensation, some fail to leave their mark. With so many new artists coming up, releasing a top-of-the-chart song is all the more difficult. There have been different studies to predict the success of a song based on its audio features, lyrics, and emotion. Some researchers have also tried to study the artist's impact, releasing date, and other metadata surrounding the song. In this project, we aim to study the impact of acoustic features in predicting a song's popularity without any information on the artist. We also present a new dataset containing information only on Indian Songs; analyze the difference in the impact of acoustic features on song popularity in India as opposed to globally.

1. Introduction

Acoustic features of audio are the physical features that can be recorded and analyzed. They include but are not limited to loudness, valence, tempo, and danceability. Many studies have found a correlation between these acoustic features and song popularity [1]. This project proposes classifying songs as Successful (Hit Song) or Unsuccessful (Unpopular Song) based mainly on acoustic features. Past studies show that the artist's information has a significant effect on song popularity [1]. However, our focus is only to analyze and study the impact of acoustic features on song popularity when the song's artist is not known. This will be helpful, especially for newer artists who do not have any previous hits and want to predict the success of a song based on its physical characteristics. The model can also be helpful for music producers and investors looking to find potential talent.

We hereby put forward the idea of binary classification models to classify songs as successful or unsuccessful. Additionally, we look for a way to allocate a popularity score using regression models. We aim to find the suitable features that affect popularity using different feature learning techniques; test and compare multiple machine learning models for our proposed task.

2. Literature Survey

In this section, we go through some of the work done in predicting song popularity.

The paper Dance Hit Song Prediction [2] explores the possibility of predicting dance hit songs by using basic musical features and capturing advanced features that model the temporal aspects. The research in the paper categorized different characteristics of a song into three different features, namely, meta-information(like artist name, artist location), basic analyzer features(like acoustics, danceability, energy), and temporal features (timbre and beat difference). A total of five models, namely, Decision Trees, Ripper Ruleset, Naive Bayes, Logistic Regression, and Support Vector Machines, were built for each dataset using diverse classification techniques and tested through 10-fold cross-validation.

Musical trends and predictability of success in contemporary songs in and out of the top charts [1] takes into account the acoustic features as well as the correlation between the songs success and the artists associated by considering songs released in the UK between 1985 and 2015 to understand the dynamics of success and study various multi-decadal trends. A random forest has been used to predict success. The paper deals with both the 'musical' factors and the 'socio-economic' factors such as popularity ('superstar') status and the demographics of the artist(s), the level of promotion, the label/company and other such meta-information, and their correlation with the song's success.

Hit Song Prediction for Pop Music By SIAMESE CNN with ranking loss [3] explores predicting the popularity via training the model based on hit scores and relative ranking

through a convolutional neural network instead of regression/classification models. Using euclidean loss and pairwise ranking loss, they were able to devise relative ranking among songs, and by using a sampling method called A/B, they were able to predict results with high accuracy. The Siamese architecture optimizes the model parameters by jointly considering rating and ranking-based loss functions. The proposed model is a multi-objective Siamese CNN which optimizes both the mean squared error (MSE) in predicting (i.e. rating) the hit song scores for both songs and the pairwise ranking loss in deciding which one of the two is likely to have a higher score.

3. Dataset

3.1. Data Description

The dataset contains some metadata and various audio features about the songs. The Meta-Informative features include track id, track name, artist name and album name. The Acoustic features include the measure of acousticness, dancebility, energy, loudness among others.

Feature Name	Datatype Name
Song ID	String
Song Name	String
Artist Name	String
Album Name	String
Release Date	Integer
Acousticness	Float
Danceability	Float
Song Duration	Integer
Energy	Float
Instrumentalness	Float
Key	Integer
Liveliness	Float
Loudness	Float
Mode	Integer
Speechiness	Float
Tempo	Float
Time Signature	Integer
Valence	Float
Popularity	Integer

Table 1: Description of Dataset.

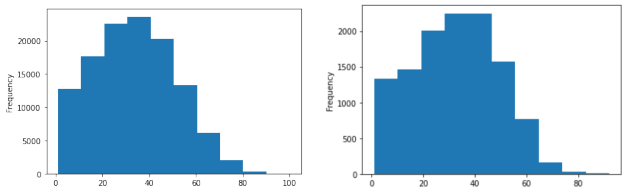
3.2. Data Extraction

We used the Spotipy library to access the Spotify API for song information retrieval. We obtained information of 70,000+ songs using different queries based on genre,year and with the help of different public playlists. However, as Spotify has restricted the offset size to 1000 recently, we were not able to extract more songs using a query. Hence, we decided to merge our dataset with the publicly available dataset [4].

3.3. Preprocessing

For preparing the dataset for the model, we removed all the features(fields) which were unique to the song (such as Song ID, Song Name) and information related to the artist. We initially gathered a dataset of 300,000+ entries however, there were many duplicates. We dropped all the duplicate entries using the pandas library. Also, we analyzed the dataset and realized a huge chunk (around 30%) of the songs in the dataset had 0 popularity. This was because either the songs were released by unknown artists and were never publicised or due to insufficient data. To ensure a normal distribution of our dataset, we decided to drop these songs. Further, we visualized the distribution in popularity of songs against all the features and removed all the extreme outliers like songs of duration more than 8 minutes. We decided to consider only the 'year' parameter of release data in order to avoid complexity while training. We normalized the audio features using min-max normalization on the column. This normalization technique is quite useful to avoid the biases of certain features with respect to others during model training. We then used this normalized data for feature extraction and model training. After the preprocessing stage, we were left with 118,795 different songs which were composed by 14,222 different artists.

3.4. Data Visualization



(a) Global Dataset (b) Indian Dataset

Figure 1: Popularity vs Frequency

The above plots show the frequency distribution of songs based on their popularity. It can be observed that the highest frequency is of the songs that have a popularity in the range from 30-50. We also observe that both the global and Indian data set are almost normally distributed which represents a good dataset.

3.4.1 Indian Dataset

For our analysis on Indian Songs, we collected data on songs made by Indian Artists by querying track information based on 'Bollywood', 'India Indie', 'Tollywood' genres and by fetching online playlists consisting of songs by

Indian Artists. We also collected data from our original dataset, whose artists were Indian. Following preprocessing techniques similar to our original dataset, we were able to extract 11,844 different songs by 1,394 different artists

3.5. Feature Selection

As the dataset had a lower dimension, we decided to perform feature selection instead of feature extraction. We used the following methods to distinguish which features greatly affected the predictions of the model and which features did not play a great role and could be dropped from the dataset.

3.5.1 Fisher's Score

It is a supervised algorithm that returns the rank of each feature based on the fisher's score. This rank can be used for feature selection among different variables. The higher the rank of the variable, the more useful is the feature in predicting the target variable.

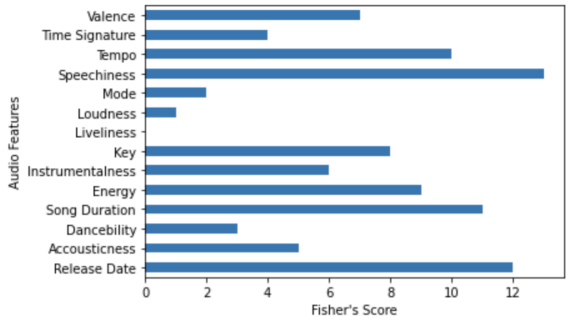


Figure 2: Fisher's Score

3.5.2 Information Gain

Information gain for each variable is calculated in the context of the target variable and is used for feature selection. It is calculated by subtracting the weighted entropy for each variable from the original entropy. The higher the information gain, the greater is the decrease in entropy.

3.5.3 Correlation Coefficient

It is a measure of the linear relationship between 2 or more variables. It helps in predicting a variable based on the value of another variable. It helps in deciding the features which are largely correlated with each other and can be dropped after determining their correlation with the target variable and therefore help in feature selection.

3.5.4 Inference

Therefore, after observing the results of all the above techniques we arrived at the conclusion to drop the features

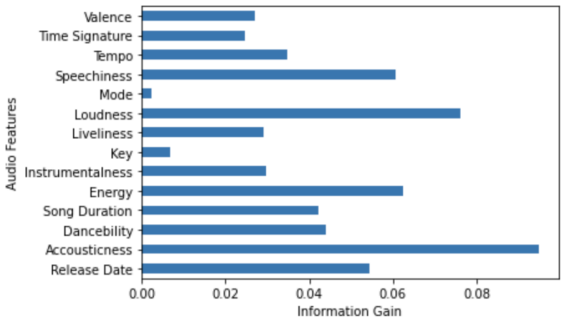


Figure 3: Information Gain

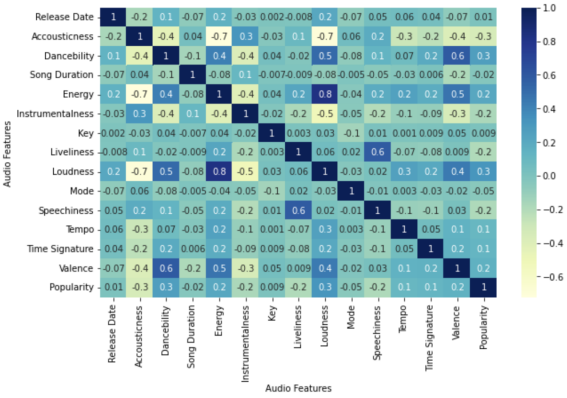


Figure 4: Correlation Coefficient

'Mode', 'Key' and 'Time Signature'. The reason is that the features 'Mode', 'Key' and 'Time Signature' are providing relatively lower information as compared to other features[Figure 2 (3.5.2)]. Also, it can be observed that the Fischer score value for 'Mode' and 'Time Signature' is relatively low. It can be also seen that a relatively lower value of Fischer score is obtained for the features 'Liveliness' and 'Loudness' but one of the reasons for this could be that as the computational complexity for Fischer score is relatively high for large amount of data, therefore we performed the Fischer score technique on a smaller subset of data. Since, in all the other techniques the features 'Loudness' and 'Liveliness' provides a good amount of information, therefore we decided not to drop these features.

3.6. Final Data

The data used for the regression and classification problems are thus different. The regression models use a custom data(extracted from spotipy) after the feature selection process which has a 'Popularity' attribute which refers to a popularity score. On the other hand, the classification data make use of a dataset from Github [5] which have a 'target' attribute signifying hit/loss.

4. Methodology

The main objective is to classify songs into popular or unpopular sets. The secondary objective is to also assign a popularity score to songs on the basis of musical factors. The initial steps involve preprocessing the data using data visualization and feature selection techniques.After being done with the preprocessing and normalization of the data, the data is free of useless parameters. After the preprocessing stage, different models have been tried for both classification and regression. In our preliminary dataset which is being used for regression based tasks we use a 8:2 test train split with 95036 training data size and 23759 testing data size.

4.1. Classification

We plan to apply binary classification to segregate data into popular and non-popular songs.We use a certain threshold value (45) for binarizing the popular/non-popular songs in the dataset. The median of popularity lies at 33.3 and observes a peak frequecncy at around 40. Thus we hypothesize that a song becomes popular on receiving a score of more than 45. We have used logistic regression,Gaussian Naive Bayes, K-Nearest Neighbours, Decision Tree, Random Forest, SVM, Neural Network and Gradient Boosting Classifiers in the project. The classification models trains on the training data and predicts whether a song could become popular or not. This prediction is made on the basis of the features of the songs.

4.2. Regression

Regression techniques have also been planned on the data to assign a popularity score to songs on the basis of musical factors. As of now, Linear regression has been performed, and both the regularisation techniques i.e. Lasso and Ridge have been used additionally. More models will also be used later on in the project.

5. Results and Analysis

We have received satisfying results comparable to state of art models in the preliminary models that we have trained. The result and the analysis can be seen below:

5.1. Binary Classification

In conclusion, SVM, Neural Network and the Random Forest classifiers perform best and the random forest model has a slightly better accuracy than the other two.

Classification Models	**Precision**	**Recall**	**F1**	**Accuracy**
Logistic Regression	0.93	0.73	0.80	0.73
Gaussian Naive Bayes	0.66	0.63	0.64	0.6
K-Nearest Neighbors	0.76	0.75	0.75	0.75
Decision Tree	0.73	0.73	0.73	0.73
Random Forest	**0.81**	**0.81**	**0.81**	**0.81**
SVM	0.81	0.80	0.80	0.80
Neural Network	0.81	0.80	0.80	0.80
Gradient Boosting	0.80	0.80	0.80	0.80

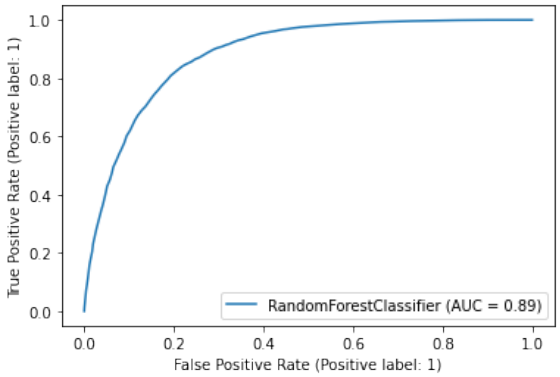


Figure 5: ROC curve for Random Forest

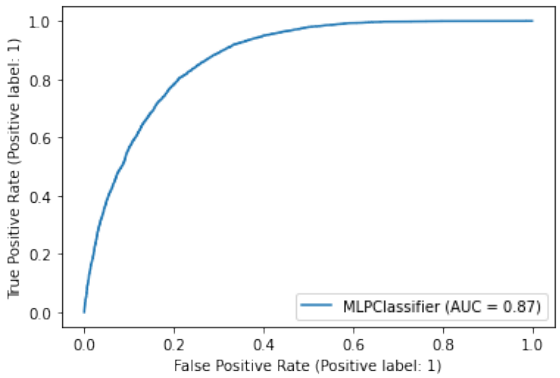


Figure 6: ROC curve for Neural Network

5.2. Popularity Score Prediction

We trained several regression models to predict the popularity score for a song. These models include Linear Regression, Ridge Regression, XGB Regression among others. The root mean square error obtained across all the model was in the range of 14.37-16.05. The regression models trains on the training set and tries to establish a relationship between the features of a song and their popularity. The model then predicts a popularity score on the testing

data. We observe that Random Forest Regressor gives us the least RMSE (best performance) of 14.376. We selected this model for hyperparameter tuning by taking number of estimators and max depth of each tree as parameters. We found the best performance to be at max depth=14 with 200 estimators.

Model	Test RMSE
Linear Regression	15.934903
Ridge Regression	15.934899
Lasso Regression	16.051243
XGBRegressor	14.774658
RandomForestRegressor	**14.376308**
AdaBoostRegressor	15.626482
DecisionTreeRegressor	15.073415

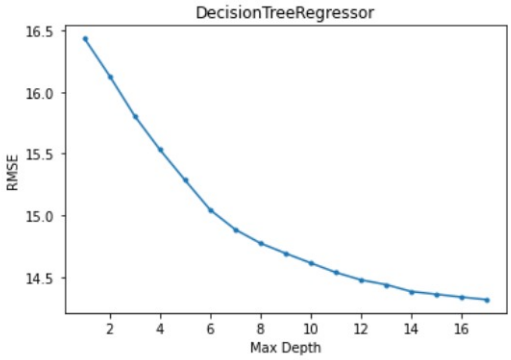


Figure 7: RMSE for Different Max Depths

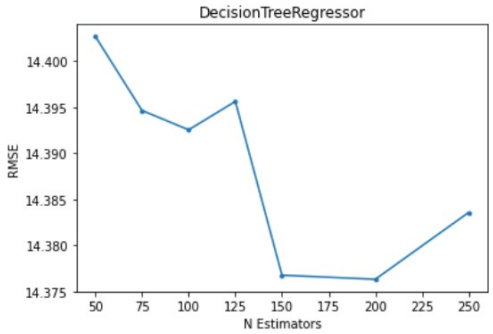


Figure 8: RMSE for Different Max Depths = 14 and diff value of N estimators

6. Conclusion

From our results we observe that the performance of our models is at par with the current work in this field.

6.1. Outcomes

After evaluating the results from the models, our conjecture that only Acoustic features can also have a great impact on the popularity of a song and can predict its success with a decent accuracy without any bias of artist and production house details.
Also, we observe a stark difference in the performance of model trained on our global dataset when we tested it on Indian Song Dataset. We predict popularity scores on our best Regression Model (Random Forest Regressor) for the Indian Songs and observed an RMSE of 16.29 (considerably higher than that of Global Dataset). On training a separate model using Indian Dataset Only, we get an RMSE of 14.34 indicating that features have different importance in the Indian Dataset. Thus we conclude that on analyzing indigenous songs we can have better predictions.

6.2. Learnings from the project

This project gave us insights on many aspects of designing a pipeline for a machine learning project. We realized that collecting accurate data is as important as building the final machine learning algorithm/model. We went ahead and introduced ourselves to Spotify API, using which we extracted the data and prepared the dataset on our own. We became acquainted with different tools for data visualization, which helped us analyze different trends in our data. Through this project, we also became familiar with different feature selection techniques like fisher score, chi-square test, and others that are not covered in our course. As we are training different models to compare their performance for our task, we continue to learn the in-depth working of models and will also look at how to improve their performances in the future.

6.3. Discussion

We have been able to follow the tentative timeline we had proposed. At the end of the final week, we have tested some preliminary models and received a satisfying result. As mentioned in our timeline, we trained several classification and regression models. The classification models predicts whether a song could become popular or not whereas the regression models assigns a popularity score to the songs. The future work which could be possible with this project involves the relation between the album names and popularity of artists in making a song popular. There are several other meta-informative features which makes a song popular but for the sake of simplicity we have only discussed the relation between the audio features and the popularity of a song. We could also extend this work to take into consideration a smaller subset of songs such as bollywood songs or late-90s songs to establish more accurate results.

6.4. Member Contribution

- **Akhilesh Reddy**: Literature review, Data Extraction and Collection by using available datasets online, Applying models Linear regression , Logistic Regression, Ridge and Lasso regularisation , Decision Tree, Random Forest,K-Nearest Neighbours, and Analysis and inference of the data
- **Gitansh Raj Satija**: Literature review, Data Extraction and Collection by fetching various playlists online, Indian Dataset Generation, Data cleaning and Preprocesssing, and Analysis and inference of the data, Gaussian NB, Regressor Models, Indian Dataset Analysis
- **Vinay Pandey**: Literature review, Data Extraction and Collection : Using spotify api based on year, Feature selection using Information gain , correlation coefficient and mean absolute difference, and Analysis and inference of the data
- **Yatharth Taneja**: Literature review, Data Extraction and Collection : Using spotify api based on genre , Feature selection using l1 regularisation, fisher index, Information gain , correlation coefficient , Applying models like Neural Network, SVM,Gradient Boosting, and Analysis and inference of the data

References

[1] K. K. Interiano M, Y. J. Wang L, and Y. Z, "Musical trends and predictability of success in contemporary songs in and out of the top charts," *The Royal Society*, 2018. 1

[2] K. S. Dorien herremans, David Martens, "Dance hit song prediction," *Journal of New music Research*, 2014. 1

[3] L.-C. Yu, Y.-H. Y. Yi-An Chen, and Y.-N. Hung, "Hit song prediction for pop music by siamese cnn," *IEEE,International Conference on Acoustics, Speech, and Signal Processing*, 2017. 1

[4] Z. Hamidani, "Spotify tracks db," https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db. 2

[5] F. Ansari, "The-spotify-hit-predictor-dataset," 2020. 3