Song Popularity Prediction

Vaishnavi Mocherla
Masters in Data Analytics
San Jose State University
San Jose, USA
vaishnavi.mocherla@sjsu.edu

Vijay Rama Raju Penmatsa
Masters in Data Analytics
San Jose State University
San Jose, USA
vijayramaraju.penmatsa@sjsu.edu

Sri Mounika Jammalamadaka
Masters in Data Analytics
San Jose State University
San Jose, USA
srimounika.jammalamadaka@sjsu.edu

Nikhil Gudur
Masters in Data Analytics
San Jose State University
San Jose, USA
nikhilsarma.gudur@sjsu.edu

***Abstract*—In today's fast-paced generation, songs are less determined by music and the instruments used and more determined by the popularity it has gained over time. A few songs of artists hitting chart busters constantly have made us wonder if popularity depends on any combination or correlation. In this project, we intend to develop a predictive model that forecasts the popularity of the song by leveraging the Kaggle dataset. We look forward to building a predictive model using the dataset which comprises historic data. We would like to analyze how music has changed and evolved and based on this we will predict the future using time series to get to know what music might be popular. The machine learning models we would like to use are Decision Trees, Random Forest, SVM, Logistic Regression, and ensemble methods which can handle the complexity and variability inherent in the data.**

***Index Terms*—Support Vector Machine, One-hot Encoding, Ensemble Methods, Hyper Parameter Tuning, Prediction**

I. Introduction

In the current trends of the music industry, popularity and the reach play a vital role in the success of a song. The key features like popularity alongside of other critical features like the duration, artist background, different keys involved in a song are some of the metrics which piqued our interest to explore this topic further. Given the comprehensive data available on the Kaggle dataset which is given by the one of the leading digital music service that has access to millions of songs, it captures a wide array of historic data which helps us understand and helps us construct a predictive framework that can forecast a song's popularity. With the help os the comprehensive dataset and the intent to build the machine learning models such as Decision Trees, Random Forest, SVM, Logistic Regression, and ensemble methods which can handle the complexity and variability inherent in the data, we aim to build a song popularity prediction.

II. Motivation

A. Motivation driven for the project

Our motivation for this project is mainly that the popularity of the song is not entirely random but might be influenced due to various external factors. By understanding the underlying patterns and correlation between various columns of the dataset we intend to create an algorithm that can be used by streaming services to improve their recommendation algorithms which will enhance the user experience, offering a more personalized listening experience. By comparing the models based on historical data we hope to generate new trends in the music choices, the preferences that impact various factors of a song's popularity and song's rating. As mentioned, we intend to use machine learning models such as Decision Trees, Random Forest, SVM, Logistic Regression, and ensemble methods which can handle the complexity and variability inherent in the data. Based on this project we look forward to delivering a significant impact to budding artists and producers, streaming services, and music industry analysts. This project helps in the field of ensuring sustainable consumption and production patterns. This benefits the music industry in creating and producing sustainable, correct and likeable music that are predicted to be a hit!

III. Literature Survey

In the music industry, a lot of researchers have taken interest in applying machine learning methods to predict how popular songs can be. In solving this problem, different approaches and methodologies have been used by several studies which offer distinct reflections as well as restrictions. The audio features, lyrics and popularity of songs were considered in relation to one another by a certain study that suggests a model based on machine learning for predicting these outcomes [1]. This is done with an aim of forecasting commercial success through integration of many musical aspects. In their work titled "HIT SONG PREDICTION FOR POP MUSIC BY SIAMESE CNN WITH RANKING LOSS," the authors focus on pop music using Siamese Convolutional Neural Network (CNN) along with ranking loss function. They explore how well CNN models can identify potential hits within limited genres but with emphasis being put on genre specific features [2]. Another important contribution made was "Machine Learning and Chord Based Feature Engineering for Genre Prediction in Popular Brazilian

Music" which centers around chord analysis as a means to predict genres in Brazilian music. It points out that the dataset used was too specific thus limiting generalization of results to other

IV. Methodology

The initial phase of the project comprises of Data Collection, Data Prepossessing, feature and model selection. To address our problem statement we have chosen the following machine learning models, SVM, Logistic regression, decision tree, Random forest ADA Boost, XGBoost and Perceptron Algorithms, different phases of methodology are explained in a detailed structure below

V. Implementation

Songs and albums are released in thousands every year but most of them go unrecognized and very few of them get popular. Analyzing the song patterns can give us some insights about them for artists and genres also. Using all these features of artists, their genres and songs produced to predict their popularity is the main goal. This can also help in gaining insights into what are the important features of a song that contribute to its popularity and audience preference.

A. Introduction to the Dataset

The dataset chosen for this project is the Kaggle dataset which includes a wide range of features; the Kaggle dataset is divided into 4 significant datasets, which comprise the information about different genres, artists, ids of different tracks, the release or the re-release years, various other song metrics like danceability, energy, liveness, and loudness. The historical popularity scores are also denoted in the dataset. Live Data from Spotify API: We tried fetching the live data through the Spotify API; we were successful in fetching the 1000 songs within the year 2024 which was viable to integrate live data in our analysis. However, the limitation of the rate limit has hindered our progress posing a significant challenge to expand our dataset as initially planned. We have also made requests to the Spotify developer App to increase the rate limit, but the live data exploration could not be completed as the requests made were unaddressed.

B. Problem Statement

Our aim is to build and test models which predicts the song popularity based on 72 different features.

C. Data Preprocessing and EDA

During the process of Data preprocessing and cleaning, the necessary pandas, numpy and seaborn libraries are imported as the first step, loading the datasets. We have displayed the dataset information and summary statistics using the functions like info() and describe (). We have also created a correlation matrix to identify the key features and filter the pairs with the correlation. We have also used the selected types to filter the data to only numeric types. The numeric features are observed outliers and distributions using the box plot.

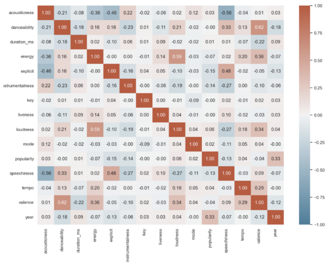


Fig. 1. Heat Map : Correlation matrix

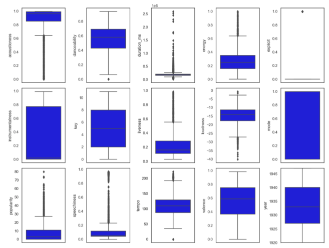


Fig. 2. Box Plot

Proceeding further with cleaning the dataset, we have considered the artists column, converted the strings, cleaned the data by removing brackets and quotes, lowercase, split commas, and exploding to separate artists, and counted the number of songs available per artist. To understand the trends over time, we have created a grid of line plots for each feature to observe the trends over time. To better understand the critical features of the song's popularity and the related metrics, we have ensured that categorical features such as 'key', 'mode', and 'year' are converted into categorical data types. We have also identified the unique artists and counted their numbers. The songs with popularity over 70 are thoroughly explored by the mean popularity by artist and year. With further exploration,, it is understood that the dataset contains 5694 unique artists; further features like key and valence are explored.

The Fig 1. depicts correlation matrix is a heatmap that helps us understand the correlation coefficients between various musical features and the year of release for songs. From the above heatmap, we can see that loudness and energy, danceability and valence have strong positive correlations, and the pairs with strong negative correlations are acousticness and energy, acousticness and loudness, and acousticness and explicit.

The boxplot represented in Fig 2. for various musical attributes is visualized. The boxplot provides insights into the central tendency, spread, and skewness of the data. The box plot is well suited to identifying the outliers. From the above boxplot, it is well understood that popularity is spread out with many outliers. This indicates that there is varied popularity across songs.

Given the histogram in Fig 3. The keys are mapped corresponding to musical key names. From the histogram, we can

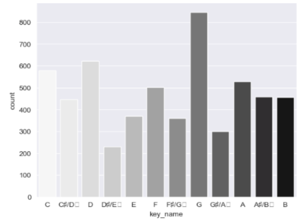


Fig. 3. Histogram depicting various keys

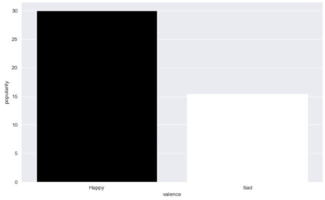


Fig. 4. Valence v/s Popularity

depict that keys like G(7) and D (2) are more frequent in the song. Given different genres of music, the popularity of keys is variable.

The Fig 4. shows that the ‘Valence’ of popular songs is higher than that of sad songs.

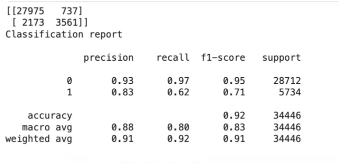
D. Dimensionality Reduction Technique

The scree plot shows that most of the variance in the dataset is captured by the initial principal components in the PCA analysis. The sharp drop seen in the graph indicates that dimensionality can be greatly decreased without sacrificing important information. Efficient data representation is crucial for improving both model performance and interpretability.

E. Machine Learning Models

For addressing this problem, we have chosen 7 different models in which we used 1 regression and 6 classification models, i.e Logistic Regression, Decision Tree, Random Forest, SVM, Ada Boost, XG Boost and Perceptron Algorithms. On comparing the results of each model manually, to get an overall understanding and clear idea we implemented all of these models and used the Ensemble method to get the final accurate score for the problem statement.

The popularity of the songs in the dataset is ranged from 0 - 100 but while performing the EDA on the data most of the songs tend to fall on the less popular side, which is mostly true in real world scenarios. To handle such imbalance, the range of the popularity is changed from 0 - 100 to 0 and 1 i.e.. Songs that fall under the 0 - 50 popularity range are considered as 0 and songs that fall under 51 - 100 range are considered as 1. But the popular songs class is in minority compared to the non popular songs. To obtain the classification accuracy metrics such as F-1 score recall and precision must be employed in order to get a robust measurement of the model’s accuracy.



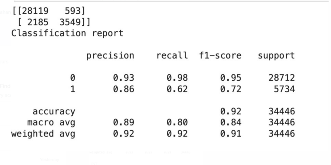
```
[[27975   737]
 [ 2173  3561]]
Classification report

              precision    recall  f1-score   support

           0       0.93      0.97      0.95     28712
           1       0.83      0.62      0.71      5734

    accuracy                           0.92     34446
   macro avg       0.88      0.80      0.83     34446
weighted avg       0.91      0.92      0.91     34446
```

Fig. 5. Classification report for Logistic Regression



```
[[28119   593]
 [ 2185  3549]]
Classification report

              precision    recall  f1-score   support

           0       0.93      0.98      0.95     28712
           1       0.86      0.62      0.72      5734

    accuracy                           0.92     34446
   macro avg       0.89      0.80      0.84     34446
weighted avg       0.92      0.92      0.91     34446
```

Fig. 6. Classification report for Decision Tree

F. Logistic Regression

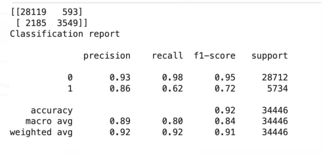
The test accuracy of the logistic regression model, which was optimized using grid search, achieved 92%. The confusion matrix and classification report show excellent results, especially for the dominant class (0), achieving a precision of 0.93 and a recall of 0.97. Nonetheless, the results for the minority class (1) were mediocre, with a recall of 0.62 and an F1-score of 0.71, highlighting the opportunity for enhancing the classification of positive cases.

G. Decision Tree

The Gini used instead of entropy to get slightly better accuracy and this is concluded after grid search through 108 epochs.

H. Random Forest Classifier

They can be used for both classification and regression analysis but here they were employed for classification to train the model. The change in performance was very little upon grid search and was not satisfactory, which also led to dropping this technique in the further ensemble learning.



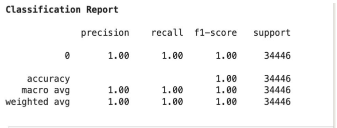
```
[[28119   593]
 [ 2185  3549]]
Classification report

              precision    recall  f1-score   support

           0       0.93      0.98      0.95     28712
           1       0.86      0.62      0.72      5734

    accuracy                           0.92     34446
   macro avg       0.89      0.80      0.84     34446
weighted avg       0.92      0.92      0.91     34446
```

Fig. 7. Classification report for ADABoost

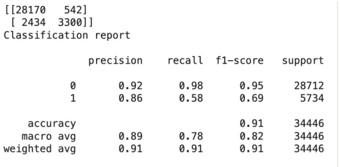


```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     34446

    accuracy                           1.00     34446
   macro avg       1.00      1.00      1.00     34446
weighted avg       1.00      1.00      1.00     34446
```

Fig. 8. Classification Report for Random Forest



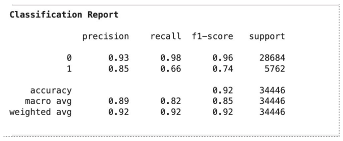
```
[[28170   542]
 [ 2434  3300]]
Classification report

              precision    recall  f1-score   support

           0       0.92      0.98      0.95     28712
           1       0.86      0.58      0.69      5734

    accuracy                           0.91     34446
   macro avg       0.89      0.78      0.82     34446
weighted avg       0.91      0.91      0.91     34446
```

Fig. 9. Classification report for SVM



```
Classification Report

              precision    recall  f1-score   support

           0       0.93      0.98      0.96     28684
           1       0.85      0.66      0.74      5762

    accuracy                           0.92     34446
   macro avg       0.89      0.82      0.85     34446
weighted avg       0.92      0.92      0.92     34446
```

Fig. 10. Classification report for XGBoost

I. Support Vector Machine

While experimenting it is observed that the SVM can generalize better. The kernel trick is also applied to exploit the math in order to map the data in higher dimensions. The regularization parameters (C) were also used in grid search to get the best F-1 score.

J. Ada Boost

In the AdaBoost classifier model tuned with grid search, the top parameter combination resulted in a remarkable cross-validation accuracy of 92% and a corresponding test accuracy of 92% as well. The analysis of the ROC curve indicated a discriminative ability between the classes, with an AUC of 0.92, showing a high level of performance. The classification report notes that the model's accuracy in classifying both the majority and minority classes is supported by the precision, recall, and F1-score.

K. XG Boost

The XGBoost model was fine-tuned by experimenting with grid search. The search is made with learning rates [0.01, 0.1, 0.2], maximum depths of [3, 5, 7] and n_estimators [50, 100, 200] to yield the best F-1 score. Additionally, they were further fine tuned to improve generalization.

L. Perceptron

The perceptron's learning is tuned through a number of epochs. Once the learning is not significantly increasing upon further epochs, they were stopped. # the best : {'eta0': 0.01, 'max_iter': 10000}

M. Ensemble methods

Ensemble learning involves the use of multiple learning algorithms to improve performance. We limit our scope to classification problem using tree-based algorithms, except for voting classifiers where multiple algorithms need to be used.

- Voting classifiers take a few classifier's decisions to aggregate them and predict a final decision. Often a classifiers' individual classification is much weaker than a voting classifier's that is the aggregate of all of them.
- We use hard voting and soft voting classifiers and use all the classifiers with the hyperparameters that we have tuned before.
- We use pipelines to assemble several steps that can be crossvalidated together while setting different parameters for each of the different classifiers.
- Hard Voting Classifier: We use SVM, Logistic Regressor, Decision Tree and Perceptron as the voters of this classifier. The class with majority votes from all these classifiers are predicted by the voting classifier.
- Soft Voting Classifier: We use all the classifiers except Perceptron as the voters in this, as Perceptron does not give class probabilities. These voters' decisions are pipelined into the voting classifier weighted by the predicted class probabilities. This classifier is expected to give better results as the more confident votes are given higher weightage

VI. Conclusion

With this project, we have successfully predicted the popularity of of the song with the given machine learning models, however in the implementation of this project we have touched upon various areas which would still need room for improvement such as Deciding upon a classification or a regression problem : While deciding between a classification problem or regression problem it is important to analyze the kind of target variable that to be dealt with. In this project the popularity of a song is ranged from 0 - 100, and considering distribution of the dataset and the imbalance that is present in it, (very less high popular songs) the regression eligible problem statement is converted into a classification one, by categorizing it, which also significantly reduced the imbalance while using SVM and Decision Trees.Evaluation metrics : It is learned that R square evaluation metric, that is mostly used to get the performance of a model can not be used to get the performance of any model, because classification and regression are two different kinds of problems and models that deal with each of them should evaluated differently when that is not done properly it leads to false assumptions and conclusions. Feature engineering : Not all the features that are present in the dataset should be used to train the model, through these learnings, implementations of various models and visualizations of this project we uncover a valuable resource for other researchers and practitioners.

VII. APPENDIX

A. Code Walkthrough

The code walkthrough for this project is simple, the code have been pushed into the github repository and is arranged in a very neat and co-ordibated manner. Each files and the notebooks are sorted in folders making it easy to access the files and understand the project.

B. Presentation Skills

Our presentation capabilities have been honed, allowing us to articulate our ideas and the scope and findings of our project with precision and engagement.

C. Presentation Skills

We have made sure to articulate various steps taken in every phase of the project to be incorporated in the presentation slides. Make sure to add the highlights of every step of the project to help our peers understand key takeaways and milestones.

D. Discussion / Q&A

Ensured to have pitch meetings and timed demo presentations to help us understand the flow of the presentation making sure we give adequate time to address the questions and project related queries. We believe this as a crucial step for learning as it helps us reflect on the project goals and encourages us to be open to constructive criticism

E. Demo

We have ensured to run our machine learning models, with improved performance and the code files are uploaded to our Git repository to ensure the possibility of quick demo. All the files are uploaded post making sure they are run in the local environment which helps in effective code readability.

F. Visualization

The visualization have played a major role in understanding how the data is distributed. With the help of visualizations like box plot, the outliers are determined within the huge amount of data.

G. Report

We have ensured to detail every step of the project in the report making sure we adhere to academic integrity. The report is written in IEEE format which is the most used report format by students and young professionals

H. Version Control

Version Control : To ensure version control we have created a github repository solely to collaborate and maintain up to date code changes made as a part of the project. The responsibility of maintaining the git repository is distributed among all the team members.

I. Relates to sustainability

This projects relates to sustainability, accurately predicting the songs popularity helps in promoting emerging artists which can be easily overlooked. This supports cultural diversity without focusing only on established artists. This project also focuses on the more predictable songs popularity is, the more sustainable the industry becomes. This helps the artists concentrate on patterns rather than experiments.

J. Lessons learned

Various insights gained while building this project is documented, the challenges faced while fetching in the data and other phases of the project were solved strategically by brainstorming sessions and collective efforts within the team. The brainstorming sessions to overcome these challenges posed during the project have offered valuable lessons which will be a primary point of consideration for future projects/works.

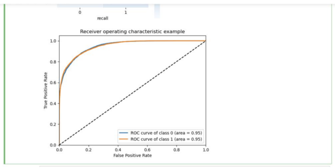


Fig. 11. ROC of Xgb

K. Prospects of winning competition / Publication

The project on Song Popularity prediction is most suited for competing and publishing given it's comprehensive data and the various machine learning algorithms. The focus of this project to help emerging artists and improve the recommendation system poses a great idea in the real world scenario. Considering the project follows a data driven approach and interdisciplinary approach we strongly believe that our project is well suited and has higher potential of wining competitions and publications

L. Innovation

Our project uses advanced machine learning models such ADA Boost, XGboost. We have used Hyper parameter tuning for grid search and better model performance. The accuracy is significantly improved with the help of these algorithms. We have made sure to train, test and apply grid search for all the models used. With this project we could implement the ensemble methods and choose th best model which answers problem statement to predict the song popularity. This project can be an example for the real world scenario and help other practitioners. Given its applications it can be widely used to improve the recommendation systems and help the upcoming artists without focusing on established artists. It also posses a great pathway to improve the music industry and help in making sustainable decision in the industry

M. Evaluation of performance

Testing the model on unseen data using evaluation metrics F1-score, AUC - ROC. The ROC curves are the graphical representations on the performance of a binary classifier system as its discrimination threshold is varied. The ROC curve for Decision tree classifier represents two classes 0 and 1 with both having an area under the ROC curve (AUC) of 0.77. This suggests the model has a fair ability to distinguish between two classes, but there is still room for improvement

From Fig. 11. For an XGBoost classifier, the AUC for both classes is 0,95. This indicates the XGBoost model has a very good performance in terms of distinguishing between the positive and negative classes.

From Fig. 12. Testing the model on unseen data using evaluation metrics F1-score, AUC - ROC. The ROC curves are the graphical representations on the performance of a binary classifier system as its discrimination threshold is varied. The ROC curve for Decision tree classifier represents two classes 0 and 1 with both having an area under the ROC curve (AUC) of

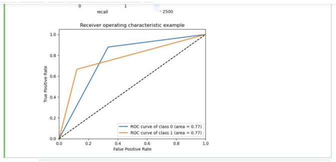


Fig. 12. ROC of Decision Tree

0.77. This suggests the model has a fair ability to distinguish between two classes, but there is still room for improvement

Given the above two, we can conclude that the XGBoost model is performing substantially better that Decision Tree model.

N. Teamwork

The various phases of this project are distributed among the team members

- Vaishnavi Mocherla and Sri Mounika Jammalamadaka : Understanding various topics available for the project ideas, summarising and choosing the domain for the project
- Vijay Rama Raju Penmatsa and Nikhil Gudur : Initial data collection and finalising the dataset in the plethora of dataset
- Vijay Rama Raju Penmatsa : initial Data Preparation and Exploratory Data Analysis
- Sri Mounika Jammalamadaka : Further Exploratory Data Analysis and Data Preprocessing
- Nikhil Gudur and Vaishnavi Mocherla : Feature selection, model selection and finialing the machine learning models
- Vaishnavi Mocherla : Implementing the models
- Vaishnavi Mocherla Vijay Rama Raju Penmatsa Nikhil Gudur Sri Mounika Jammalamadaka : Evaluate the model performance, implementation of different metrics to improve model performance
- Vaishnavi Mocherla and Vijay Rama Raju Penmatsa : Documentation and report writing
- Sri Mounika Jammalamadaka, Nikhil Gudur : Visualizations and Presentation slides

O. Technical difficulty

Given the API of spotify dataset, we have encountered major challenges while retrieving the data from the Spotify API link, initially the data couldn't be retrieved with the single API end point which was addressed by calling multiple API end points and merged the data collected from multiple end points. However, this did not seem fruitful for our project as there a certain rate limit for the developer token. Attempts to request for higher rate limit have not been successful.

P. Practiced pair programming

We have practiced pair programming which was crucial as part of implementing machine learning models in understanding the key features and evaluate the models. With the state of art facilities available at SJSU library right at our disposal pair programming was easier to implement

Q. Practiced agile or scrum

Our process is well documented, we have made sure to practice agile/ sprint with regular weekly and bi-weekly sprints. Our approaches and sprint meetings are documented using Trello

R. Used Grammarly or other tools for language

To ensure correctness, clarity and detail in the content we have used Grammarly which is well suited and highly recommended by professionals in documentation

S. Slides

The slides are well designed with minimal content to make sure the viewer understands the key takeaways at a glance. Made sure to have one point descriptions and self explanatory to help the audience understand the topic better and grasp their attention while presenting.

T. Saving the model for quick demo

All the model code files are uploaded to GIT along with the results to help the viewer understand the code at a glance, the models uploaded in git are ready to implement for a quick demo

U. Used LaTex

LaTex which is highly recommended for IEEE documentation is used to minimise manual error while formatting and adding project content, and maintain the professional writing style throughout the report the .tex files are submitted.

V. Used creative presentation techniques

Presentation includes various box plots, heatmaps and other visuals which help the audience understand and follow the presenter throughout the demonstration. Made sure to add one pointer information slides to leave room for any queries and other points and not have crowded slides.

W. Literature Survey

Literature survey was conducted extensively, literature survey culminates various papers on Song Popularity Prediction. This survey has helped us gain insights and understand the challenges and limitations posed in the available papers.

X. Git Link

https://github.com/Vaishnavi-mocherla/song-popularity-prediction

Y. Trello

https://trello.com/b/83d49Lnc/ml-project-sprint

Acknowledgment

The authors would like to thank Professor Dr. Vishnu Pendyala, Teaching Assistants for being supportive throughout the process. This work is done as a part of Machine Learning course work.

References

[1] Chiru, C., & Popescu, O. (2017). Automatically determining the popularity of a song. In Lecture notes in computer science (pp. 392–406).

[2] ”HIT SONG PREDICTION FOR POP MUSIC BY SIAMESE CNN WITH RANKING LOSS,” 2017. [Online]. Available: https://arxiv.org/pdf/1710.10814v1.pdf.

[3] ”Machine Learning and Chord Based Feature Engineering for Genre Prediction in Popular Brazilian Music,” 2019. [Online]. Available: https://arxiv.org/pdf/1902.03283v1.pdf.