CSE343 Machine Learning - Interim Project Report

Kunal
2022260
Kunal22260@iiitd.ac.in

Akshat Gian
2022051
akshat22051@iiitd.ac.in

Dhairya
2022157
dhaiya22157@iiitd.ac.in

Ashwin Verma
2022117
ashwin22117@iiitd.ac.in

Arun Singh Rawat
2022106
arun22106@iiitd.ac.in

1. Abstract

We all have a fondness for music, don't we? Most of us have favorite songs, some we truly enjoy, and others we simply don't like. While it might seem that our musical preferences are based on chance, this project seeks to investigate whether there is a data science element influencing why certain songs are more popular than others. Our objective is to analyze musical characteristics to predict a song's popularity among listeners and gauge the extent of that popularity. This research could also assist music composers, singers, and instrumentalists in crafting new music that resonates with audiences. The entire team finds this topic both intriguing and innovative, with ample opportunities for exploration, as music and its traits can be analyzed through various factors. Additionally, as we enhance our skills in machine learning and signal processing, we anticipate numerous possibilities for further exploration in this field, making it a valuable project for ongoing development and iteration in the future https://github.com/arun22106/HitTrack(GithubLink)

2. Introduction

This project focuses on analysing if there exists a relationship between music popularity and song characteristics. This project dives deeper into trying to answer the question of whether a particular song's popularity has some underlying data-driven factors leading to it. The main objective of this project is to leverage attributes of a song to predict its popularity and also predict the degree of popularity it may achieve.

3. Literature Review

3.1. Reference 5

In this paper, they are basically building a methodology that can predict whether a song will appear on Spotify's Top 50 Global ranking after a certain amount of time. They approach the problem as a classification task and use the data from the past platform's Top 50 Global ranking collected using Spotify's web API, The model uses information on the songs previously observed in that list, Support Vector Machine classifier with RBF kernel reached the best results in our experiments with an AUC higher than 80% when pre- dicting the popularity of a song two months in advance.

3.2. Reference 6

This project focuses on predicting the popularity of songs on the Million Song Dataset, a crucial aspect for maintaining competitiveness in the expanding music industry. The dataset contains audio features and metadata for around one million songs, the study assesses various classification and regression algorithms to determine their effectiveness in forecasting song popularity. The investigation also identifies the specific types of features that possess the greatest predictive capability in this context. In this paper, they evaluated different classifications and regression algorithms on their ability to predict popularity and determined the types of features that hold the most predictive power.

3.3. Reference 7

This paper primarily focuses on the Popularity Prediction of Music by related Python tools and various ma- chine learning models like xgboost, XGBRegressor, and Polynomial Regression, the dataset for the study contains 114000 data points and 19 features. The research evalu- ates models using metrics such as mean-squared error and R-squared. Ultimately, XGBoost emerged as the model, demonstrating a strong correlation between music attributes and popularity. The paper also discusses potential areas for improvement, including refining categorical variable encoding and exploring interactions between independent variables

3.4. Mathematical Concepts

Mean: It is the total sum of values in the dataset divided by the total number of data points, it is a measure of the central tendency of a probability distribution along median and mode.

Standard deviation: It is the measure of the spread or dispersion of a set of data points from their mean, It helps assess the consistency and reliability of data which in turn draws meaningful conclusions.

Covariance: It is a measure of the relationship between two random variables and to what extent, it indicates whether an increase in one variable corresponds to an increase or decrease in another variable. A positive covariance suggests a positive relationship, while a negative covariance indicates an inverse relationship and a covariance of zero implies no linear relationship between the variables.

Correlation: It is a statistical measure that expresses the extent to which two variables are linearly related.

4. Dataset Description

The primary source of our dataset is the Spotify API which analyses the songs uploaded on its platform on the basis of various sound characteristics. The Spotify API essentially provides us with 2 useful datasets, one which primarily contains the numerical parameters such as:

1. song duration ms: This refers to the length of the track in milliseconds.
2. acousticness: This is a confidence measure from 0.0 to 1.0 that assesses whether the track is acoustic. A value of 1.0 indicates high confidence that the track is acoustic.
3. danceability: This describes how suitable a track is for dancing, considering elements like tempo, rhythm stability, beat strength, and overall regularity. Least danceable is 0.0 to 1.0 being most danceable.
4. energy: This is a measure from 0.0 to 1.0 that represents the perceptual intensity and activity of the track. Energetic tracks typically feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.
5. instrumentalness: Detects the number of different instruments involved in the songs.
6. key: This represents the estimated overall key of the track, with integers mapping to pitches using standard Pitch Class notation.
7. liveness: Detects the presence of an audience in the recording. Value is proportional to probability that music was recorded in a live event
8. loudness: Measured in decibels (dB). It is basically the decibel value for each instance averaged across the entire track and is useful for comparing relative loudness of tracks. Values typically range between -60 and 0 db.
9. audio mode: This indicates the modality (major or minor) of the track, the type of scale from which its melodic content is derived. Major is represented by 1, and minor is represented by 0.
10. speechiness: Measure of the presence of spoken words in a track. When the track is something like an audiobook or podcast where there are mostly clearly said and complete words with a lot of understanding and speech involved this parameter tends to 1 and the other part of the spectrum is the songs which have absolutely no wordings involved in them.
11. tempo: beats per minute (BPM). i.e. the pace of a given song
12. time signature: This provides an estimated overall time signature of the track, specifying how many beats are in each bar (or measure).
13. audio valence: range: [0, 1] is a measure of the musical positiveness conveyed by a track. Higher value means happier the song is while lower the value means that the song is sad, intermediate values convey the balance between positivity and negativity of the song
14. song popularity: Based on rating and number of listens and re-listens by the audience.

This dataset is the song data.csv, other than this there is another dataset for some additional information such as:

1. artist name: is the name of the singer of the track
2. album names
3. playlist: is the playlist name in which the song is launched in spotify.

4. song name

This dataset is named song info.csv

For our project, we started off by making a new dataset using their primarily available datasets. To form our dataset we have used the concat() function horizontally ie: on axis1 to both the datasets, since the artist name column is common in both of them we dropped it from the second dataset, while since our target variable is the song popularity we added it as the target column towards the end.

Final dataset dimensions: (18835, 19)

This is followed by classifying the numerical score into categories to turn it into a bounded classification problem:

Category	Popularity Score
Potential Masterpiece	78-100
Popular	69-78
Mildly Popular	56-69
Potential Flop	0-56

Table 1. Category wise popularity score

These boundaries have been decided based upon certain percentile calculations which has been discussed more in detail further.

Category	Percentile
Potential Masterpiece	90%
Popular	75-90%
Mildly Popular	50-75%
Potential Flop	0-50%

Table 2. Category wise percentile

We found out what are the percentile scores for each and also plotted distribution plots and quartiles to decide on the numbers

Field	Value
mean	52.991877
std	21.905654
min	0.000000
5%	8.000000
10%	21.000000
25%	40.000000
50%	56.000000
75%	69.000000
80%	72.000000
90%	78.000000
95%	85.000000
max	100.000000

Table 3. Caption

Other than this we have done the Standardization of the dataset to improve the model predictions and accuracy later on however minor tweaking is expected to be needed as we move on further into the project. Encoding was needed for the varchar values in the dataset ie: song name, artist name, album name, playlist we have used one-hot encoding for it. Missing values were printed however the dataset has no NaN or missing values hence not needed to be removed/replaced with the average. Distribution curve was plotted to find the percentile and get an idea of the distribution. The resultant plot is a normal distribution as expected. Feature selection and outlier detection has also been done which has been described in more detail in the methodology.

5. EDA

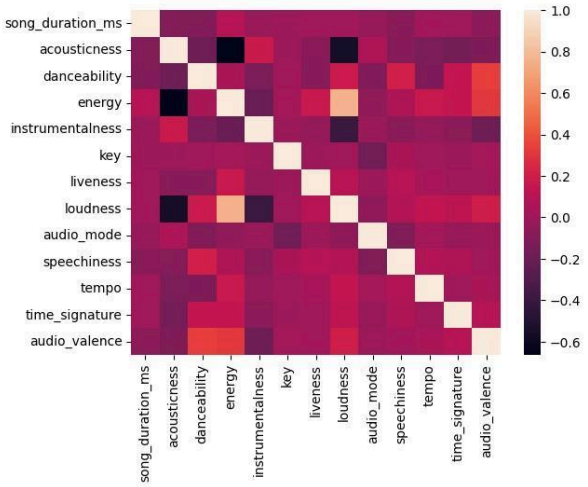

Figure 1. Heatmap of Features

Observations from Figure 1 are:

1. loudness is inversely correlated with number of instruments.
2. loudness is correlated with energy.
3. loudness is inversely correlated with acousticness.
4. acoustic ness is inversely correlated with energy.

We have performed a very detailed EDA for every feature to see that please refer to annexure.

6. Methodology and Model description 6.1.

Data Collection

We obtained two primary data files, namely 'song data.csv' and 'song info.csv', and details about these files are described above in the 'Data description' section.

6.2. Data Integration:

We merged these two datasets using the Python pandas library to create a unified dataset.

6.3. Categorization of Songs

To categorize the songs effectively, we utilized the describe function to calculate percentiles and scores. We classified the songs into four distinct categories based on their popularity score:

- Potential Masterpiece: This category comprises songs with exceptionally high scores, indicating significant potential for success.
- Popular: These are songs that have already achieved a high level of popularity.
- Mildly Popular: This category includes songs with moderate levels of popularity.
- Potential Flop: Songs in this category have scores suggesting a lower likelihood of success.

For categorization of data in described categories we computed the percentage of data falling into 4 categories:

- Percentage of songs that are potential masterpieces: 10.96%
- Percentage of songs that are popular: 14.57%
- Percentage of songs that are mildly popular: 24.48%
- Percentage of songs that are potential flops: 49.97%

6.4. Data Pre-processing

We conducted initial data pre-processing, which involved: Checking for Null or Missing Values: We systematically examined the dataset for any missing or null values. We did min-max scaling on the dataset.

6.4.1 Feature Engineering

Since we decided to only have the song name as the categorical data since the name of the song can have an impact on its popularity. First we created a bag of words then replaced the song name with the most occurring word in that name and then did one hot encoding on the song name after that we ran PCA on the one-hot encoded dataset and found that variance is too much distributed among the features. After encoding, features set became of size more than 6000 and 1000 features were incorporating about 60% of the variance in the data as can be seen in figure 2. As we can see the number of features have increased too much computational complexity will increase exponentially.

So we looked for any other method to do this and we decided to do sentiment analysis. In this case we used a well known NLP library to judge the sentiment of the song just by looking at its name. We used TextBlob and NTLK to do the sentiment analysis. We found similar results in the cases of both and we just decided to stick with TextBlob. Based on sentiment analysis we assigned -1 (negative sentiment),0

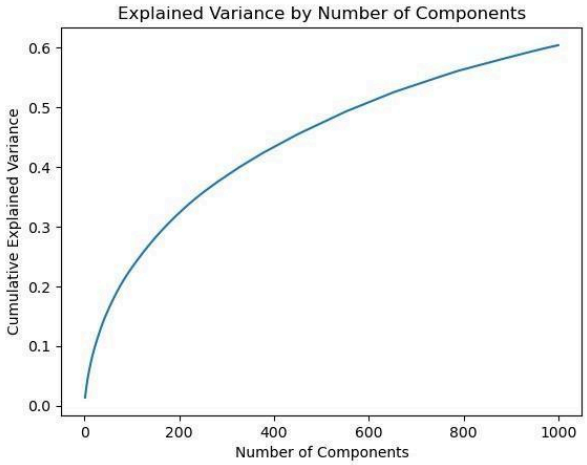

Figure 2. Explained variance vs. Number of components

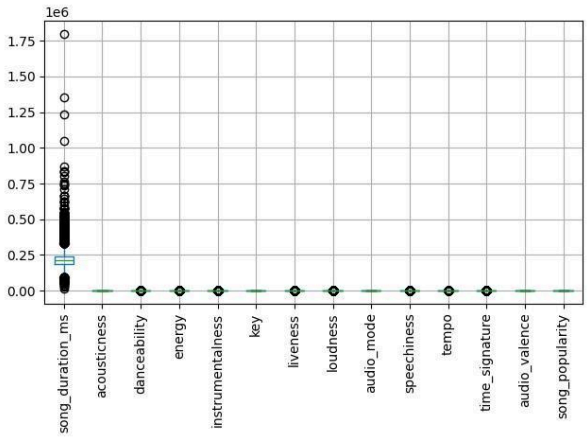

Figure 3. Boxplot of features (I)

(neutral sentiment) or 1 (positive sentiment) to a data sample depicting its sentiment.

6.4.2 Outlier Detection and Removal

We used a boxplot to detect the outliers in the data. From Figure 3 and Figure 4 we can clearly see that there are many sample points in the dataset which are deviating from the central tendency of the data therefore there will be outliers in the dataset and we have to figure out some methodology to remove them.

We used Local Outlier Factor (LOF) score with 1% contamination to remove the outliers when we increased the contamination the accuracy was getting affected. Using this we removed around 180 data samples.

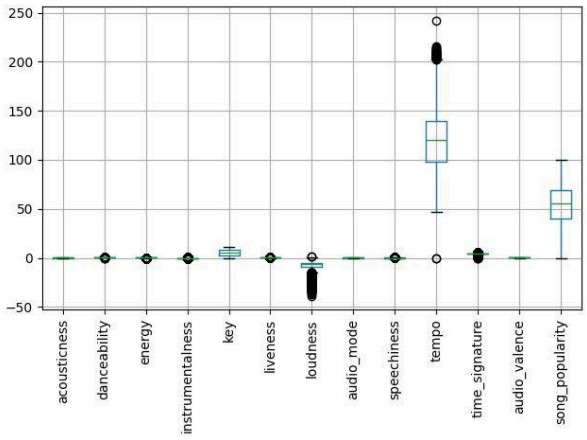

Figure 4. Boxplot of features (II)

6.5. Model description

We have tried several classification models on the dataset. Following are the best accuracy scores we obtained after using grid search.

Model	Accuracy Score
Naive Bayes	0.511
Logistic Regression	0.523
Decision Tree	0.632
SVM (Linear)	0.530
SVM (RBF)	0.705
Random Forest	0.783
KNN	0.669
MLP	0.539
Ensemble (Random Forest and KNN)	0.713
AdaBoost	0.538

Table 4. Accuracy scores of different ML models

As we can see Random Forest performed best out of all the models so we further tried to improve the accuracy by applying the ensemble methods. We trained 3 different Random forests, one with different subsets of 90 % features, one without bootstrapping and last one just with best parameters, no feature sampling and with bootstrapping we obtained using grid search and we also added KNN because we were getting fairly good accuracy by using it. But the overall accuracy is reduced as we can see in the above table.

We found the random forest to be the best model.

The best parameters for random forest we found are [’n estimators’: 1000, ’max _depth’: None (Other parameters were default)].

```
Confusion Matrix:
[[ 288   11  602    0]
 [  20  306  182    7]
 [  65   22 1887   12]
 [   1    4   50  310]]
Classification Report:
                     precision    recall  f1-score   support

       Mildly Popular       0.77      0.32      0.45       901
              Popular       0.89      0.59      0.71       515
       Potential Flop       0.69      0.95      0.80      1986
Potential Masterpiece       0.94      0.85      0.89       365

             accuracy                           0.74      3767
            macro avg       0.82      0.68      0.72      3767
         weighted avg       0.76      0.74      0.71      3767

Recall Score: [0.31964484 0.59417476 0.95015106 0.84931507]
Precision Score: [0.77005348 0.89212828 0.69349504 0.94224924]
```

Figure 5. Confusion Matrix - Random Forest

7. Result and Analysis

After performing 10 K-fold cross validation tests on our Random Forest model we achieved a mean accuracy of almost 75% with a peak of 78% accuracy. The confusion matrix for the Random Forest model we created using the best parameters is shown in the fig 5.

8. Conclusion

In our quest to predict song popularity, this accuracy is a significant accomplishment in the intricate and subjective world of music. It is essential to understand that perfection is nearly impossible in music prediction due to the diverse and ever-changing nature of musical tastes. Our success in correctly predicting popularity in over three-quarters of cases underlines the model’s robustness and the

importance of sound characteristics in determining a song's appeal. This outcome is a validation of our approach and this model can be a valuable tool for artists and the industry, offering insights that go beyond intuition to a more data driven understanding of what makes a song a hit.

9. References

1. https://www.kaggle.com/datasets/edalrami/19000-spotify-songs
2. https://towardsdatascience.com/songpopularity-predictor-1ef69735e380
3. https://developer.spotify.com/documentation/web-api
4. https : / / www . kaggle . com / code / amansorout/spotify-song-popularityclassification
5. https : / / www . researchgate . net / publication / 341420234 _ Predicting _ Music _ Popularity _ on _ Streaming _ Platforms / link / 5f0d0bd8a6fdcca32ae97ccc/download
6. http://cs229.stanford.edu/proj2015/140_report.pdf
7. https : / / www . researchgate . net / publication/370712842_Popularity_ Prediction_of_Music_by_Machine_ Learning_Models