# Spotify Music Popularity Prediction

Munukutla D. V. Kashyap
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois, United States
dmunuk2@uic.edu

Sai Mahesh Vemulapalli
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois, United States
svemul20@uic.edu

Naga Sumanth Choudary M.
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois, United States
nmovva@uic.edu

## ABSTRACT

Identifying and understanding the estimated popularity of a track before it is shown to the public based on its features is an integral part of music production. Musicians depend on this to best plan their releases and tune their tracks better, and music platforms use this estimation to better cater to the trends on their platform. Spotify is one of these platforms, and it is also one of the leading platforms in the entire industry.

Our project aims to improve this part of the music industry, which involves estimating or predicting whether a music track would be popular or not in the near future on Spotify's platform using the features of the track as the predictors. Existing works surrounding this topic often concentrate on the exact popularity index prediction, which may be obtuse to a newcomer. This project instead is focused on being more accessible and approaching the task from a classification standpoint using the acoustic features of the track. The Random Forest classifier was picked as the top choice from the tested models, with an accuracy of 78% when predicting whether the song would be popular.

## 1 INTRODUCTION

Music has existed since ancient times and has become an integral part of almost everyone's life. Due to this immense demand and the internalization efforts in the last few centuries, the music industry has reached a staggering 28.29 billion USD in the year 2023 and estimated to grow to 42.62 billion USD by the year 2028, according to a report from Globe Newswire. And the people who create music are, by extension, numerous. Most of these artists rely on this craft to make a living. This means that it is crucial for them to understand the trends of highs and lows in the industry to make sure that they are ahead of the curve and would not release a track at the wrong time and risk a considerable amount of their work going unnoticed.

Out of all the platforms that listeners use to listen to music in all formats, Spotify occupies the top spot, with a 30.5% share of the entire music streaming industry, followed by Apple Music at a much lower 13.7%. Hence it is imperative to make sure that the artists are able to obtain an estimation of whether their track would be popular in the near future on Spotify's platform, as it would account for the majority of their revenue on their work. Spotify uses a proprietary algorithm to assign a popularity index to all the tracks on their platform. This algorithm is highly complex and obscure, using attributes like the number of listens for the track and how recent the listens are in order to generate this popularity index. For an outside viewer, it is generally not possible to gauge the popularity trends on the platform without proper analysis. Here, our project enters the scene to bridge the gap by building and training a model that can understand the popularity trends and provide the artist with a reliable estimate of their track's popularity on the platform in the upcoming weeks. On the other hand, music streaming platforms rely on popular tracks being shown to viewers in order to generate more and more traffic on their platforms. A report by MusicAlly estimates that 15% of the most popular tracks on Spotify account for 95% of the platform's revenue in the year 2022. So it is crucial for the platform to identify these undiscovered diamonds in the rough as soon as possible to provide quality recommendations for the users.

## 2 KEYWORDS

XGBoost – Extreme Gradient Boosting

API – Application Programming Interface

## 3 METHODOLOGY

### 3.1 *Data Collection*

The data was collected through the usage of Spotify's WebAPI, available for all registered Spotify users. For the sake of comprehensiveness, we collected data from 114 most common genres of music.

For each genre, we collected information about the audio features and the basic track features provided by the API. These include attributes like acousticness, danceability, explicit, track name, duration, tempo, popularity, etc. Due to the rate limits imposed by Spotify on its API usage, only a limited number of API calls can be made every 30 seconds. However, we had to make an API call initially to gather the track ids for tracks in a genre in batches of 100 (limit imposed), and subsequently made an API call for each of the tracks to gather their audio features. This means we had to make nearly 12500 API calls over the course of this data collection over the course of multiple days to stay under the rate limitations of the platform.

### 3.2 *Data Preprocessing*

Data has been preprocessed by removing certain aspects, such as features and parameters that have a very low impact on the model selection. These also include the null values in the dataset.

In order to perform any type of analysis on the data, the data must be converted to a numerical format from the categorical format that is used in the dataset. For this, we performed one-hot encoding on such columns. A similar procedure was contemplated for the column containing the artist's name. However, due to its minimal influence on the track's popularity and the resulting excessive sparsity of the dataset, we decided against pursuing it.

### 3.3 *Data Analysis*

To determine the popularity of a track, we establish a threshold based on a numerical value ranging from 0 to 100. In order to achieve a more even distribution, a threshold of 40 has been established for the popularity value. This implies that any track with a value over 40 is classified as popular, while any track with a value below 40 is classified as not popular. From which we can see that there are 46674 popular songs and 67325 Unpopular songs.

Additionally, we examined the frequency of popular tracks categorized by genre in the data. It is understood that the dataset consists of 1000 tracks for each genre. However, examining the proportion of popular tracks within these 1000 for each genre provides insights into the platform's patterns. Given the large variety of genres, we conducted a detailed analysis of the top 5 genres that have the biggest quantity of popular songs. Out of which Sertanejo being the most popular genre with 999 popular tracks while the runner-up being Brazil with 964 popular tracks.

Moving on to correlation analysis, we calculated the correlation between features and the target variable popularity_flag. The results are sorted and analyzed, with a focus on identifying features with the highest absolute correlation. A thresholding mechanism is employed to filter features that exhibit a correlation significantly higher than the average.

### 3.4 *Feature Selection*

Earlier we have dropped some columns such as ‘artists’, ‘track _genre’, ‘track_name’, etc., due to their lack of relevancy for the final predictions as an artist’s name does not matter when our focus is on what song would be popular in the coming days based on the features of the song. Instead we have used columns such as ‘explicit’, ‘danceability’, ‘loudness’, etc., as they are very good at determining what a particular song is all about.

With these features we have created a correlation matrix.

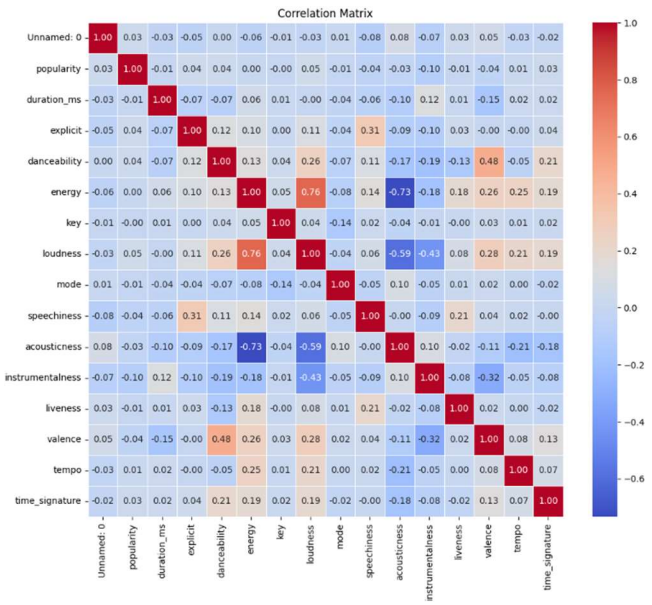


Fig 1. Correlation matrix depicting the correlation between the various numerical attributes of the data set.

We can observe from the figure that the columns labelled "loudness," "danceability," and "explicit" have a strong relationship with how popular a particular song is. On the other hand, we have made the decision to preserve the other columns as well because they will have some kind of impact on the popularity_flag in some way or another. And we have decided to remove the columns such as ‘track_id’ as they do not play any role in determining the popularity of a song.

### 3.5 *Model Training*

Train-Test Split: The dataset, comprising 114,000 rows and 55 feature columns, was split into 85% for training and 15% for testing. This split ratio is a standard practice in machine learning, providing a substantial amount of data for training the models while reserving a representative portion for testing their performance on unseen data. The training set allows the algorithms to learn patterns and relationships, while the test set evaluates their predictive accuracy. The following three algorithms are selected to train the models for the classification problem of categorizing whether a track is popular or not.

Decision Trees: Our project utilized Decision Trees due to their interpretability and straightforward nature. These models work by creating a hierarchical structure of decisions, where each node represents a feature and branches represent decision rules. This approach is advantageous for unraveling the complex relationships between various musical features and a song's popularity, making it possible to directly trace how each attribute influences the outcome.

Random Forests: To address the limitations of Decision Trees, namely overfitting, we incorporated Random Forests in our model training phase. Random Forests are an ensemble learning method that construct a multitude of decision trees at training time and output the average prediction of the individual trees. This method not only reduces the risk of overfitting but also enhances the overall accuracy and stability of the model, making it particularly suitable for handling our large and diverse dataset.

XGBoost: XGBoost was selected for its high efficiency and scalability, crucial for our extensive dataset. This advanced algorithm builds upon the concept of gradient boosting, focusing on optimizing computational speed and model performance. XGBoost improves upon the predictions of previous trees and continually adjusts to achieve more accurate results. Its robust handling of various data types and missing values, combined with an efficient implementation, makes XGBoost a powerful tool in predicting music trends.

### 3.6 *Model Tuning*

The project involved tuning Decision Trees, XGBoost, and Random Forest models. The goal was to fine-tune these models to suit the specifics of the dataset and enhance predictive accuracy. The tuning process was guided by a scoring criterion focused on accuracy, ensuring that the chosen parameters maximized the correct predictions out of all predictions made.

Decision Trees Tuning: The tuning process focused on parameters like 'criterion', 'max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', and 'splitter'. The choice of 'entropy' as the criterion and a 'random' splitter, combined with no limit on 'max_depth', indicates a preference for a more complex model capable of capturing nuanced patterns in the data. The 'auto' setting for 'max_features' and a higher count for 'min_samples_leaf' further suggest a model that balances specificity with generalization.

XGBoost Tuning: For XGBoost, parameters like 'learning_rate', 'n_estimators', 'max_depth', 'min_child_weight', 'gamma', 'subsample', and 'colsample_bytree' were finely tuned. The optimized settings, particularly a higher 'learning_rate' and 'max_depth', coupled with an increased 'n_estimators', demonstrate the model's enhanced capability to learn complex relationships in the dataset. The 'colsample_bytree' and 'subsample' values suggest a strategy to ensure diversity in the model's training process, boosting its predictive strength.

Random Forests Performance: The Random Forest model yielded the best performance using its default settings, which speaks to the inherent strength of the algorithm. Its ensemble nature, leveraging multiple decision trees, inherently provides a balanced approach to handling the dataset. This robustness and adaptability make it a powerful tool even without extensive hyperparameter tuning, as it can effectively manage the diverse features and complexities present in the dataset.

### 3.7 *Model Evaluation*

In this project, the evaluation of machine learning models for predicting music popularity provided insightful results. The Decision Tree model, with an accuracy of 72.26%, precision of 69.63%, and recall of 62.88%, offered a balanced performance. However, when compared with the Random Forest and XGBoost models, its effectiveness was slightly lower. The Random Forest model emerged as the most proficient, boasting the highest accuracy of 82.95% and the best recall of 74.38%. This high recall is particularly significant for music platforms like Spotify, as it ensures popular songs are accurately classified, enhancing user experience by recommending the most relevant tracks.

The XGBoost model also performed admirably, achieving an accuracy of 78.01%, precision of 74.85%, and recall of 68.95%. While it outperformed the Decision Tree in all metrics, it fell short of the Random Forest model, particularly in terms of recall. In the context of music popularity prediction, the recall metric holds substantial importance. High recall reduces the likelihood of missing out on genuinely popular tracks, a critical aspect for streaming platforms that rely on accurately capturing user preferences and trends.

Overall, the Random Forest model's superior performance across accuracy and recall makes it the most suitable choice for this project. Its ability to effectively identify popular songs without substantial false negatives is essential for platforms aiming to deliver curated and engaging content to their users. This model aligns well with the objectives of music streaming services, which prioritize accurate trend prediction to enhance user engagement and satisfaction.

## 4 CONCLUSION

In this project, we present a model that predicts whether a given track would be popular on Spotify platform. We collected data from the streaming platform Spotify and trained classification models that are able to make predictions into the near future. Based on these models, the best results were obtained by the Randon Forest classifier with an accuracy of 82.95%.

While our work at this moment is limited to predicting the popularity of a track on Spotify platform, this can be further extended to other platforms in the future, like Apple Music and Youtube Music. We also identified a few improvements that could be made for our implementation, with the key improvement being including the social network data to keep track of the trends and better predict the near future popularity of the tracks.

## 5 PROJECT LINK

GITHUB: https://github.com/kashyap-1234/spotify