Spotify Data Analysis: Predicting Song Popularity

Zuoqi Zhang
Department of Computer Science
Boston University
Boston, MA 02215
zqzhang@bu.edu

Charles Huang
Department of Computer Science
Boston University
Boston, MA 02215
chahuang@bu.edu

Abstract

In this paper, we aim to predict how popular a new music track will be in the future based on its audio features. We model this problem as a classification task and apply different machine learning classifiers on the dataset obtained from Kaggle and the Spotify Web API. We then evaluate the performance of each model using its accuracy. We also compare the results of using different subsets of features in order to find the best predictors. Finally, we propose our own model which results in an 84.3% accuracy for song popularity prediction.

1 Introduction

Currently, music streaming is ubiquitous. Thus, predicting song popularity is particularly important in keeping businesses competitive within a growing music industry. The ability to make accurate predictions of song popularity also has implications for customized music suggestions.

What exactly makes a song popular? It is obvious that the artist would greatly impact the popularity of a new song, but we wanted to focus on the audio features of the song in order to find what features top songs have in common. In particular, our ultimate goal was to build a model which can predict how popular a new song will be in the sense of ranking or streams.

In our project, the input is a dataset from Spotify which contains the audio features of all of the top songs. We then proceed to use a number of machine learning classification algorithms (SVM, neural network, KNN, and decision tree) to output whether or not the song is popular (or relatively more popular than the others). From this, we were able to output which audio features are the best at determining the popularity of a song.

2 Data Description

We used a dataset from Kaggle and combined it with a dataset of audio features of songs crawled from Spotify in order to generate our dataset for model training.

2.1 Song ranking dataset

We first downloaded the dataset from Kaggle, which can be found at

`https://www.kaggle.com/edumucelli/spotifys-worldwide-daily-song-ranking`

This dataset contains the daily rankings of the 200 most listened to songs in 53 countries from 2017 and 2018 by Spotify users. It contains more than 2 million rows, which comprises 6,629 artists and 18,598 songs for a total count of one hundred and five billion streams. Each row contains a ranking position on a specific day for a song. For instance, the first 200 rows present the ranking for the 1st

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

of January in Argentina. The following 200 rows will contain the ranking for the 2nd of January in Argentina. The regions are alphabetically sorted.

See Table 1 for description of columns in the dataset.

Table 1: Kaggle dataset column metadata

Column name	Description	Data type
Position	Position on charts	Numeric
Track name	Title of song	String
Artist	Name of musician or group	String
Streams	Number of streams	Numeric
URL	Spotify URL of song	String
Date	Date of ranking	DateTime
Region	Country code	String

Since we would like to focus on the song rankings in the United States, we removed all of the rows of the other regions. There are 74,200 rows in the dataset that we used.

2.2 Audio features extraction

With the help of the Spotify Web API, we were able to obtain the audio features for each song in our dataset.

Audio features include accousticness, danceability, duration_ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time_signature, and valence. The definition of these audio features can be found at

`https://beta.developer.spotify.com/documentation/web-api/reference/tracks/`

From this, we created a data frame which contains the audio features of 1,966 unique songs.

2.3 Dataset generation

Since we already had the features, in order to do classification, we needed to label our data samples, that is, we needed to determine whether or not a song in our dataset was popular. To do so, we first used the dataset from Kaggle to calculate the total number of streams for each of the 1,966 songs. If the number of times a song was played was greater than or equal to the average number of streams, we labeled it as 1, which indicated it was relatively more popular, otherwise we labeled it as 0, which indicated it was less popular.

Our resultant dataset is a 1,966 $\times$ 14 data frame. The first 13 columns are the audio features and the last column is the label that we wanted to predict.

3 Data Analysis

For our initial data analysis, we generated a correlation matrix graph and the trend graphs for the audio features.

3.1 Correlation

The correlation matrix graph was used to see which audio features were most likely correlated. Using it, we were able to get an idea of which features can be grouped together, while we do our experiments to determine the features that can predict popularity. The higher the correlation, the more likely the two features can be used together to determine popularity. On the other hand, if two features have a highly negative correlation, we could determine that those two cannot be grouped together.

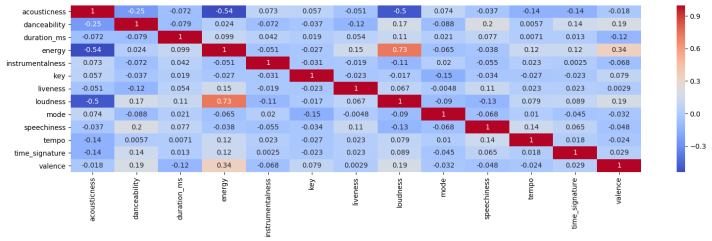

Figure 1: Audio Features Correlations

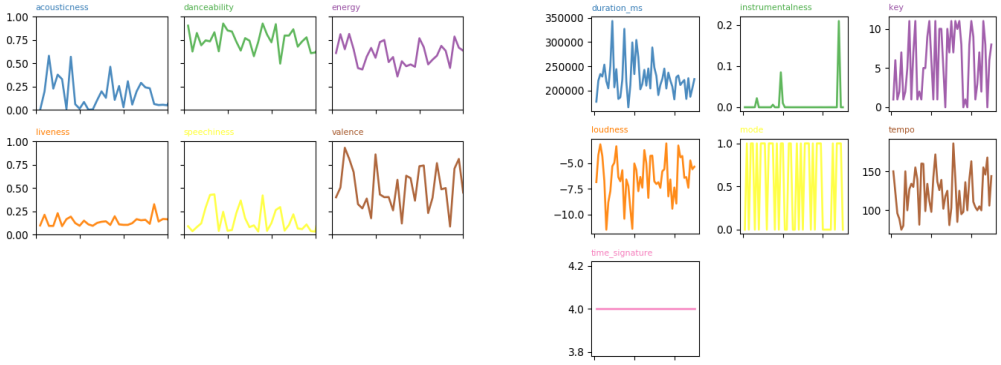

Figure 2: Audio Features Trends

3.2 Trends

The trend graphs were generated in order to see if there were any clearly visible upward-sloping or downward-sloping trendlines for certain audio features as the song ranking dropped. If one of the features had a downward-sloping trendline, as the rank decreased, we would have some evidence that this feature most likely can be used to determine high popularity. In contrast, if one of the features had an upward- sloping trendline, we would have evidence that this feature can be used to determine low popularity. Unfortunately, we were not able to obtain any concrete information from the audio features' trends.

4 Experiments

We first randomly split our dataset into two parts. We used 90% of the samples as the training set and 10% as the testing set.

4.1 Model training

The classifiers we chose are neural network, SVM (Support Vector Machine), KNN (K-Nearest Neigbors), and decision tree. The results are shown in Table 2.

Table 2: Classification results

Classifier	Accuracy
Neural network	82.2%
SVM	80.2%
KNN	81.7%
Decision tree	67.5%

4.2 Feature selection

In order to find which of the 13 audio features are the best predictors, we used all of the subsets of the features to do classification. We first used neural networks as our classifier, but it resulted in the same accuracy for all subsets as well as when using all of the features. Thus, we decided instead to use the k-nearest neighbors classifier. Our results showed varying results when with each program run. Nonetheless, from our results, we were able to deduce that danceability and instrumentalness were most frequently appearing in the top predictors. Thus, it can be assumed that they are two of the main features that decide the popularity of a song.

Table 3 shows the best predictors and their accuracy for one program run.

Table 3: Best Predictors Using KNN

Predictors	Accuracy
danceability, instrumentalness, loudness	84.3%
acousticness, danceability, instrumentalness, loudness	84.3%
acousticness, danceability, liveness, loudness, valence	84.3%
acousticness, danceability, instrumentalness, liveness, loudness, valence	83.8%
acousticness, danceability, loudness	83.2%
danceability, liveness, loudness	83.2%
acousticness, danceability, loudness, time_signature	83.2%
acousticness, danceability, loudness, valence	83.2%

5 Conclusion

From the experimental results, we can see that the neural network classifier performs the best with an accuracy of 82.2%. Unfortunately, the NN classifier resulted in all subsets of the features having the same accuracy when used as predictors. Thus, we instead used KNN, the second best classifier with an accuracy of 81.7%. From this, we were able to deduce that danceability and instrumentalness were most likely to be two of the best predictors of popularity. Using this, we produced our final model that reached an accuracy of 84.3% when predicting the popularity.

6 Future Work

Currently, we are using a binary classification method, and we want to switch to a multiclass classification method in the future in order to make our model more practical. For instance, we may divide the top 200 songs into four groups, the top 50 songs would be in the first group, the songs ranked between 51 and 100 would be in the second group, and so on and so forth. We also want to try to use some regression algorithms in order to be able to predict the number of streams for a new music track.

To improve the performance of our model, we would like to add other features like the dates on which the songs were released, since the release dates may impact song popularity and newer songs are typically assumed to be more popular. We will also try different values for the parameters of our model to find the best combination of parameters.

Acknowledgments

We would like to thank Prof. Peter Chin, the two TAs, Kieran Zhou and Gavin Brown, and the grader, Ken Zhou, for their guidance and help.

References

[1] J. Pham, E. Kyauk, E. Park. Predicting Song Popularity. 2015.

[2] N. Behrens. Making Your Own Spotify Discover Weekly Playlist. 2017.