MI-DSBA-AI Project Final Report

Dorveaux, CHEN MIN, Mohamed Nafees ABUBACKER, Jella Carillo

TOTAL POINTS

45 / 50

QUESTION 1

1 Final Report 45 / 50

- **0 pts** Correct
- **1 pts** Click here to replace this description.
- **2 pts** Click here to replace this description.
- **3 pts** Click here to replace this description.
- **4 pts** Click here to replace this description.
- ✓ **- 5 pts** **Click here to replace this description.**
- **6 pts** Click here to replace this description.
- **7 pts** Click here to replace this description.
- **8 pts** Click here to replace this description.
- **9 pts** Click here to replace this description.
- **10 pts** Click here to replace this description.
- **12 pts** -12
- **0 pts** Click here to replace this description.

Predicting Song Popularity

Explaining Song Popularity from Audio and Text Features

Mohamed Nafees ABUBACKER
b00759248@essec.com

Jella Marie CARILLO
b00767845@essec.com

Thomas DORVEAUX
b00604063@essec.edu

Min CHEN
b00758772@essec.edu

ABSTRACT

The project aims to explore the factors that make a song popular and to predict whether a song will be popular or not. The data that will be a combination of two datasets. One is from Spotify's API, where we look into the song's audio features (such as danceability, duration, etc). The other is from a lyrics database, where we look into the song's text features (such as repetitiveness or compression rate). We do this by first selecting the optimal number of features to analyze. After comparing various methods for feature selection (variance threshold, chi-squared, random forest, and recursive feature elimination), a total of 60 features were selected. After gathering the relevant features, we then test out 6 classification models to obtain the accuracy of song popularity prediction. We ran each model 5 times, one run per feature selection method aforementioned, and compared results. The models were evaluated using the ROC curve. The best model, Random Forest, correctly predicted song popularity at a rate of 79.7%. We conclude the project by identifying the characteristics of popular songs.

MOTIVATION

Ever since the advent of Napster, the music industry has completely transformed itself from using cassettes and CDs to streaming apps like Spotify, SoundCloud, iTunes etc. The revenue generated by the music industry has grown by 9.7% (US$ 19.1 Billion [1]) in 2018. This figure has been growing steadily for the fourth consecutive year in 2018 and is expected to grow furthermore in the upcoming years. During 2018 alone the revenue from streaming platforms grew a massive 34% [1] versus the previous year. Corresponding to the increase in the number of online paid music streaming app users (255 million [1] in 2018), the number of artists who publish their songs in these platforms has also seen a sharp rise.

The Billboard Magazine's then senior editor Samantha Chang stated that the influence of technology in the music industry will make it much more competitive for the artists to survive in the market. Hence, both the artists and the music distributors are compelled to discover new innovative ways to predict what their customers like and dislike, and thereby sustain in the market.

In this project, we intend to discover what makes a song popular by studying the audio and text features of songs and to predict song popularity.

PROBLEM DEFINITION

In this problem, we want to explore two main questions:

1) **What contributes to a song's popularity?**

One could hypothesize that an artist's popularity can contribute to it. Another could say that a song's positivity and upbeatness can be factors. On the other hand, in an article, it was said that each year, songs are getting more and more repetitive[2]. Does this contribute to the popularity of a song? How much do lyrics matter?

In this project, we want to find out which of the features contribute the most to how popular a song will become.

2) **How accurately can we predict if a song is popular or not?**

We want to test given the data whether or not we can accurately classify songs in 2 categories, popular and not popular.

REVIEW OF RELATED LITERATURE

Studies related to the prediction of song popularity have mixed results. They also use different methods with some using binary classifiers, and some using regression models, among others.

One of the pioneer studies on using machine learning to predict hit songs was done by Dharanaj and Logan[3]. In their model, they looked at both audio and text features of songs and used classifiers (Support Vector Machines, among others) to distinguish whether a song is going to be a hit or not. They have concluded that analyzing lyrics-based features is more effective at identifying song hits than looking into just audio features. Interestingly, they found that the absence of certain semantic information contributed more to a song being a hit, rather than the presence of certain lyrics.

Reiman and Örnell [4], explores whether or not it is possible to predict hit songs with Machine Learning using only audio features, as related literature shows mixed answers. They did a binary classifier to predict whether a song would be a hit or non-hit, using the dataset from Spotify Web API. They concluded that audio is not sufficient information to predict whether a song is going to be a hit or not. They suggested adding the features: genre, lyrics and artist popularity.

Nijkamp [5] also uses Spotify database API but looks at a different metric, namely the number of streams. In his work, he analyzed 1000 songs encompassing 10 genres. He concluded that the model he built, based on regression, did not sufficiently explain the number of streams on Spotify.

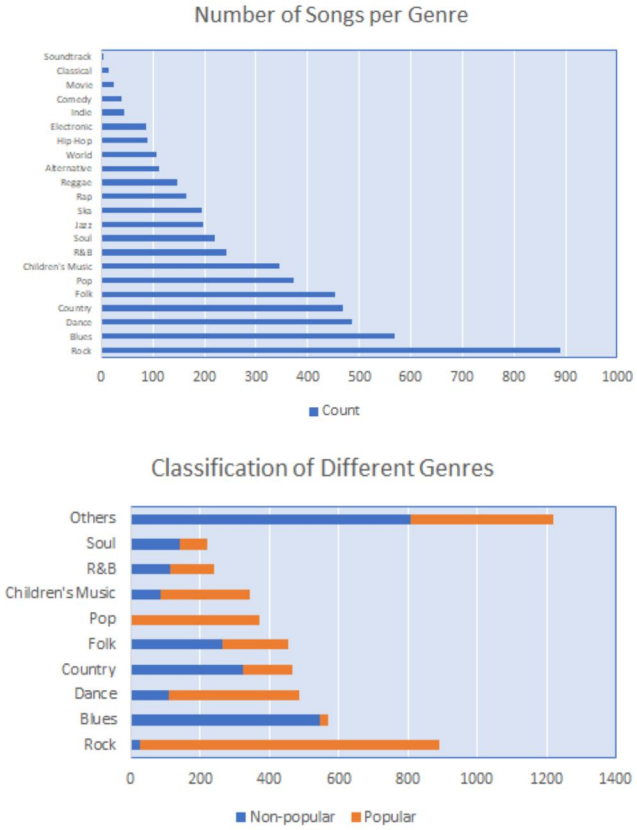
Branching out from using Spotify features, Interiano, Kazemi, et. al.[6] look into trends in the music industry, such as happiness, brightness and even "maleness". Their scope includes a larger database of 500 000 songs from the UK charts from 1985 to 2015 and includes both hit and non-hit songs. They define success as making it into the charts. They look at several features, that can be broken down into 2 categories: acoustic properties (such as timbre and danceability) and mood (such as sad, party, or happy). They conclude that successful songs are, on average, happier and score higher on danceability. They also added the "superstar" status which increased their prediction accuracy from 0.74 (only acoustic characteristics) to 0.86 (acoustic and superstar characteristics).

METHODOLOGY

I. Data Description

The dataset was obtained by merging two datasets from Kaggle: the dataset A of music features of 232,725 tracks and the dataset B of lyrics of 57,650 songs. The resulting dataset has 5261 songs after a few non-English songs were removed.

There are 22 genres in total, and the final dataset is skewed heavily towards rock music as seen in figure 1. Furthermore, it may be important to note that this genre has the highest proportion of popular music, if we were to split popularity by the top 25% (popular) and the bottom 75% (unpopular).



Feature 1: top (Number of songs per Genre), bottom (Classification of different genres)

The data has twelve music features evaluated by Spotify (including acousticness, danceability, duration in milli-second, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, valence) together with popularity, genre, artist name, track name and a unique track identity.
To begin with, it seems important to define the features that Spotify provide for each song and that we have used throughout our analysis.

For more detail, it is possible to see a complete description of these features on the Spotify API[7].

Table 1: Spotify dataset description

Spotify feature	Characteristics
duration_ms	length of the song (ms)
key	key of the track (C♯, C, D, etc.)
mode	major or minor (1 or 0)
time_signature	beats by bar (4 different types)
acousticness	measure of acousticness (from 0 to 1)
danceability	measure of danceability (from 0 to 1)
energy	measure of intensity (from 0 to 1)
instrumentalness	measure of vocalness (from 0 to 1)
liveness	is there an audience ? (from 0 to 1)
loudness	measure of loudness (dB)
speechiness	spoken words in the track (from 0 to 1)
tempo	beats per minutes (BPM)
valence	measure of happiness (from 0 to 1)

II. Data Cleaning

In this project, we obtained our desired data through two different databases. The first one (SpotifyFeatures.csv), includes song features of 232,725 tracks and the second one (only_lyrics.csv) includes uncleaned lyrics of 57,650 tracks.

***Merging of Datasets*:**
The final desired database is obtained through merging (inner join) the above-mentioned databases by matching each song's artist name and the track name. All the entries in those columns were converted into lower case and stripped of space to make it uniform between the two databases. We took the artist name also into consideration while merging because there were multiple instances in the data were a song with the same name has been sung by various artists.
The merged database did not have any missing values.

Eliminating Repetition:
In our final database, we have noticed multiple repetitions of a single song with different genres. For example, the song named 'I'm yours' by Jason Mraz has two instances for two different genres (Pop and Rock). We eliminated this repetition by sorting the database based on the song's popularity and then retain only the one with the highest popularity (first instance in the sorted database). After removing the repetition

Lyrics Cleaning:
The raw data had indicator words such as [verse 1], [chorous 1] etc. in it. These words were identified using Regular Expressions and were eliminated. Tabs, additional spaces and '\n' were also removed from the raw data. This cleaned lyrics were further used to identify the compression rate (An indicator of the repetition in the lyrics) and the sentiment values.
The code uses the function clean_lyrics() to do this cleaning process.

III. Feature Engineering

Classifying Popularity:
To tackle our main issue, which is predicting the popularity of a song according to musical features as well as sentimental ones, we have decided to turn popularity as a class, so that we can interpret our results in terms of accuracy. Basically, the idea was to train our model in order to see if it was able to classify our songs in the right category (roughly 1 if it is a popular song and 0 if it was not).

To do so, we tried to use a k-means algorithm to find a consistent classification.

In the beginning, we tried to build more than 2 categories. Figure 2 is the result of a 4-clusters kmeans. (Just a reminder, the popularity of a song is a score between 0 and 100 on Spotify).

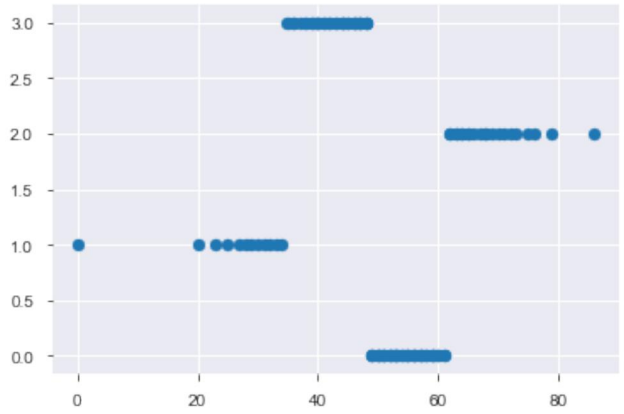


Figure 2: K-means on popularity classification

Due to poor results, we decided to simply put a binary classification at the 75th quantile or 3rd quartile. We considered the minimum score of 25% more popular songs. This binary classification was thus built with respect to the proportion of popular song in our dataset.

Designing Sentimental features:
As it was planned in our project proposal, we aim at using sentimental analysis on the lyrics of the songs to help predict popularity. We created 4 features to conduct our sentimental

study: senti_negative, senti_neutral, senti_positive and senti_total.

To do so we use a sentiment analyzer from the library NLTK called *VaderSentimentAnalyser*[8]. We followed the
This sentiment analyser gives a unidimensional metric sentiment for each word. By choosing thresholds of 0.5 and 1 over this metric, we can distinguish between neutral, negative and positive lyrics

For each song, we thus get a number of negative, neutral and positive sentences that consists of the features senti_negative, senti_neutral and senti_positive.
To build the feature senti_total, we only compute the difference between senti_positive and senti_negative to get a total score.

In addition to these features, we computed a compression rate. The algorithm uses here basically compress the text of the lyrics. The more the track is compressed, the more there is a repetitiveness in the word.

Analysing Musical & Sentimental features:
After adding the sentimental features to the classic Spotify features, we needed to analyze them to decide which transformation we had to make to improve their performance

To begin with, we plotted some graphs to have a look at the distribution of our independent variables as well as their influences on the dependent variable - the popularity. their distribution to see if they need to be scaled centred and - for continuous variables- turn into categorical/discrete data.

First, thanks to python, we drew scatter plots figure 3 of the variables related to the content of the lyrics :

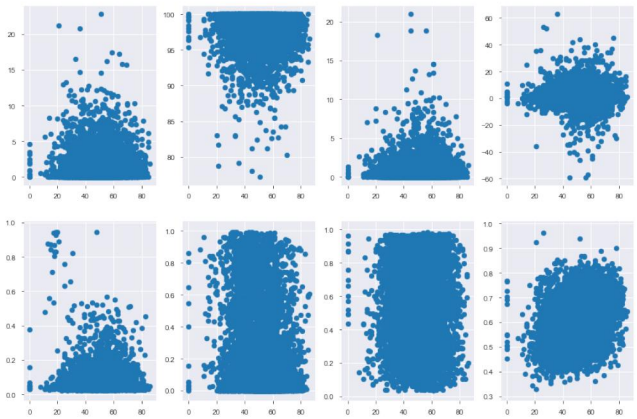


Figure 3: Scatter plot of lyrics variables

In these graphs, we can try to observe the relationship between what corresponds to the text features and popularity (from the top left: senti_positive, senti_neutral, senti_negative, senti_total, speechiness, acousticness, valence and compression_rate). From the plots above, it is difficult to identify any particular patterns that should guide us into the modification of the variables related to the content of the lyrics, at least for now.

Figure 4 are explanatory variables that Spotify provides (danceability, energy, instrumentalness, etc.) gives us a better insight to the feature engineering task we have to do. These variables below are not related to the lyrics but to the musicality of the songs :

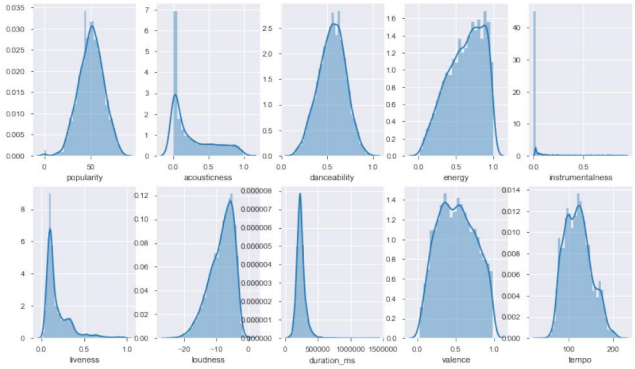
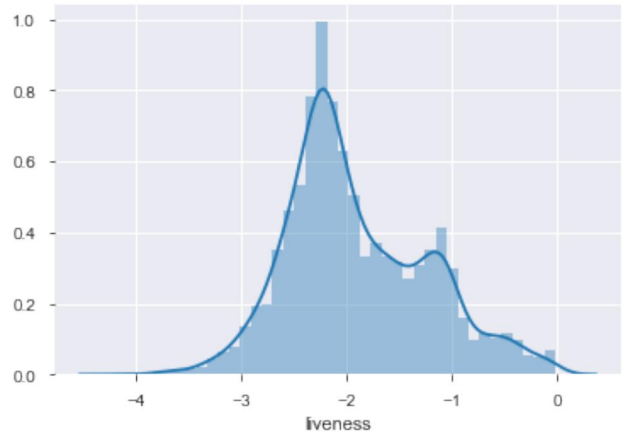


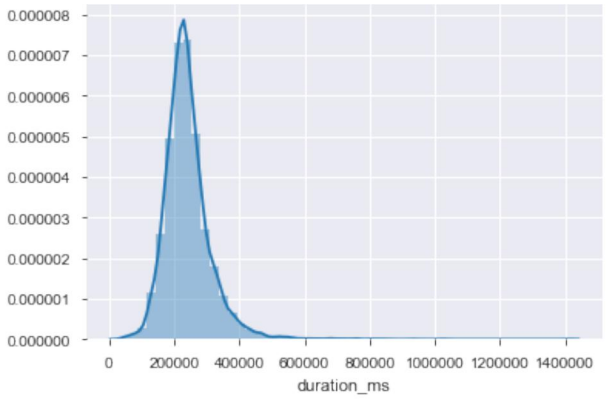
Figure 4: Explanatory variables from Spotify dataset

In this multi plots, we can see that certain distributions need to be cleaned up. In particular, we can see that acousticness (second from top left), instrumentalness (last first from top right), liveness (first from bottom left) and duration (third from the bottom left) have very unbalanced distribution.

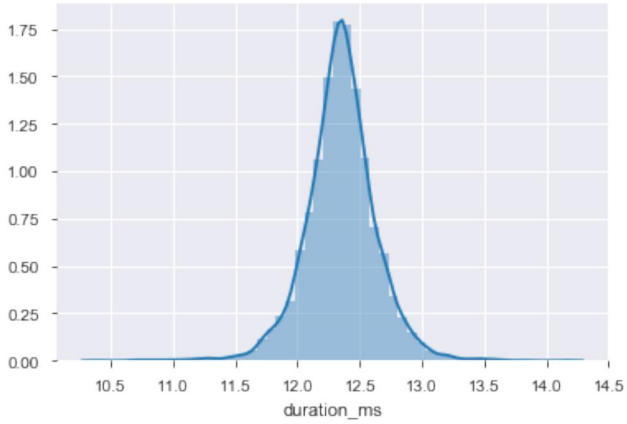
To tackle this issue, we consider scaling the distribution for liveness. Using logarithm, we get a far better distribution :



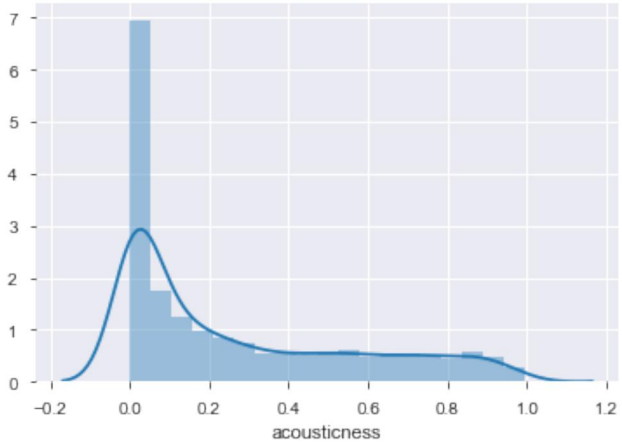
Regarding the feature giving the duration of the song, using the logarithm scale could be a good thing :



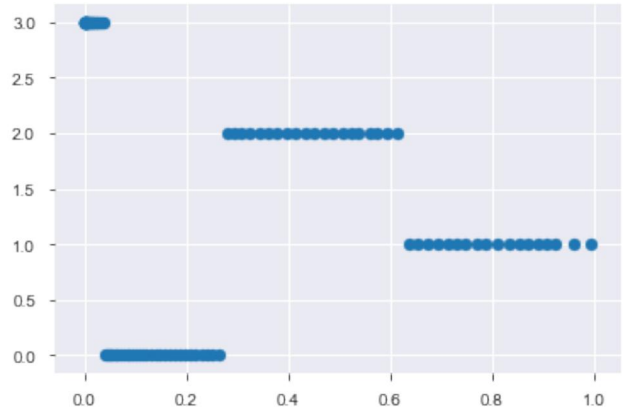
Indeed :



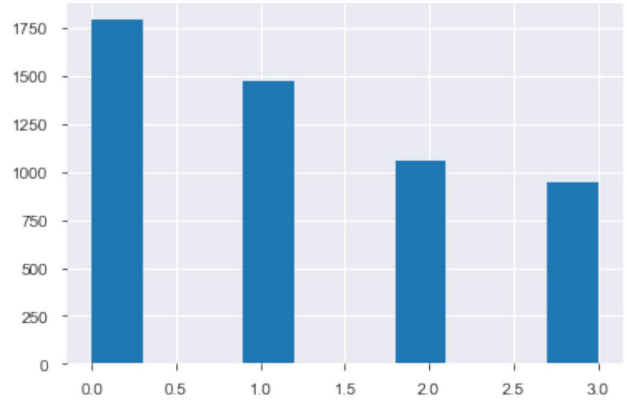
The distribution of acousticness as shown below is quite unbalanced and seems to require a categorisation.



To do so we split the feature into 4 categories, basing our decision on the graph above, and thanks to a k-mean algorithm:



Thus, we get a quite balanced discrete distribution for this variable which help us to discriminate more in our future model.



Encoding

In addition to these features, we encoded the name of the artist, which obviously plays a role in the popularity of a song.

This operation completely modifies the dimensionality of our data as we pass from around twenty variables to 475 features.

We also encoded the other categorical variables such as key or acousticness.

Sum up

Table 2. Variable type and feature engineering method

Variable	**Type**	**Engineering**
acousticness	Musicality	Categorization & Encoding
artist_name	Lyrics/Sentiment	Encoding
compression_rate	Lyrics/Sentiment	Design & Cleaning
danceability	Musicality	Only cleaning

duration_ms	Musicality	Logarithmic scaling
energy	Musicality	Only cleaning
genre	Musicality	Only cleaning
instrumentalness	Musicality	Only cleaning
key	Musicality	Encoding
liveness	Musicality	Logarithmic scaling
loudness	Musicality	Only cleaning
mode	Musicality	Only cleaning
popularity	Regressand	Binary Classification
senti_negative	Lyrics/Sentiment	Design & Cleaning
senti_neutral	Lyrics/Sentiment	Design & Cleaning
senti_positive	Lyrics/Sentiment	Design & Cleaning
senti_total	Lyrics/Sentiment	Design & Cleaning
speechiness	Lyrics/Sentiment	Only cleaning
tempo	Musicality	Only cleaning
time_signature	Musicality	Encoding
valence	Lyrics/Sentiment	Only cleaning

IV. Feature Selection

Minimum Number of Features:
Given the high dimensionality of the data set, it is important to evaluate the minimum number of features that will provide stable and high-quality prediction. Hence the cross-validation score and prediction accuracy score of logistic regression are plotted against the number of features used in the model with stepwise increase in figure 5. It is found that when the number of features used is more than sixty, the cross-validation score and prediction accuracy tend to stabilize at 0.80. Following this finding, sixty features are selected by the selection models.

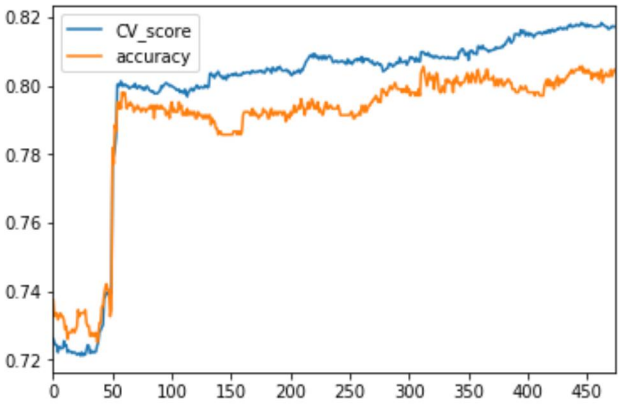


Figure 5: Model accuracy against number of variables

Feature Selection Models:
Due to the mixture categorical and numerical features, four methods are proposed to select the best features. Variance threshold selected features with within-class variance equal to or more than 0.1. This only generates 14 features as majority of the features in this dataset has low variance.

Another method is to use select Kbest based on Chi-square score to select the top sixty features. As Chi-square score is for categorical values, it cannot process negative values. Dataset is scaled between minimum and maximum value to remove negative values.

Random forest (RF) classifier returns top sixty features with higher importance and recursive feature selection (RFE) eliminates five features per step until there are only sixty features.

It is worth to note that different selection models generated different top features. Table 3 shows the top five features selected by each selection model. The most common features are compression rate. instrumentalness., genre_blues.

Table 3. Top five features selected

Variance	SelectKbest	RF	RFE
track_length	instrumentalness	genre_pop	compression_rate
duration_ms	label_acou_3	genre_rock	instrumentalness
loudness	genre_alternative	compression_rate	genre_alternative
tempo	genre_blues	loudness	genre_blues
senti_%+	genre_comdey	duration_ms	genre_classical

V. Model building & Training

Classifier Selection:
Six classifiers are tested including Logistic Regression, Random Forest, K-Nearest Neighbors(KNN), Decision Tree, Support Vector Machine(SVM), AdaBoost, to find out the best performing model.

To establish the baseline, all 475 features are tested with six models first, followed by features selectied by various models. This comparison illustrates the interaction between classifiers and features.

For KNN model, the optimal number of neighbours is set as 12 based on iteration of k values from 1 to 20. After reaching optimal accuracy at k = 12, accuracy starts to decrease.

Metrics for Classifier Evaluation:
To evaluate the performance of the classifiers, data is divided into a training set and test set at 80% - 20% split. For the training set, the cross-validation score at 10-folds is recorded to check the consistency of the performance. For the test set, the prediction accuracy is recorded. ROC (receiver operator characteristic curve) is plotted which true positive rate is plotted against false positive. AUC (area under the curve) is calculated for test set too.

High cross-validation score and prediction accuracy score indicates better performance model. AUC from ROC represents the ability of the model to make correct prediction. As the target is to differentiate the top 25% from the rest of the music, the baseline for this study is 75% accuracy.

RESULTS & EVALUATION

Summary of Results:
Random Forest has shown to be the best performing model with the highest accuracy of 0.797 among the six models with all 475 features. Logistic regression achieved the second best prediction accuracy with RFE features. K-nearest neighbour, AdaBoost and support vector machine attained the best accuracy with top 60 RFE features, decision Tree with top 60 chi-square features.

Further examination of logistic regression shows it also has the highest area under the curve from the ROC curve, which indicates it has a higher probability to make accurate predictions.

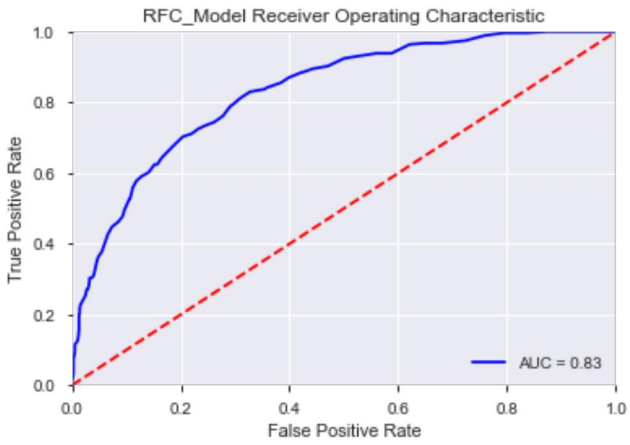


Figure 6: Random Forest ROC survey

Table 4. Best performing features

Classifier	Best feature	Accuracy
LogisticRegression	RFE_features	0.793
RandomForest	all_features	0.797
K-Nearest Neighbors	RFE_features	0,786
Decision tree	Chi_2 features	0,792
Support Vector Machine	RFE_features	0.791
AdaBoost	RFE features	0.787

10-fold Cross-Validation:
10-fold cross-validation is performed for all six models with 475 baseline features. Support vector machine has the lowest variance hence most consistent performance. Whereas K-nearest neighbour has shown to be least stable model with the highest variance from the 10 folds cross-validation score.

Table 5. 10-fold cross validation with 475 features

Classifier	Mean CV score	Variance
LogisticRegression	0.737	3.952e-7
RandomForest	0.806	3.531e-4
K-nearest Neighbour	0.707	9.188e-4
DecisionTree	0.796	2.267e-4
Support Vector Machine	0.726	3.953e-7
AdaBoost	0.807	2.321e-4

Receiver Operator Characteristic Curve：

The ROC curve of the best performing feature for each model is as below. Logistic regression and random forest have the highest AUC = 0.83. Ada boosting have attained similar AUC = 0.81, followed by KNN & SVM and decision tree has the lowest AUC. For logistic regression and random forest, their ROC performance is consistent with prediction accuracy. However, for KNN, their AUC and prediction accuracy is conflicting, it may indicate that the model is less stable.

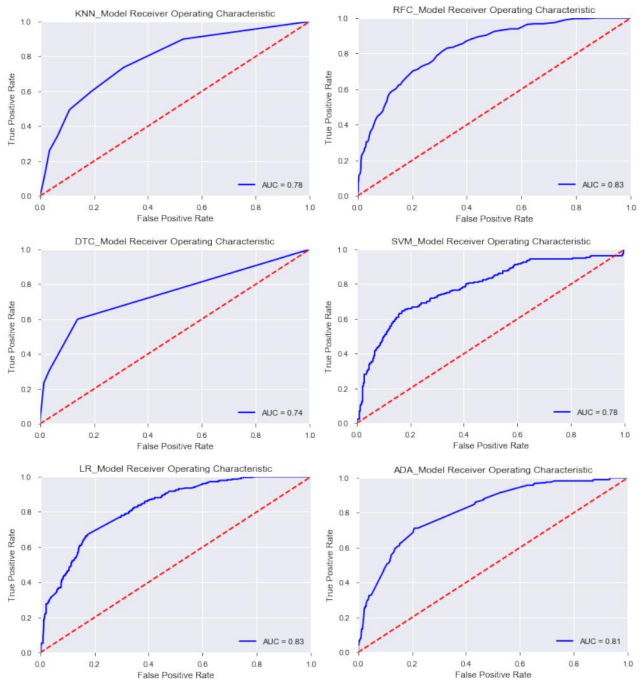


Figure 7: ROC curver for KNN, Random Forest, Decision Tree, SVM, Logistic regression and Ada boosting.

CONCLUSION

Song popularity can be predicted using binary classification at 0.797 accuracy with random forest with all feature used. A similar level of accuracy can be achieved with much fewer features using feature selection method. With a larger size of the database, such dimensionality reduction will be beneficial.

As we initially planned to also implement some regression models, we finally decided to focus on a classification problem as this latter seemed to have better business application. Indeed, a music producer or a studio is probably more interested by broad indicators like “this song will be popular” or “this song will not be popular” rather than a thinner measure that would not bring substantial additional information.

FUTURE WORK

Our project mainly focused on song popularity at a point in time. We explored what factors makes a song popular in this day and age. What was outside of the scope of the project is the changing music taste of the masses. Over time, people’s preferences change, and what is popular now will not necessarily be popular tomorrow. It will then be useful to look at song popularity as a time series, to analyze how music preferences have changed over the years.

Another thing that can be explored is a combination of the work that we have done and some natural language processing algorithms. There may be more features in the text that can be explored for the song popularity prediction to be improved.

REFERENCES

[1] *https://pudding.cool/2017/05/song-repetition/*

[2] *The figures are from the IFPI Global Music Report 2019*

[3] Dhanaraj, R. and Logan, B. (2005). Automatic Prediction of Hit Songs. In *6th International Conference on Music Information Retrieval*, (London, UK), 488-491.

[4] Reiman, M. and Örnell, P. (2018). Predicting Hit Songs with Machine Learning.

[5] Nijkamp, R. (2018). Prediction of product success: explaining song popularity by audio features from Spotify data. In *11th IBA Bachelor Thesis Conference*, (Enschede, The Netherlands)

[6] Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., & Komarova, N. L. (2018). Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society open science, 5(5)*, 171274. doi:10.1098/rsos.171274

[7] https://developer.spotify.com/documentation/web-api/reference/tracks/get audio-features/

[8] https://kvsingh.github.io/lyrics-sentiment-analysis.html

1 Final Report **45 / 50**

- **0 pts** Correct
- **1 pts** Click here to replace this description.
- **2 pts** Click here to replace this description.
- **3 pts** Click here to replace this description.
- **4 pts** Click here to replace this description.
- ✓ **- 5 pts** **Click here to replace this description.**
- **6 pts** Click here to replace this description.
- **7 pts** Click here to replace this description.
- **8 pts** Click here to replace this description.
- **9 pts** Click here to replace this description.
- **10 pts** Click here to replace this description.
- **12 pts** -12
- **0 pts** Click here to replace this description.