Music Genre Classification Based on Lyrics

Mingrui Zhang
Northeastern University
zhang.mingr@northeastern.edu

Abstract

This project aims to analysis different kinds of music genres and build a model to classify music genres based on lyrics. In the project, we have a dataset that contains lyrics of songs with three labels: Rock, Pop and Hip Hop. We do explanatory data analysis and draw plots and word clouds to show and analyze music genres. In addition, we build four machine learning models: Random forest, Linear SVM, Logistic Regression and stacking classifier for multiclass classification. Finally, we can conclude that Hip Hop is the most distinctive music genres among these three genres, which can get the highest recall score to around 0.8. And stacking classifier achieves the best performance.

Introduction

A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions. It is to be distinguished from musical form and musical style, although in practice these terms are sometimes used interchangeably. It can vary from elegant classic music to fast beat modern music. Usually we will assume that rhythm and instruments are strong effected by music genres. But from Wikipedia, it says that A music genre or subgenre may be defined by the musical techniques, the cultural context, and the content and spirit of the themes. Also, Lyrics can be different from its length. It can vary from the frequency and complexity of words in a song as well. It seems like there can be some relationship between lyrics and music genres. This is how I start this project. Also, there are two questions that I try to answer for this project: What are the differences and similarities of these three genres on lyrics? Can they be classified by machine learning techniques?

Music genres classification based on lyrics is the beginning and important step in many music related research and application fields. For example, it is widely combined with music information retrieval. Music information retrieval is interdisciplinary science of retrieving information from music. We can gain information from the same music genre with its own similarities on lyrics. It can also be used in recommender system. Recommender system is able to make a quick and accurate recommendation to users from their previous preference for music genres and lyric styles when it is combined models classified music genres. Music genres classification based on lyrics is a part of song lyric generator as well. Song lyric generator can generate the full lyrics of a song based on models that learn from thousands of songs. It is highly needed to generate lyrics with different music genres.

In this project, we classify music genres only based on lyrics. It is not easy work because as we all know that much information of music genres is in audio data. But with only lyrics, we can investigate more deeply about the similarities and differences among music genres. Thus, we would implement data analysis skills to analyze data and use supervised machine learning algorithms to build models that try to classify three music genres. Finally, we draw some conclusions from comparing results on different models and music genres.

Background

In the project, Rock music, Pop music and Hip hop music are the music genres to be classified. Rock music originated in the United States in 1950s. It is famous for its energetic performance, fast beat rhythm and electric instruments. Pop music is a kind of commercial music whose main purpose is profit. It contains many music genres: Jazz, Rock, Blues and other commercial music genres first appeared in 20th century. It has short structure, easy-read content and catchy melodies. Hip hop music formed as music in 1970s in New York City.

It is popular for rapping which is a rhythmic and rhyming speech.

Lyrics is a kind of text data. Therefore, in this project, we need to deal with a multiclass text classification problem. For feature engineering on text data, we use TF-IDF method to convert text data into numerical vectors. TF-IDF means "Term Frequency times Inverse Document Frequency". It can evaluate relativity of a word to a document in a bunch of documents. Suppose we have a corpus of documents D. For a word t in document d we compute:

Term frequency:

$$tf(t,d) = \frac{f_{t,d}}{|\{t' \in d\}|}$$

Inverse document frequency:

$$idf(t) = \log \frac{|D|}{|\{d \in D | t \in d\}|}$$

$f_{t,d}$ is the raw count of a term in a document.

$|\{t' \in d\}|$is the number of terms in document d.

$|D|$is the number of total documents in D

$|\{d \in D | t \in d\}|$ is the number of documents that contain the term t.

Then TF-IDF can be calculated as:

$$tfidf(t,d) = tf(t,d) \times idf(t)$$

In this project, we choose to use logistic regression, LinearSVM, random forest and stacking classifier to build a model.

Logistic regression is a binary linear classifier. The logistic function is

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where $z = \ln \frac{P(y=1|x,w)}{P(y=0|x,w)}$.

The output of logistic regression can be interpreted as the probability of belonging to the positive class. Logistic regression is very efficient to train and easy to implement. Also, because it is a linear model, it is hard to be overfitting.

SVM is also a binary classifier. It aims to find the hyperplane with maximum margin for support vectors. Apart from linear SVM, it can also use kernel trick to computing inner products in feature space without actually transforming nonlinearly separable data into higher feature space. But in the project, we choose only linearSVM because the dimension of origin feature space is over a thousand which will cause much time to computing when fitting data.

Since logistic regression and linearSVM are both binary classifiers, we will apply OVO method to achieve multiclass classification. In OVO, every classifier will be trained to classify two classes from the whole classes. And OVO will make the final decision by voting.

Random forest is a kind of bagging methods. It is built by a number of decision trees. Every tree in random forest is trained by random subset of training samples with replacement. Random forest will make final decisions by voting as well.

Stacking method is an ensemble method which combines base classifier and build a classifier based on the prediction of base classifiers and true labels.

Project Description

In this project, we use Python as programming language, jupyter notebook as platform and import libraries numpy, pandas to tidy data, seaborn to do data visualization, NLTK to process text data and sklearn to train models.

Dataset

Dataset for this project is from Kaggle. The source URL is https://www.kaggle.com/neisse/scrapped-lyrics-from-6-genres . This dataset is scraped from the Brazilian website Vagalume. It has two datasets: artists-data.csv and lyrics-data.csv. The artists-data.csv has six variables: Artist, Songs, Popularity, Link, Genre, Genres. The lyrics-data.csv has five variables: ALink, SName, Slink, Lyric, Idiom. The artists dataset is mainly about the information of artists. and the lyrics dataset is about information of each song including lyrics. The dataset is not formally structured for classification as there are two datasets needed to merge together, some duplicates and unformatted text. Thus, we need to clean and preprocess this dataset before modeling.

First, we drop the duplicates of Link variable in artists dataset and Slink, Lyric variables in lyrics dataset to make sure that the primary key is unique and lyrics of a song only appears once. Then we merge two datasets on the primary key of Link. We extract songs with only English language and keep only Rock, Pop and Hip Hop for Genre. Finally, remove all the irrelevant variables. The dataset after removing has two variables: Lyric and Genre.

Then we find that there are some samples in lyric variable are not lyrics but numbered musical notation. We drop these data using regular expression. Further we lower all the letters. We apply RegexpTokenizer to keep only English words and PorterStemmer to stem all the words in lyrics with NLTK package.

The dataset has 86294 samples. We count the proportion for each genre in data set (Figure 1). In this data set, there are over 50% samples are Rock and only about 15% samples are Hip Hop music. Thus, there is a data imbalance issue in this data set that we need to deal with when we do modeling part.

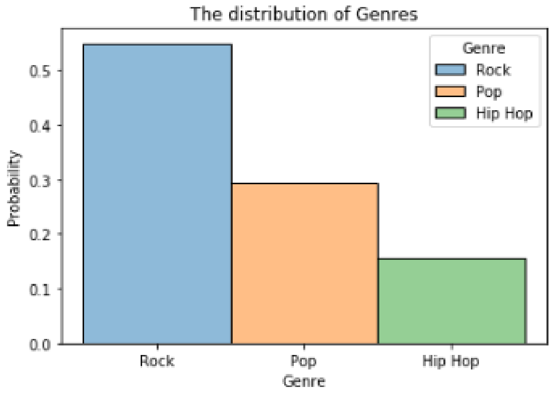


Figure 1: Distribution of Genres

Explanatory Data Analysis

After preprocessing, we do some Explanatory Data Analysis to deeply understand this data set and the problem. We apply python and seaborn package to accomplish data visualization.

First, we evaluate word count of each song and create a new variable called ‘word_sum’ to store data. Then we draw a histogram of word count with three music genres (Figure 2). It can conclude that Hip hop music tends to have more words in a song. Other two music genre have similar word count. We speculate that rap which is a speech form widely used in Hip hop music and speaking lots of words in a short amount of time causes this phenomenon that Hip hop have more words in a song on average.

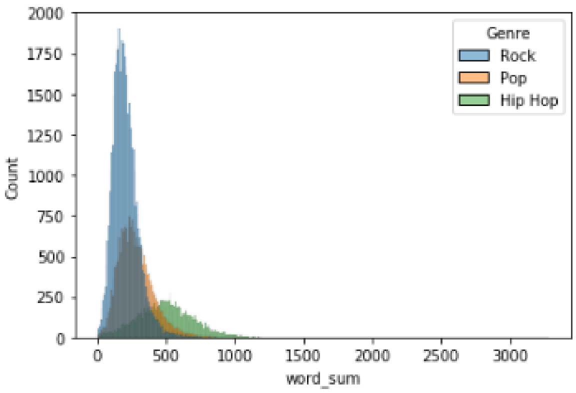


Figure 2: Histogram of word count for each genre

Then we append lyrics in each genre together to form text data of three music genres and draw three wordclouds of each genre with unique icon to represent a broadly general view of music genres in lyrics (Figure 3,4,5). A word cloud is a picture filled with words which have top highest frequencies in a text data. The larger the word is, the higher its frequency is. Three icons which represent music genres are a rock and roll gesture, a CD and a skateboard. As we find among three wordclouds, many words appear and have very high frequency like the word: *Love*, *Know*, *Like* and *Got*. Also, Hip Hop music has more unique words comparing with other two music genres.

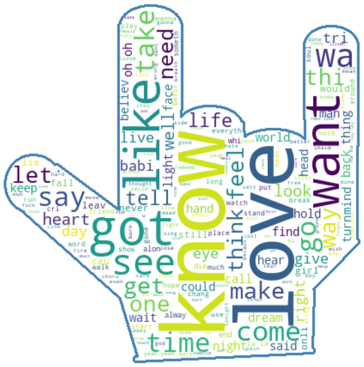


Figure 3: Word cloud of Rock

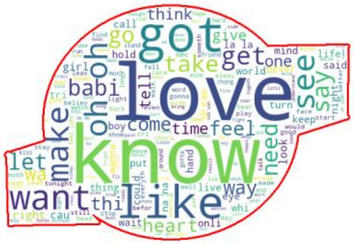


Figure 4: Word cloud of Pop

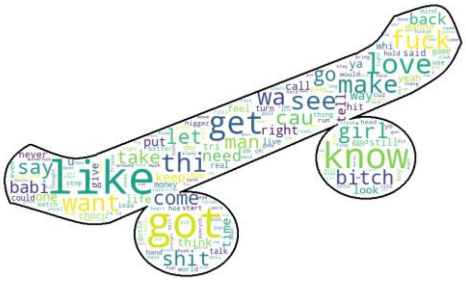


Figure 5: Word cloud of Hip Hop

Modeling and results

We first split data into 80% training data and 20% test data. Then we use TFIDF vectorizer to convert text data into numerical vectors. We set max_feature= 5000 in TfidfVectorizer function. If we don’t set max_feature, dimension of the whole feature space will be over 30,000 which may occupy a large memory space and consume long computing time when we do modeling. Thus, setting max_feature to a proper number can drop some redundant features and reduce ineffective memory storage.

Next, to deal with data imbalance issue, we decide to use under sampling method from imblearn package. The main reason why we choose under sampling method other than over sampling or SMOT is due to the limit of my computer memory. After under sampling, each genre has 10827 samples. Then we spare 20% of training data as validation set.

We train three models: logistic regression, linearSVM and random forest on training data. Since logistic regression and linearSVM can only deal with binary classification, we apply OneVsOneClassifier to accomplish multiclass classification. In this situation, using OneVsOneClassifier will have no data imbalance issue and large complexity comparing with OneVsRestClassifier. Then we want to do hyperparameter tuning on validation set. We tune the hyperparameter C (Regularization parameter) on logistic regression and LinearSVM and n_estimators which means the number of trees on random forest. The distributions of hyperparameters with score are listed below (Figure 6,7,8).

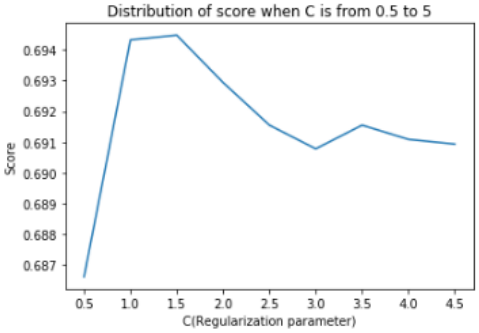


Figure 6: Distribution of score when C is from 0.5 to 5(logistic regression)

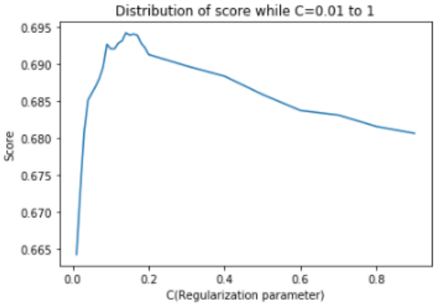


Figure 7: Distribution of score when C is from 0.01 to 1(linearSVM)

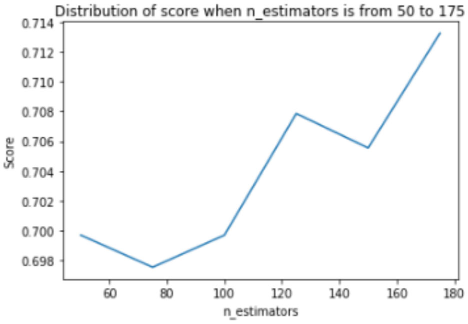


Figure 8: Distribution of score when n_estimators is from 50 to 175(random forest)

Score is the mean accuracy of given data and labels. To find best hyperparameters, we need to find hyperparameters which can achieve the highest score. Thus, we set C=1.3 in logistic regression model and C=0.13 in linearSVM model. We set n_estimators=125 in random forest. Random forest model doesn't reach the best performance when n_estimators=125. However, it is a good budget when considering the improvement of score with modeling time and memory consuming. We find that scores for three models on validation set are all around 0.7 which is not a satisfying result.

Then we apply these three models into test data with best hyperparameters that we choose to find out how well models can perform on the test data. The test data has 17243 samples which is larger than training data. Thus, test data can be broader than training data and we are able to achieve a more general result on test data. The Following are classification report and confusion matrix of three models on test data (Figure 9,10,11)

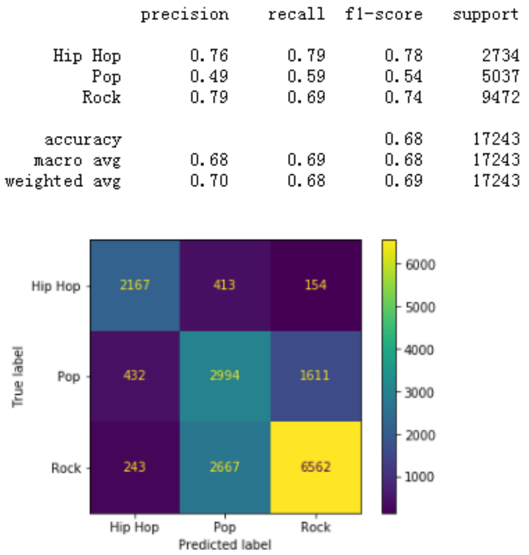
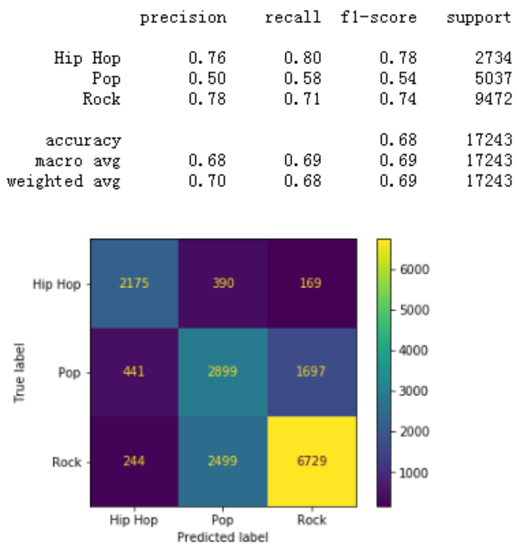
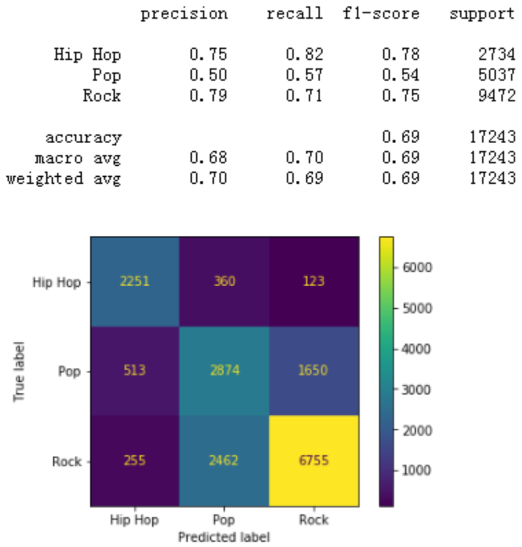

```
              precision    recall  f1-score   support

     Hip Hop       0.76      0.79      0.78      2734
         Pop       0.49      0.59      0.54      5037
        Rock       0.79      0.69      0.74      9472

    accuracy                           0.68     17243
   macro avg       0.68      0.69      0.68     17243
weighted avg       0.70      0.68      0.69     17243
```

Figure 9: Classification report and confusion matrix of logistic regression



```
              precision    recall  f1-score   support

     Hip Hop       0.76      0.80      0.78      2734
         Pop       0.50      0.58      0.54      5037
        Rock       0.78      0.71      0.74      9472

    accuracy                           0.68     17243
   macro avg       0.68      0.69      0.69     17243
weighted avg       0.70      0.68      0.69     17243
```

Figure 10: Classification report and confusion matrix of linearSVM

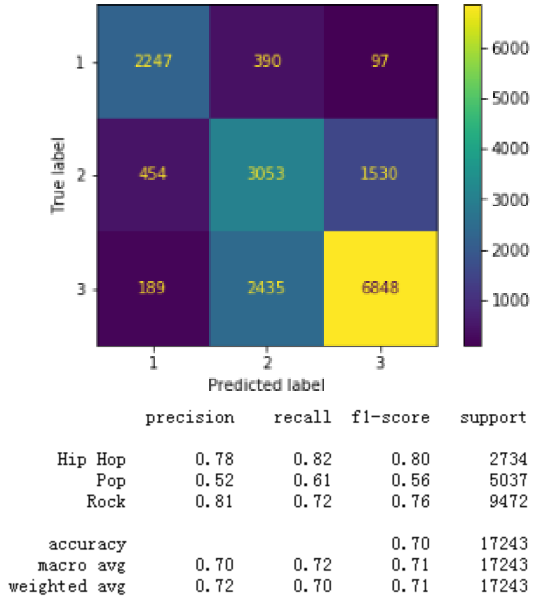


```
              precision    recall  f1-score   support

     Hip Hop       0.75      0.82      0.78      2734
         Pop       0.50      0.57      0.54      5037
        Rock       0.79      0.71      0.75      9472

    accuracy                           0.69     17243
   macro avg       0.68      0.70      0.69     17243
weighted avg       0.70      0.69      0.69     17243
```

Figure 11: Classification report and confusion matrix of random forest

From figure 9,10,11, we can find that the performance of three models is similar to each other while random forest model performs slightly better than other two models. For classification report, we prefer to focus more on recall score to see if true labels can be classified correctly. Among three music genres, Hip hop music can reach about 0.8 of recall score while Pop music can only reach about 0.6 of recall score. Hip hop music is the music genre that can be classified most correctly by these three models. And Pop music can is hard to be classified correctly by these three models. In confusion matrix of three models, we can find that test data is not balanced as what original dataset looks like. The proportion of Rock music is over 50%. Also, it is clearer to reveal that almost 30% of Pop music is classified as Rock music and about 28% of Rock music is classified as Pop music. Thus, there is some trouble that we need to deal with about how to correctly classify Pop music and Rock music.

We try to use ensemble method and train a stacking classifier to deal with this problem. We build a stacking classifier using StackingClassifier. Three above models with best hyperparameters are the estimators of StackingClassifier. In addition, we set logistic regression as final classifier of StackingClassifier. It means that the results of estimator and true labels will feed into a logistic regression model to make predictions. We first train stacking classifier with training data and then implement the classifier on test data to gain results. Below is the figure of result on test data (Figure 12).



```
              precision    recall  f1-score   support

     Hip Hop       0.78      0.82      0.80      2734
         Pop       0.52      0.61      0.56      5037
        Rock       0.81      0.72      0.76      9472

    accuracy                           0.70     17243
   macro avg       0.70      0.72      0.71     17243
weighted avg       0.72      0.70      0.71     17243
```

Figure 12: Classification report and confusion matrix of stacking classifier

If we compare classification report of stacking classifier with classification report of random forest, it can conclude that model performance has improved after using ensemble method though it is not much. Recall score of Pop music improve the most, from 0.57 to 0.61. But in confusion matrix, the misclassification problem between Pop music and Rock music still exist.

Conclusion and Future Work

In this project, we find some differences and similarities of music genres, using explanatory data analysis and machine learning algorithms. From the three wordclouds and the distribution of word count, we can say that Hip hop is the most distinctive music genre among three music genres. It usually has more words in a song than other two music genres. Also, it has many unique words comparing to Pop and Rock music. In all three music genres, the highest frequency words in wordclouds are including *love*, *like*. So, it seems like love and romance may be the most frequent topic and theme. It matches to our expectation and common sense.

In modeling part, among three base classifiers, random forest can get the best performance. But in fact, precision score and recall score of three models are very close. Thus, it is hard to assert that random forest can perform better than the other two models when meeting new data from real world. The ensemble method does slightly improve the performance but it is not effective due to its long fitting time. All the four models' performance is not very satisfying for they all have difficulty on classifying Rock music and Pop music. The reasons I concern are that there is not a very clear decision boundary between Rock music and Pop music since Pop music usually contains some Rock music. Also, as we use TFIDF method to represent feature space, we can only hold the information about the popularity of words but lack of information between words to words or sentence to sentence. There may have some differences on sentences because different music genres could have different rhythm structure associated with lyrics. Thus, we can use other methods like n-grams to convert text data in the future work. Due to the limit of time, I would like to choose models that are easy and efficient to train. Since the performance of four models is not satisfying, we can use more complex models such as LSTM, CNN to achieve better performance in the future. Also, oversampling or SMOTE method can take place of under sampling if we can have more computer memory space to store the data. It is highly possible that implementing oversampling or SMOTE can have better results on test data since models use more data to train.