DSCI 632 Project:
EDM Songs Classification with Pyspark

Dayun Piao[1], Tien Nguyen[2]

dp636@drexel.edu[1], thn44@drexel.edu[2]

GitHub Repository.

Abstract

The team consists of two team members with unique backgrounds and skill sets who will be using this opportunity to polish their data science skills specifically using Pyspark, MLlib and derive conclusive results from the data. The dataset is a collection of 21,000 electronic dance music from seven unique categories. Each row of the dataset represents a song, and each row has 13 audio features. The purpose of the project is to automatically classify the genre of a given song using machine learning algorithms for easier organizing the song library. Random Forest, Naive Bayes, Logistic Regression, and One Vs All machine learning algorithms were trained and tested on the dataset. The statistics results, as well as confusion matrices, were generated to compare between the models. Random Forest model produced the highest accuracy, precision, recall, and F-measure equally at 0.84. Additionally, drum and bass (dnb) genre was labeled most accurately among all seven categories.

1 Introduction

Electronic dance music (EDM) is any form of music that is produced electronically with digital and analog equipment. It is the combined term for all genres within the dance music space. It has various genres such as drum and bass, house, hardstyle, and techno. EDM has risen from the clubs to the pop charts, and it has been used popularly in large-scale performances, such as music festivals. As the popularity of EDM increases, more new listeners would expose to various EDM genres that they have never heard before. Discovering new music or listening to an EDM song for the first time could be extremely exciting. However, it could also be confusing for listeners who want to organize their music library for new songs. This project explores the ability of four machine learning algorithms, Naive Bayes (NB), Random Forest (RF), Logistic Regression (LR), and One Versus All (OVA), to classify a given song. With high accuracy and precision, it automatically recognizes a song's genre and allows users to organize their EDM library better (Wolf 2020).

2 Dataset

The dataset is a collection of 21,000 EDM songs from seven unique categories including hardstyle, trap, techno, psytrance, techhouse, drum and bass, and trance. Each row of the dataset represents a song, and each row has 13 audio features. The audio features are the measurements of danceability, energy, key, loudness, mode, acousticness, instrumentalness, speechiness, liveness, valence, tempo, time signature, and duration. The dataset is completely balanced with 3,000 observations for each category. The dataset was originally constructed using Spotify's API by Travis Wolf (Wolf 2020). The final dataset which was used in this project was generated on February 2022.

3 Exploratory Data Analysis

3.1 Basic Statistics

summary	danceability	energy	key	loudness
count	21000	21000	21000	21000
mean	0.6024792	0.8716178571428513	5.600952380952381	-5.869760619047621
stddev	0.14494403095623315	0.11602636326459131	3.623975153312106	2.8627058998246024
min	0.0891	0.188	0	-26.172
max	0.988	1.0	11	3.108

Figure 1: Snippet of Data Statistics

From Pyspark datafrme, df.describe() method was used to generate the overall columns statistics.

3.2 Class Balance

When grouping the dataset by genre types, the counts for each genre is consistent through out all. So there is no need to perform data balance task.

genre	count
hardstyle	3000
trap	3000
techno	3000
psytrance	3000
techhouse	3000
dnb	3000
trance	3000

Figure 2: Data Balance

3.3 Missing Values

From Pyspark dataframe, df.na().drop() was used to drop out rows with the missing values. The dataset is very clean and does not contain any missing values.

3.4 Feature Distribution

After observing the features distribution plots in Figure 3, among the 13 features, three of them are not very helpful on distinguish each genre group: Key, Mode, and time_signature; 3 of them are almost normalized and the mean of the each genre are different from on another: Duration, loudness, dancebility. So, dropping those 3 features which are not helpful and use rest 10 of the features to predict the song genre. Now, after applied these steps, there is an clear understanding of the dataset. The total 10 features selected based on feature distribution plot.

4 Methodology

4.1 Data Preprocessing

After running the EDA, it is found that features: 'loudness', 'tempo', and 'duration' need to be scaled between 0 and 1. This is done by using Pyspark.ml.MinMaxScaler().

Ensemble the features including scaled ones and transform the data through Pyspark.ml.feature.VectorAssembler(). Now in the Pyspark dataframe, there is one clean column of 'features' with values between 0 and 1. A map function is also created to be able to convert encoded index to original genre name.

Label column 'Genre' have seven different categorical value and these are converted to ordinal number through Pyspark.ml.feature.StringIndexer().

Select only the 'label' column and 'features' column from Pyspark dataframe and they are ready to feed into models.

Data can be found in the GitHub Repository.

4.2 Machine Learning Models

As mentioned earlier, the four machine learning classifiers that were used in this project are NB, LR, RF, and OVA. NB classifier is a probabilistic machine learning model based on the Bayes theorem (Gandhi 2018). This theorem finds the probability of y happening when x occurs where y represents the final class, which is the genre of the song in this project, and x represents all the measured audio features. LR determines the impact of multiple independent variables presented simultaneously to predict the membership of one or other dependent variable categories (Rawat 2017). The independent variables in this project are the measured audio features, and the dependent variables are the seven EDM genres. RF is an ensemble approach for classification. This classifier builds multiple decision trees by selecting a random set of features. Lastly, OVA is a method that trains N distinct binary classifier, each classifier is designed to recognize a class. In this project, OVA is an extension of LR. It trained seven distinct binary LR classifiers, and each classifier could identify a song genre.

4.3 Efficiency Analysis

Statistics results such as accuracy, average precision, average recall, and average F-measure for each model were calculated to evaluate and compare the classifiers. Precision for each label measures the percentage of classified label that actually belongs to this label. Recall for each label measures the percentage of the correctly identified label. F-measure is a weighted harmonic mean of precision and recall. Accuracy is the amount of correctly identified classifications over the total number of classifications. A confusion matrix for each classifier was also generated to summarize the prediction results with count values which are broken down by each class. Stacked error bars for each class are also plotted for better visualization from the confusion matrix results.

5 Results & Discussion

The summary of statistic results for all four machine learning models are reported in Table 1. For each model, the four statistic results, recall, F-measure, precision, and accuracy, are generally equal since the dataset is balanced. Therefore, we can use accuracy to compare the models. Based on Table 1, the model accuracy are 0.61, 0.69, 0.83, and 0.75 for NB, LR, RF, and OVA, respectively. OVA, which is an extension from LR, shows an increase in accuracy which is from 0.69, LR, to 0.75. Linear logistic regression is popularly used for binary classification problem. This project, however, is a multi-classification with seven different classes. There fore, OVA would be more sufficient than LR in this project. RF model is significantly better than the other three models with roughly equal 0.84 for accuracy, F-measure,

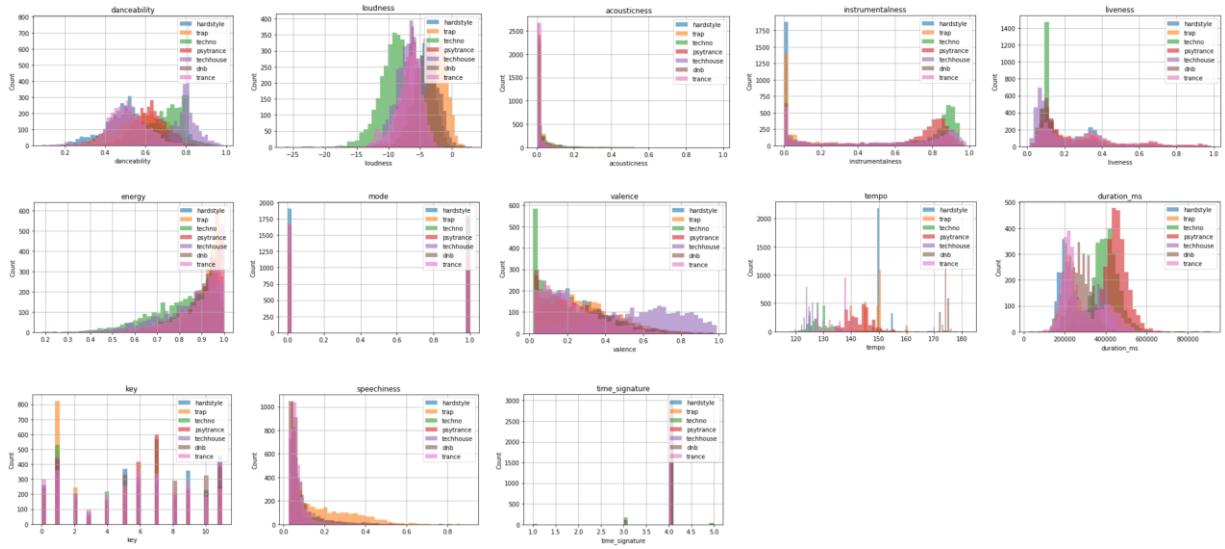

Figure 3: Feature Distribution Bar Plots For Each Feature

Table 1: Statistic Results for The Testing Dataset

Model	Average Precision	Average Recall	Average F-Measure	Accuracy
Naive Bayes	0.64	0.61	0.61	0.61
Logistic Regression	0.72	0.69	0.68	0.69
Random Forest	0.84	0.84	0.84	0.83
One-vs-All	0.75	0.75	0.75	0.75

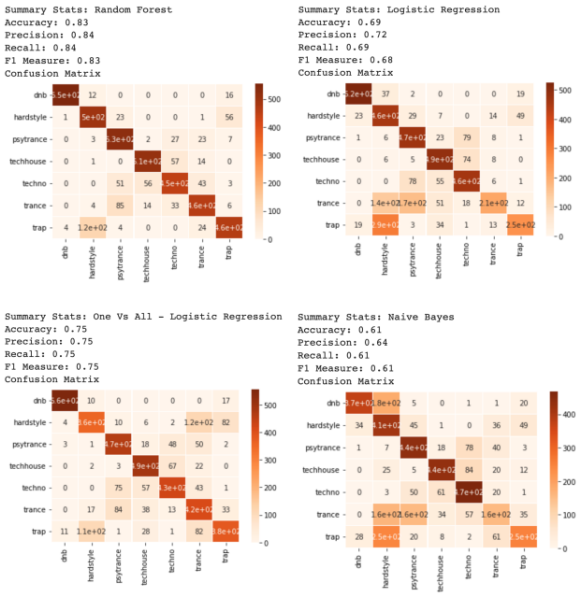

Figure 4: Multi-class Confusion Matrix for each model

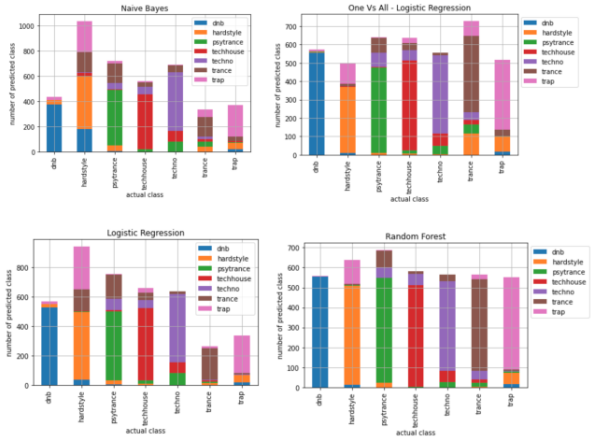

Figure 5: Predicted Error Bar plots for each model

precision, and recall. Suggesting random decision tree classifiers is a good method for classification using this dataset.

The confusion matrix for each model is also generated and is reported in Figure 4. In each confusion matrix, the y axis represents the predicted labels while the x axis represents the true labels. The heatmap color represents the degree of correction for the intersection box. Here, the darker the color of the box, the higher the number of predictions for that box. Overall, the diagonal elements on all matrices have darker colors comparing to other elements in the same matrices which means that the most of the true labels are predicted correctly. The diagonal in RF's confusion matrix has the darkest color with the highest total number comparing to the other three matrices diagonals. This result confirm again that RF has the most accurate predicted labels. Additionally, based on the color distribution, dnb and psytrance genres are confusing each model the least since it has which means these two genre are much easier to be recognized than others.

Predicted error bar plots for each model are also generated in Figure 5. Predicted error bar is a useful extension of the confusion matrix for better visualization of the misclassified classes as a stack bar. Here, each plot has seven columns represent for seven classes or categories. The height of each column is the total number of predicted labels for that class. The more colors a column has, the more misclassified classes it has. The bar plots in NB and LR have more color for each column than RF and OVA. In fact, most of the column bars in RF have the true class colors nearly over 85% of the columns which proves that RF has the highest accuracy overall. Moroever, in all models, dnb genre has less percentage of wrong predicted genre than others which again confirms that dnb is the recognized easily by the model.

6 Conclusions & Future Works

The result from Random Forest is promising on automated classifying the EDM songs based on audio features. The script in PySpark can be used on platform like Google Cloud Platform(GCP)'s Dataproc or similar big data service to conveniently scale up data and reduce time and computing power. PySpark's ml and MLlib are relatively simple and similar to scikit learn library although there are still a number of useful functions/libraries lacking from them. Thus, it would need to create own function/libraries to do the same tasks. For the future work, it is desired to test the run time difference between running the script local;y and running it in GCP's Dataproc. Due to the time limit of this project, it is not included. However, it will be an ultimate score for future work, cloud computing using PySpark to do machine learning tasks.

References

[Gandhi 2018] Gandhi, R. 2018. Naive bayes classifier.

[Rawat 2017] Rawat, A. 2017. Binary logistic regression.

[Wolf 2020] Wolf, T. 2020. Genre classification of electronic dance music using spotify's audio analysis.