Fusemachines Nepal

Analysis, comparison, and optimization of the various classification algorithms on MGC dataset

By
Aseem Priyadarshan Chemjong limbu
Aashish karn
Dipesh Shrestha
Amrit Poudel

May 2023

1. ABSTRACT

In this project, we explore the effectiveness of various classification algorithms for the task of music genre classification (MGC) using a publicly available dataset on kaggle. The objective is to identify the most accurate and efficient algorithm for MGC tasks. We begin by preprocessing and cleaning the data to ensure its quality and relevance. Next, we use multiple machine learning algorithms, including Random forest (RF), Decision Tree, and Extreme Gradient Boosting (XGBoosting), to classify the music genre of the dataset. We use hyperparameter tuning techniques to optimize the performance of the best algorithm identified. We evaluate the performance of each algorithm based on standard classification metrics such as accuracy, precision, recall, and F1 score as our labels of the target class are balanced. Furthermore, we perform a comparative analysis of the algorithms to identify the best-performing algorithm for MGC tasks. Our results indicate that Decision Tree with max depth control and Random Forest algorithms perform the best among all the algorithms tested, with an accuracy of around 82%. Overall, this project highlights the importance of selecting the right classification algorithm for MGC tasks and provides useful insights into the strengths and weaknesses of various algorithms.

2. INTRODUCTION

Music genre classification (MGC) is an essential task in the field of music information retrieval, which involves identifying the genre of a piece of music based on its audio features. MGC has various applications, such as recommendation systems, music streaming services, and music education. With the increasing amount of music data available on the internet, MGC has become more challenging, and machine learning algorithms are often used to automate the process. In this project, we aim to analyze, compare, and optimize the performance of various machine learning algorithms for MGC tasks. We use a publicly available dataset from kaggle, which contains 50,000 audio clips, each of which is of various length, acousticness, danceability, energy, tempo etc up to 16 features and finally belongs to one of ten different music genres. The project consists of several stages. First, we preprocess and clean the data to ensure its quality and relevance. We then use multiple machine learning algorithms mentioned above in the abstract of the project to classify the music genre of the dataset. We evaluate the performance of each algorithm using standard classification metrics such as accuracy, precision, recall, and F1 score. Our analysis can provide useful insights into the factors affecting the performance of MGC algorithms, such as the choice of correct audio features and the quality of the data.

3. LITERATURE REVIEW

The classification of music genres has been a challenging task for a long time due to the subjective nature of music and the overlapping characteristics of different genres. Many researchers have used machine learning techniques to classify music genres in recent years. In this project, we compared the performance of four popular classification algorithms: decision tree, random forest, and XGBoosting, on a music genre classification dataset.

Decision trees have been widely used in various classification tasks due to their simplicity, interpretability, and low computational cost. Random forests are an extension of decision trees that improve their accuracy and reduce overfitting. Logistic regression is a popular classification algorithm that is used in many fields, including music genre classification. It is simple, fast, and can handle non-linear relationships between the independent and dependent variables. XGBoosting is a gradient boosting algorithm that has become popular in recent years due to its high accuracy and ability to handle large datasets.

Several studies have compared the performance of different classification algorithms for music genre classification. For instance, a study by Li et al. (2010) compared the performance of decision trees, support vector machines, and k-nearest neighbor algorithms on a music genre classification task. They found that the decision tree algorithm outperformed the other two algorithms in terms of accuracy and training time. Another study by Fu et al. (2011) compared the performance of different classification algorithms, including decision trees, neural networks, and support vector machines, on a music genre classification task. They found that the support vector machine algorithm achieved the highest accuracy.

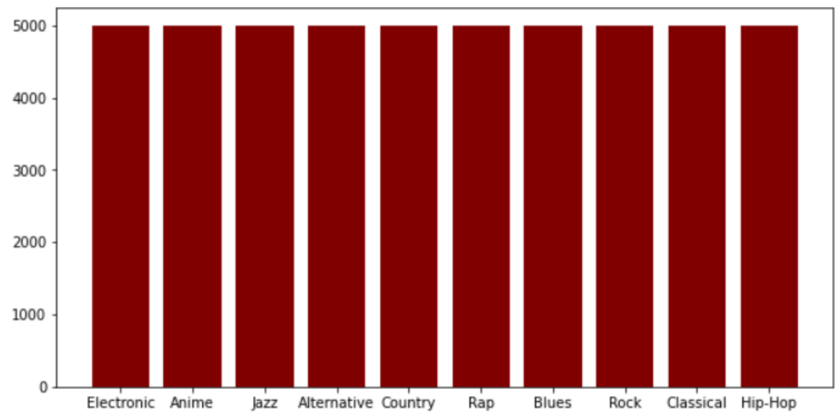
In recent years, XGBoosting has become popular in various classification tasks due to its high accuracy and ability to handle large datasets. For instance, a study by Chen and Guestrin (2016) showed that XGBoosting outperformed other popular algorithms, including random forest and gradient boosting, on various classification tasks.

4. METHODOLOGY

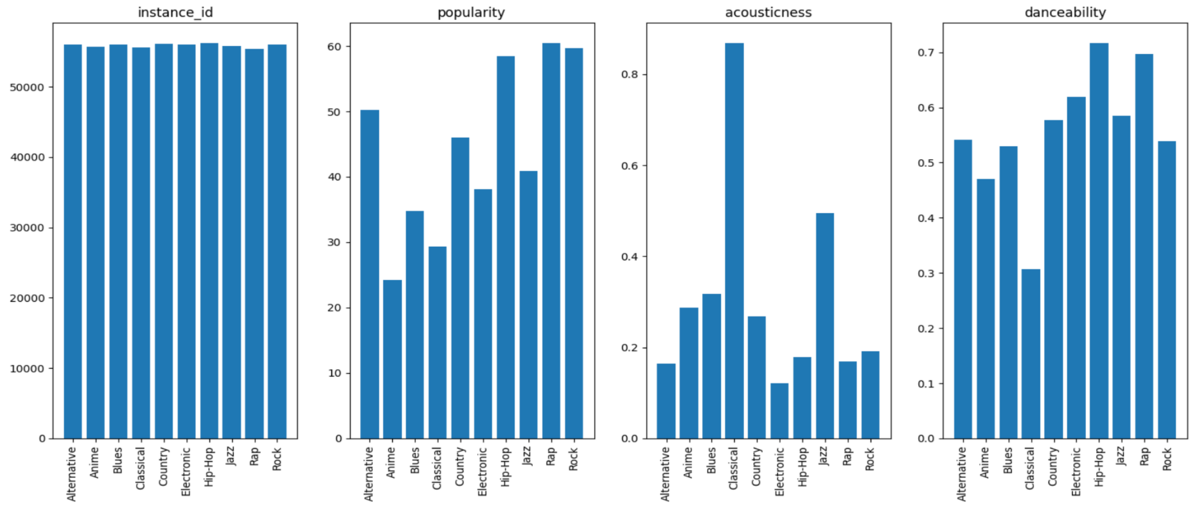
Data Collection: At first we collected the music genre dataset from publicly available source, Kaggle. The dataset found out to be diverse and balanced with an equal number of samples for each genre.

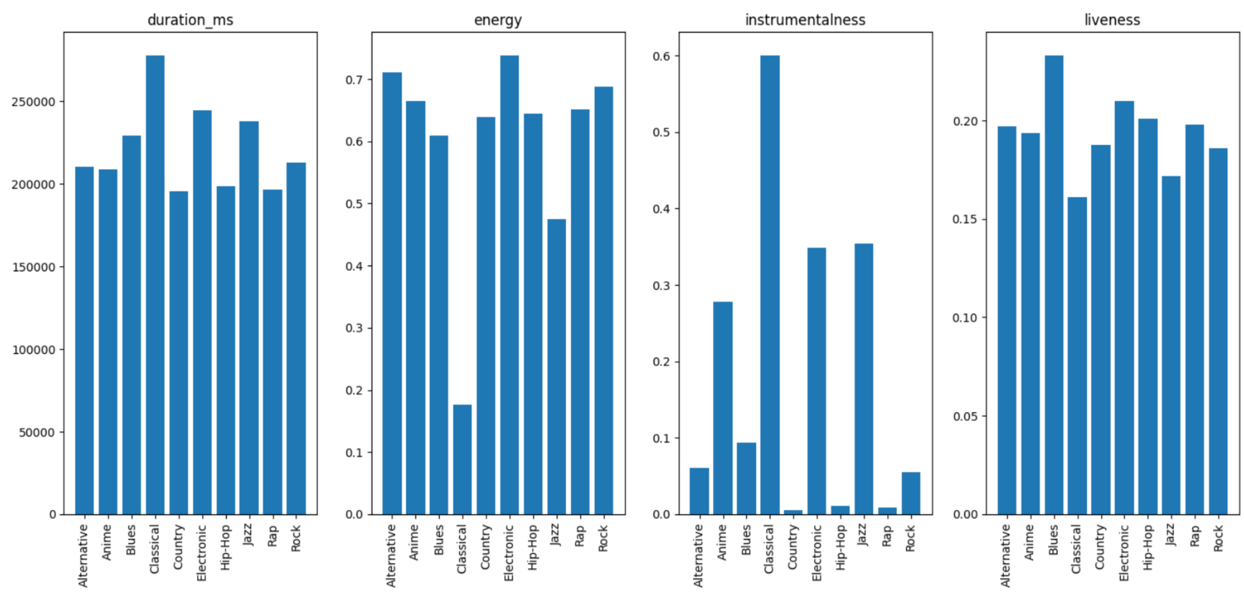
Data Preprocessing: While preprocessing the dataset we first found out the categorical, and numerical features from the total features of the dataset, we then performed the descriptive statistics of data for finding out the mean, variance, standard deviation, quartiles, and min-max values of each features. Further we performed the data visualization like histogram, heatmap, bar-plot, box-plot, plot of features vs target. We then calculated the correlation between features for finding the redundant features. We also calculated the correlation between target class and each feature for finding which feature is highly correlated with the target class. Furthermore, we found the outliers in the data using above mentioned data visualization techniques. We also performed the feature extraction and also made some new features based on currently available features. We also handled the missing values. We also performed the hypothesis testing for finding the less related features for feature extraction. We also performed the encoding in categorical features with the method like one hot encoding, and label encoding.

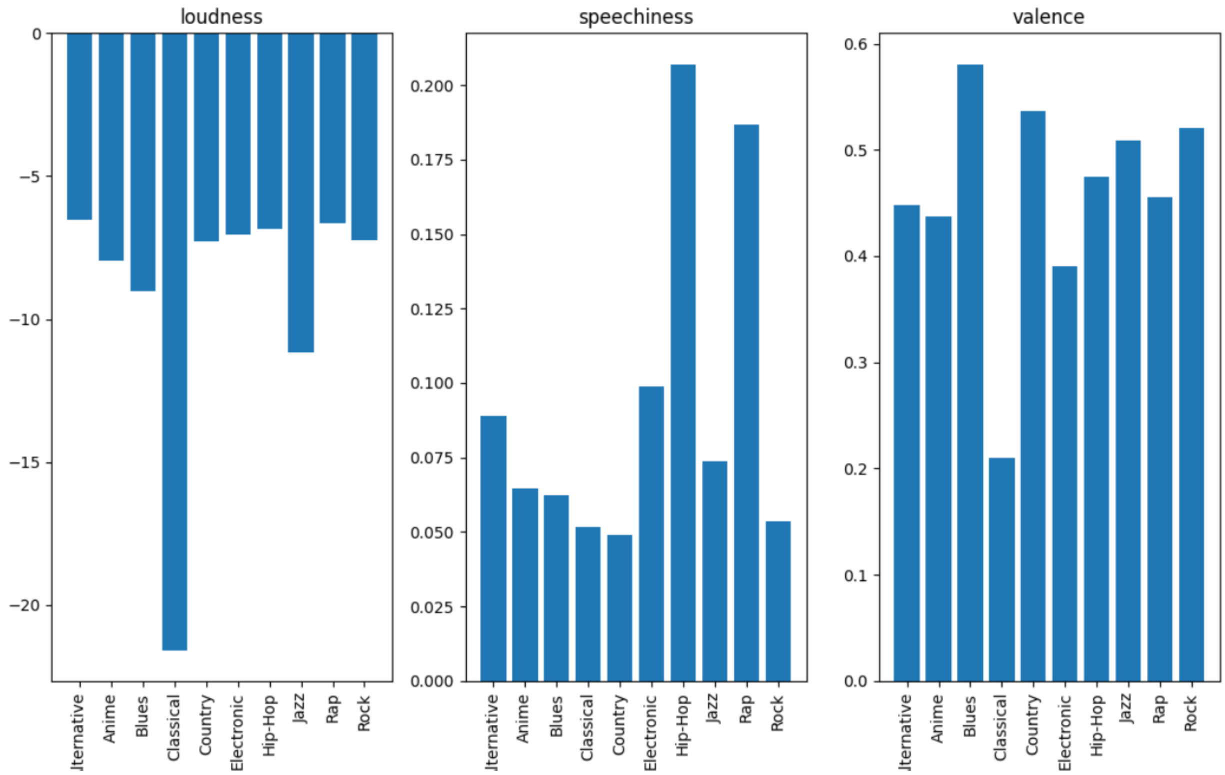
Distribution of data according to Music_genre:



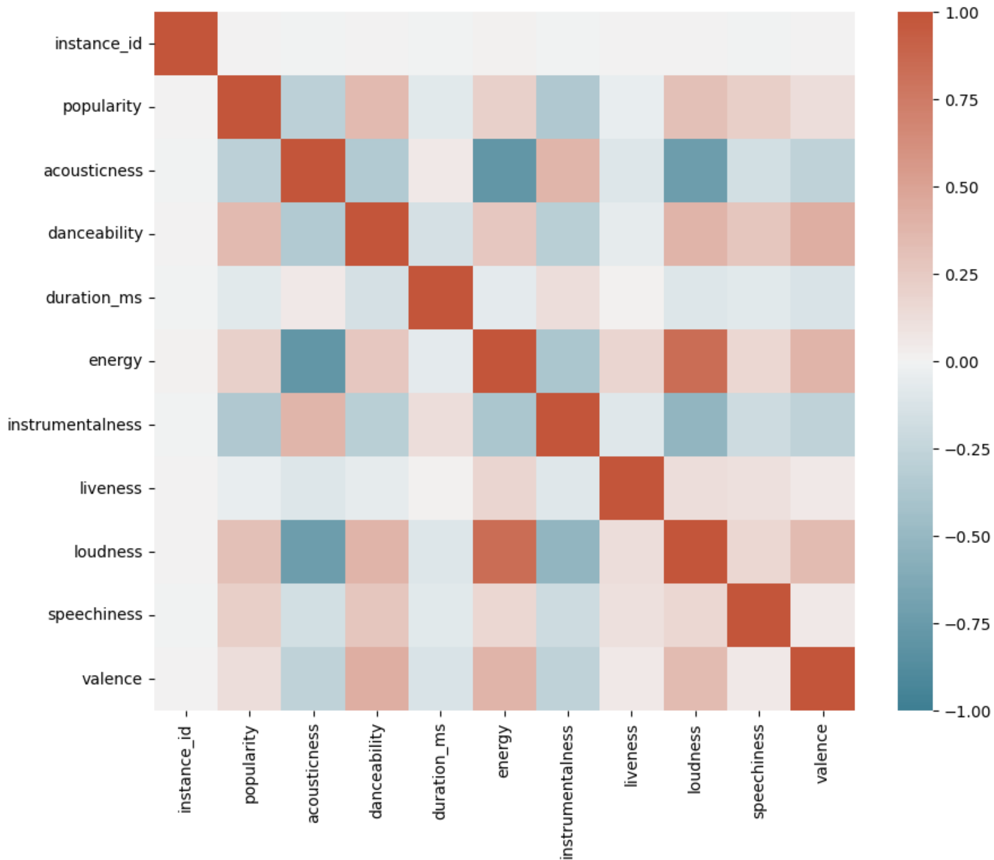
Mean quantity of each features based on Music_genre:






loudness
speechiness
valence
0
−5
−10
−15
−20
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.6
0.5
0.4
0.3
0.2
0.1
0.0
lternative
Anime
Blues
Classical
Country
Electronic
Hip-Hop
Jazz
Rap
Rock

Correlation-heatmap:



Data Splitting: Once the data was preprocessed and features were extracted, we split the dataset into two parts: training and testing. The training dataset will be used to train the machine learning models, while the testing dataset will be used to evaluate the performance of the models.

Model Selection: We trained and evaluated four different machine learning models: Decision Tree, Random Forest, and XGBoosting. These models were selected based on their performance in music genre classification tasks in previous studies (literature review).

Hyperparameter Tuning: We tuned the hyperparameters of the models using grid search or randomized search. The hyperparameters include the maximum depth of the tree, alpha or weakest link determination of the decision tree, number of estimators, learning rate, number of trees, subsample, gamma etc.

Model Evaluation: We evaluated the models based on various performance metrics like accuracy, precision, recall, and F1-score. We also compared the performance of the models using visualizations like confusion matrix.

Results: Finally, we presented the results of the evaluation in a comparative analysis of the models' performance. We discussed the strengths and weaknesses of each model in the result and discussion part of this report and recommended the best-performing model for music genre classification.

5. RESULT AND DISCUSSION

After doing a lot of feature engineering. We are finally able to optimize accuracy along with all other evaluation metrics. We had been struggling to increase the accuracy of various algorithms. Earlier we had been getting accuracy of 45-55% which was not so satisfactory. After that we performed some very critical preprocessing as Label encoding, correlation analysis and various feature engineering. We have used various algorithms such as Decision Trees, Random Forest, and also used various hyperparameter tuning as well as pre pruning and then accuracy increased by 70-80% .

Below is the summarisation of the evaluation matrices we have obtained.

```
Train f1-score =  0.9760338884167254
Test f1_score=  0.9760338884167254
Accuracy: 0.7820228046794018
              precision    recall  f1-score   support

     class 0       0.99      1.00      1.00      3146
     class 1       1.00      1.00      1.00      3148
     class 2       1.00      0.99      1.00      3129
     class 3       1.00      1.00      1.00      3150
     class 4       0.98      1.00      0.99      3140
     class 5       0.98      1.00      0.99      3126
     class 6       0.85      1.00      0.92      3164
     class 7       1.00      0.98      0.99      3165
     class 8       0.98      0.83      0.90      3153
     class 9       1.00      0.96      0.98      3193

    accuracy                           0.98     31514
   macro avg       0.98      0.98      0.98     31514
weighted avg       0.98      0.98      0.98     31514
```

Fig: Decision Tree’s evaluation metrics

```
Train f1-score = 0.84 and Test f1-score = 0.98
The depth of our tree is 9
Accuracy: 0.8198578409595735
              precision    recall  f1-score   support

     class 0       0.77      0.78      0.78      3146
     class 1       0.87      0.99      0.92      3148
     class 2       0.94      0.91      0.92      3129
     class 3       0.98      0.98      0.98      3150
     class 4       0.97      0.83      0.90      3140
     class 5       0.96      0.92      0.94      3126
     class 6       0.59      0.89      0.71      3164
     class 7       0.96      0.91      0.93      3165
     class 8       0.68      0.41      0.52      3153
     class 9       0.79      0.79      0.79      3193

    accuracy                           0.84     31514
   macro avg       0.85      0.84      0.84     31514
weighted avg       0.85      0.84      0.84     31514
```

Fig: DT with depth limitation’s evaluation metrics

```
Accuracy: 0.7835776691840663
              precision    recall  f1-score   support

     class 0       0.77      0.78      0.78      3146
     class 1       0.87      0.99      0.92      3148
     class 2       0.94      0.91      0.92      3129
     class 3       0.98      0.98      0.98      3150
     class 4       0.97      0.83      0.90      3140
     class 5       0.96      0.92      0.94      3126
     class 6       0.59      0.89      0.71      3164
     class 7       0.96      0.91      0.93      3165
     class 8       0.68      0.41      0.52      3153
     class 9       0.79      0.79      0.79      3193

    accuracy                           0.84     31514
   macro avg       0.85      0.84      0.84     31514
weighted avg       0.85      0.84      0.84     31514
```

Fig: Random forest evaluation metrics

6. REFERENCE

1. Li, T., Ogihara, M., & Taniar, D. (2010). Comparative study of decision tree, SVM and K-NN for music classification. In International Conference on Computer Science and Information Technology (Vol. 9, pp. 342-350).
2. Fu, Y., Wang, J., & Wang, Z. (2011). Music genre classification using the support vector machines. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (pp. 5650-5653). IEEE.
3. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785-794). ACM.