Music Genre Classification using Supervised Learning

Name:	**Rishikesh Kulkarni**
Registration No./Roll No.:	19173
Institute/University Name:	IISER Bhopal
Program/Stream:	Physics
Problem Release date:	February 02, 2022
Date of Submission:	April 24, 2022

1 Introduction

The aim of this project was to classify different genres of music using the given data set of spectral features. The data set has 59 features and one target class with 10 different classes. It has 900 samples with no null values. These classes were in textual form and corresponded to different music genres. The classes were not imbalanced. Since the data set had 59 features, it was necessary to select the salient features for classification without compromising on the performance of the algorithm. Therefore, different feature selection techniques were to be used. Another important part of the procedure was using different classification techniques for which different existing classifiers are used.

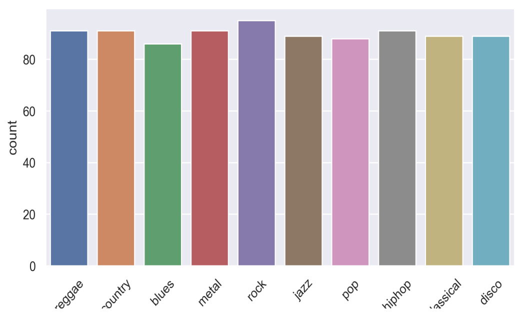

Figure 1: Class count of the ten genres

2 Methods

The main objective was to select important features from the data set. The data was first processed using `MinMaxScaler` and the data points were brought in the range of 0 and 1. The class labels were in text format. To simplify the execution of the procedure, the class labels were converted into numerical form. This was done using `LabelEncoder`[1]. The next step was feature selection. For this, I used Correlation statistics. In this method, features are correlated more than 80 % were dropped from training set. This method was only used on the scaled data only. The features were reduced from 57 to 40. This threshold percentage was kept fixed during the whole procedure. Finally, we have now two data sets which belongs to scaled data and data with high correlation features removed or the Uncorrelated data.

Before the implementation of LDA and χ^2 statistics, the classifiers were to be built with a set of parameters which after gridsearch would give the best performance. For this we used kNN, Support

[1] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn-preprocessing-labelencoder

Vector Machines and Random Forest Classifier which were proposed during the Idea development. Later, we added Logistic regression and Adaboost classifier. The Adaboost classifier contained SVC, NaiveGaussian and DecisionTree classifiers as the ensemble elements. Finally, all these classifiers were compiled in a Dictionary with there respective hyper parameters. These classifiers are now ready for GridsearchCV[2]. I have made a small function which takes the inputs as a training set and target class and would return a pandas data frame with the best model, its cross validated score, and the respective hyper-parameters for which it showed the best performance.

Finally, the feature selection is done using Linear Discriminant Analysis(LDA) and χ^2 statistics. For LDA, both the *svd* and *eigen* solvers are used. The n_components is kept at 9. This is because the sum of variance ratio is maximum at this value. Also, this the maximum value of n_components that we can have for if we increase from 9, we would get the error 'n_components cannot be larger than min(n_features, n_classes - 1)'. From LDA, we would get nine features. Using these as the training set and doing a gridsearch on the classifiers, we will get a data frame with the models with their parameters and score. For χ^2 method, the value of k was taken as 10. Although not in the code, for higher ks the vaidation score was increased but as much as the LDA gave. The codes for high k were not included because the computation time was very high.

After doing the Gridsearch we realized that the time it takes to compute the best model and best parameters is very high. It takes around 6 minutes to return the gridsearch results. Instead of using Gridesearch, we can use `RandomizedSearch`[3]. After using, the Randomized Search, the wall time significantly reduces from around 6 min to around 1.5-2 min. Moreover, there is no much compromise in the performance of our classifiers. From the two techniques we have used, it is observed that LDA performs best for feature selection. We shall continue with this technique for further analysis and evaluation.

3 Evaluation Criteria

In this section, the Precision, Recall and F-measure are showed for the best models. These were obtained from LDA method of feature selection.

	Precision	Recall	F-measure
0	0.667	0.500	0.571
1	0.800	0.889	0.842
2	0.750	0.840	0.792
3	0.631	0.705	0.792
4	0.857	0.571	0.666
5	0.800	0.800	0.685
6	0.583	0.933	0.800
7	0.857	0.8571	0.717
8	0.545	0.631	0.857
9	0.684	0.520	0.585
Accuracy			0.700
Macro avg	0.717	0.724	0.711
Weighted avg	0.712	0.700	0.695

Table 1: Performance Of Different Classifiers using Uncorrelated data

Confusion matrix

[2]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[3]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

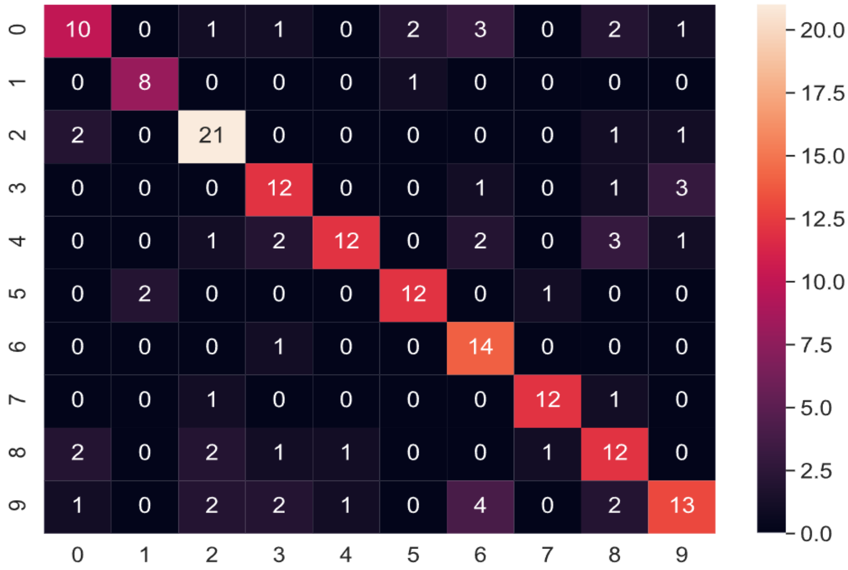

Figure 2: Confusion Matrix

4 Analysis of Results

The best method for feature selection is observed to be Linear Discriminant Analysis. Chi square did not perform that well as compared to LDA. For the classifiers, the best classifier was K-nearest neighbors. Random Forest classifier did equally well. The *eigen* and *svd* solver have almost equal scores. So using any of these would give almost same results.

5 Discussions and Conclusion

The performance of the classifiers could be increased by doing a more rigorous search of parameters. Supervised techniques for feature selection perform fairly well. Also, the use of Pearson correlation method of dropping highly correlated features may not be always helpful for feature reduction. It was observed that the performance was decreased when the features with high correlation were dropped. Few other feature selection techniques like Information gain, Principle Component Analysis can be used.