Music Genre Classification

Machine Learning in Biology

Elmer Carlos, Rita Ângelo

MSc. Computational Biology

May 14th, 2023

1 Introduction

Music Information Retrieval (MIR) is a field that deals with the development of algorithms and techniques for the automatic analysis, organization, and retrieval of music data. One important application of MIR is Musical Genre Classification, which involves automatically categorizing a music piece into one or more predefined genres based on its audio content. (Tzanetakis and Cook 2002)

Machine learning (ML) has played a crucial role in developing algorithms for MIR tasks, including genre classification. By analyzing features extracted from the audio signal, ML can learn to recognize patterns that are characteristic of different genres and make predictions on new, unseen music pieces. (Choi et al. 2017)

Musical genre classification has many practical applications, such as improving music recommendation systems, organizing large music collections, and aiding musicologists in studying and understanding different musical styles. However, it is a challenging problem due to the subjective nature of musical genres and the variability of musical pieces within a given genre. (Schedl et al. 2014)

The typical ML pipeline can be used to implement genre classification (music acquisition, audio signal pre-processing, feature extraction, feature reduction/selection, experimental analysis, and pattern recognition methods) which is the goal of this project assignment.

2 The dataset

The provided dataset (last updated in 2020) is a modification of the GTZAN dataset ("GTZAN (modified) - Music Genre Classification | Kaggle" n.d.), which is the most-popular publicly available dataset for testing in ML research for music genre recognition (MR). It contains 1000 audio tracks each 30 seconds long, which were collected in 2000-2001 from various sources to signify different recording conditions. Each track belongs to a total of 10 music genres, each represented by 100 tracks (all 22 050 Hz monophonic 16-bit audio files in .au format).

The dataset consists of 1000 lines (each music snippet), represented by 197 features (columns), where the 1st column is the filename and the 199th column is the class label that assigns each sample to one of the 10 genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock).

3 Our goal

Our goal with this project is to develop classifiers for genre discrimination.We were assigned two scenarios:

- **Scenario A** (binary classification): where we consider a one-vs-all classification.
- **Scenario B** (multiclass classification): where we consider the problem of classifying all genres together.

4 Practical assignment

In this portion we are going to work through several steps of designing and validating our classifiers:

- Data import;
- Feature Selection and Reduction;
- Experimental Analysis;
- Pattern Recognition Methods;
- Results and Discussion.

5 Import the modules

Before we even start importing the data we will be working with for the remainder of this project, we need to import the modules that are essential for its execution. The modules used and their purpose are described in the table below. (Harris et al. 2020; Hunter 2007; McKinney 2010; Pedregosa et al. 2011; Waskom 2021)

Module	**Purpose**
Pandas	Data manipulation
Numpy	Data manipulation
Matplotlib	Data visualization
Seaborn	Data visualization
Scikit-learn	ML models

6 Data import

In this step, the goal is to import the data we will be working with. Here we will create 2 different subsets according to the 2 scenarios described in the Our Goal section and then proceed to the feature selection and reduction step. For the remainder of the project, we will use 80% of the data as training/validation data and 20% as testing data.

First, we must read our data and perform a quick exploratory analysis. The raw dataset has 999 samples along with 199 features, in accordance with its description. After dropping the 1st column (**filename**), which does not provide any useful information to our model, our data is split into 1 categorical feature (**genre**) and 197 numerical features. These are, respectively, the dependent variable and the independent variables.

Since there are not any missing values present in the dataset, we continue by visualizing our data with a pair plot, to see variable distribution and correlation (**Fig. 1**).

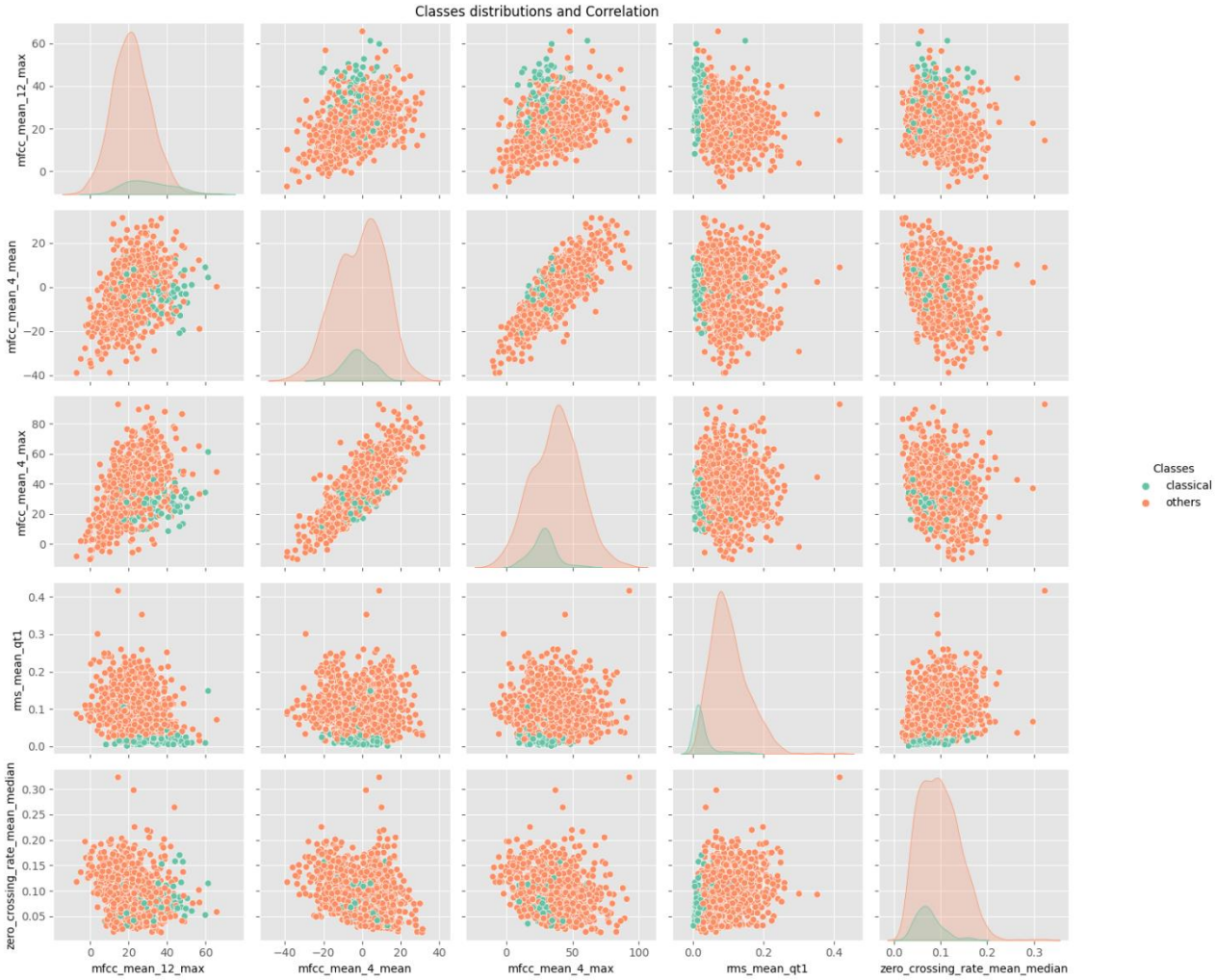


Figure 1 *- Distribution and correlation between the different classes for scenario A, where the variable of interest is the class "classical" (green).*

7 Scenario A (Binary Classification)

Binary classification is a supervised learning task in machine learning that divides new observations into one of two classes, based on a classification rule. For scenario A, we must pick a genre and treat it as our variable of interest (let us consider the genre "classical") and treat every other variable as the same.

Looking at class distribution (**Fig. 2**), we see that there is a significant class imbalance, meaning our model cannot effectively learn the decision boundary since there are not sufficient examples of the minority class. We used SMOTE (Chawla et al. 2011) to oversample the examples in the minority class.

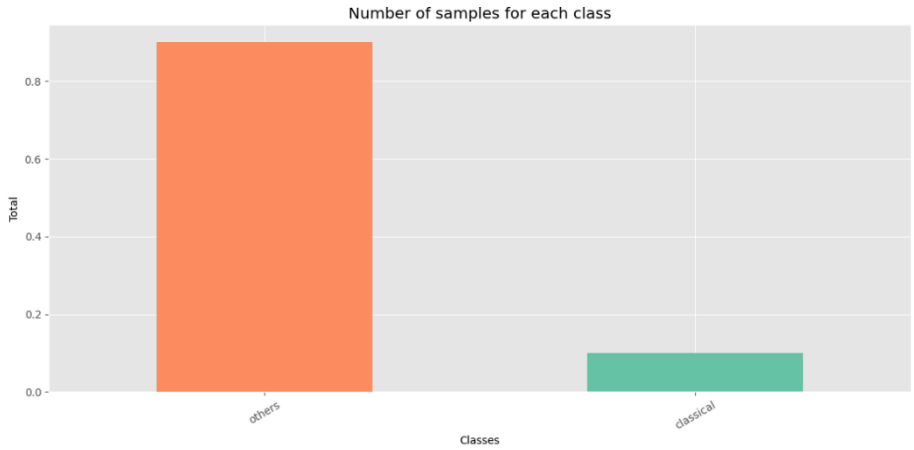


Figure 2 *- Class distribution for scenario A, where the variable of interest is the class "classical" (green).*

7.1 Feature selection/reduction

When creating a predictive model, the process of feature selection involves lowering the number of input variables. This process can enhance the performance of the model while also lowering the computational cost of modeling. Feature reduction, or dimensionality reduction, reduces the number of features without losing crucial information.

In this stage we split the data frame into our features and the target variables. We also extract the features names with the goal of being able the interpret the feature selection algorithms results.

7.1.1 Removing outliers

Models can be negatively affected by the presence of outliers by introducing a bias to the model's predictions. Removing them is important to prevent skewing in the results of data analyses. This is only done to the training data to prevent data leakage (when data from sources other than the training dataset is used to build the model) to the test data.

There are several methods to remove outliers from the dataset. We chose one already available from the scikit-learn module: the Local Outlier Factor (LOF).

The LOF algorithm calculates the local density deviation of a particular data point with respect to its neighbors. It is an unsupervised anomaly identification technique. The samples that have a significantly lower density than their neighbors are regarded as outliers.

7.1.2 Setting up a pipeline

We decided to define a pipeline with the following steps:

1. Scaled the data with a standard scaling method.
2. Checked for low variance features with a threshold of 1/4.
3. Selected the best 25 features according to the Mutual Information (MI) scores.
4. Reduced the features with PCA to the number of features that explain 95% of the variance of the data.

With this we can check which features remained from the pipeline steps (**Fig. 3**).

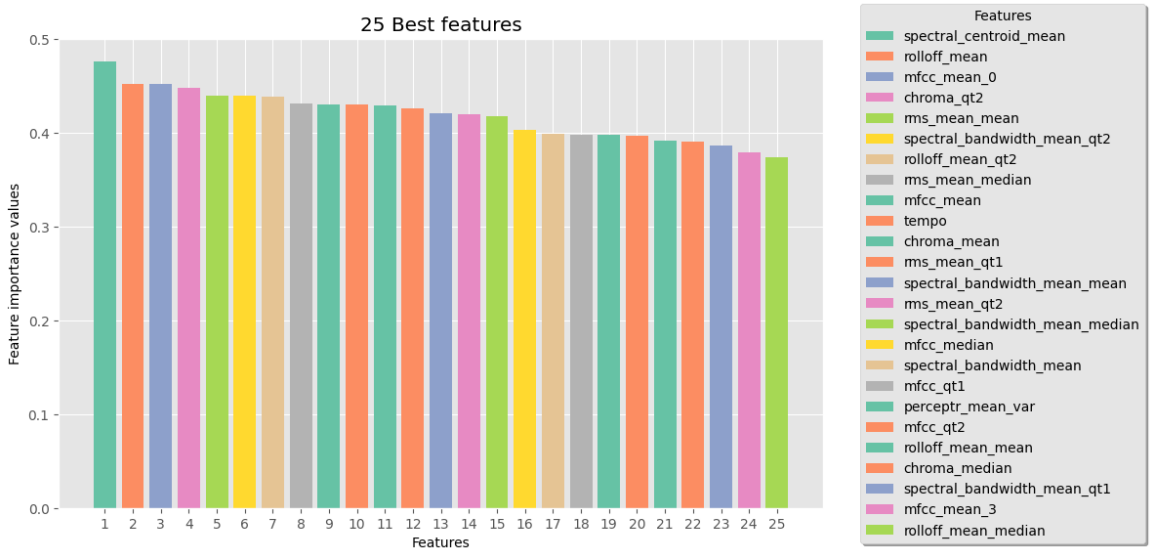


Figure 3 *– The 25 best features obtained for scenario A.*

7.1.3 Principal Component Analysis (PCA)

We use the principal component analysis (PCA) method to improve data interpretation whilst retaining the most information, and to analyze multidimensional data. We visualized the outcome of this method with a PCA graph (**Fig. 4**). In this case, there is a relatively high degree of separability between the two classes, so we wanted to check if the same would be true when considering different genres as our variable of interest. The resulting PCA graphs (**Appendix A**) showed that the classical music genre was the one that obtained a higher degree of separability.

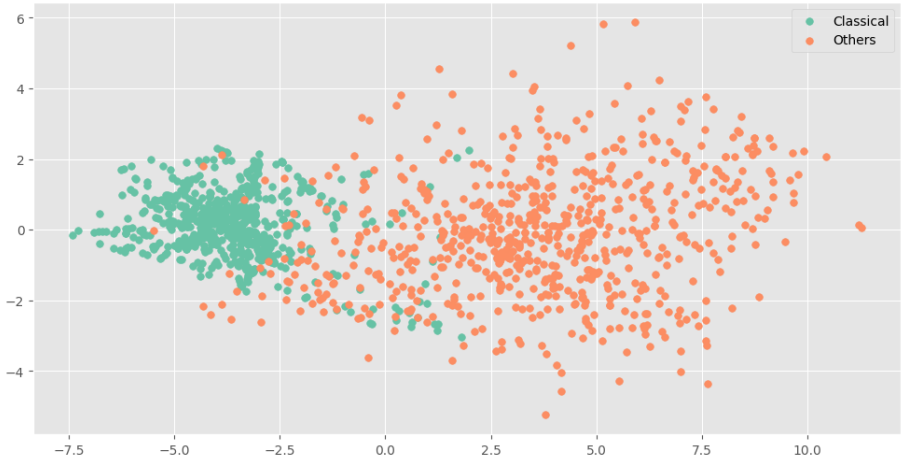


Figure 4 *– PCA plot for scenario A ("classical" vs "others").*

7.2 Classifiers

In this section we checked the performance of six models, according to each feature reduction method applied to the data. These six models are as follows:

- Logistic Regression;
- K-nearest neighbors (KNN);
- Fisher's LDA (Linear Discriminant Analysis);
- Random forest classifier;
- Support Vector Machine (SVM);
- Naïve Bayes (NB).

Grid-search was performed to check which hyperparameters were best for the logistic regression (C value between 0.01 and 100), KNN (for the number of neighbors we used an interval between 5 and 104, with intervals of 5), random forest (for the number of estimators an interval between 10 and 100, with intervals of 5, was considered), and SVM (an interval between 1 and 50, with intervals of 5 was considered for the C value) classifiers.

With these six models we performed cross-validation testing and independent testing using five different metrics:

- Accuracy;
- Precision;
- Recall;
- F1 score;
- Matthew's correlation coefficient.

8 Scenario B (Multiclass Classification)

Multiclass classification is a machine learning task that requires categorizing new observations. In scenario B, we must categorize the music excerpts into one of ten music genres. This challenge is more difficult than binary classification since it needs the model to distinguish between numerous classes at the same time.

Once again there are not any missing values in the dataset, so we can visualize our data with a pair plot, to see variable distribution and correlation (**Fig. 5**).

To guarantee that our model is not biased towards any one genre, we must evaluate class distribution (**Fig. 6**), as we did in scenario A. The GTZAN dataset is well-known for having an equal number of samples for each genre, making it an excellent choice for multiclass categorization. In this case, because there is not a significant class imbalance no further steps were used. However, data pretreatment and feature selection/reduction are still required to proceed to classification tasks.

Figure 5 *- Distribution and correlation between the different classes for scenario B.*

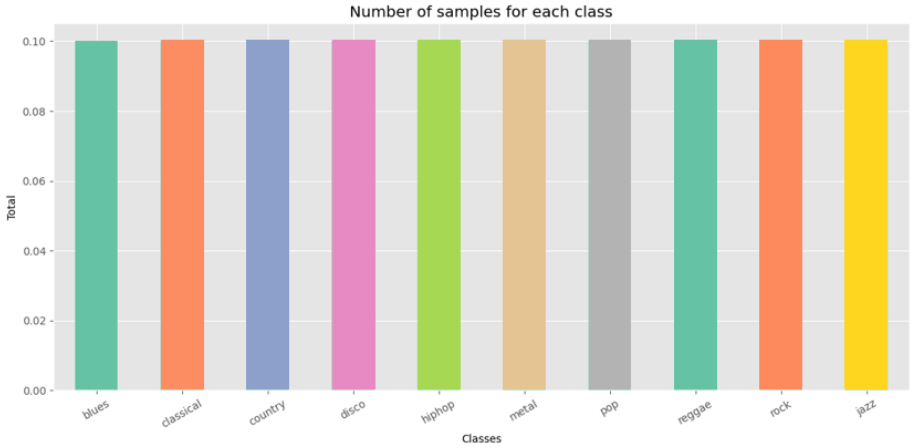


Figure 6 *- Class distribution for scenario A, where the variable of interest is the class "classical" (green).*

8.1 Feature selection/reduction

The process of feature selection/reduction for scenario B followed the same steps outlined for scenario A: outliers were removed using the scikit-learn module LOF, and the pipeline described in section 7.1.2 was applied.

8.2 Classifiers

As for scenario A, we want to evaluate our classification models' performance. We chose the same 6 classifiers, as well as the same 5 different metrics, except for the F1 score, which was swapped for the F1 weighted score.

9 Results & Discussion

9.1 Scenario A

9.1.1 Cross-validation testing

In our initial exploration of the results, we compared the calculated metrics with PCA or LDA as the feature reduction method and decided that using PCA as the feature reduction method yielded better results. In this final cross-validation test, we also ran a cross validation test for all the other genres and their results are shown in **Appendix B**.

From this figure (**Fig. 7**)., we can see that all our models achieved satisfactory results, with the Random Forest and the SVM models being the overall best classifiers, and while the Naïve Bayes is the worst classifier that we tested it still achieved a satisfactory performance across the board.

A possible reason for the lower performance of the Naïve Bayes model might be the distribution of the variables. An important assumption of this model is that the features of data we are using must be normally distributed and while the Standard Scaler might approximate the distribution of the features to a normal distribution it is possible that it is not close enough to satisfy this assumption.

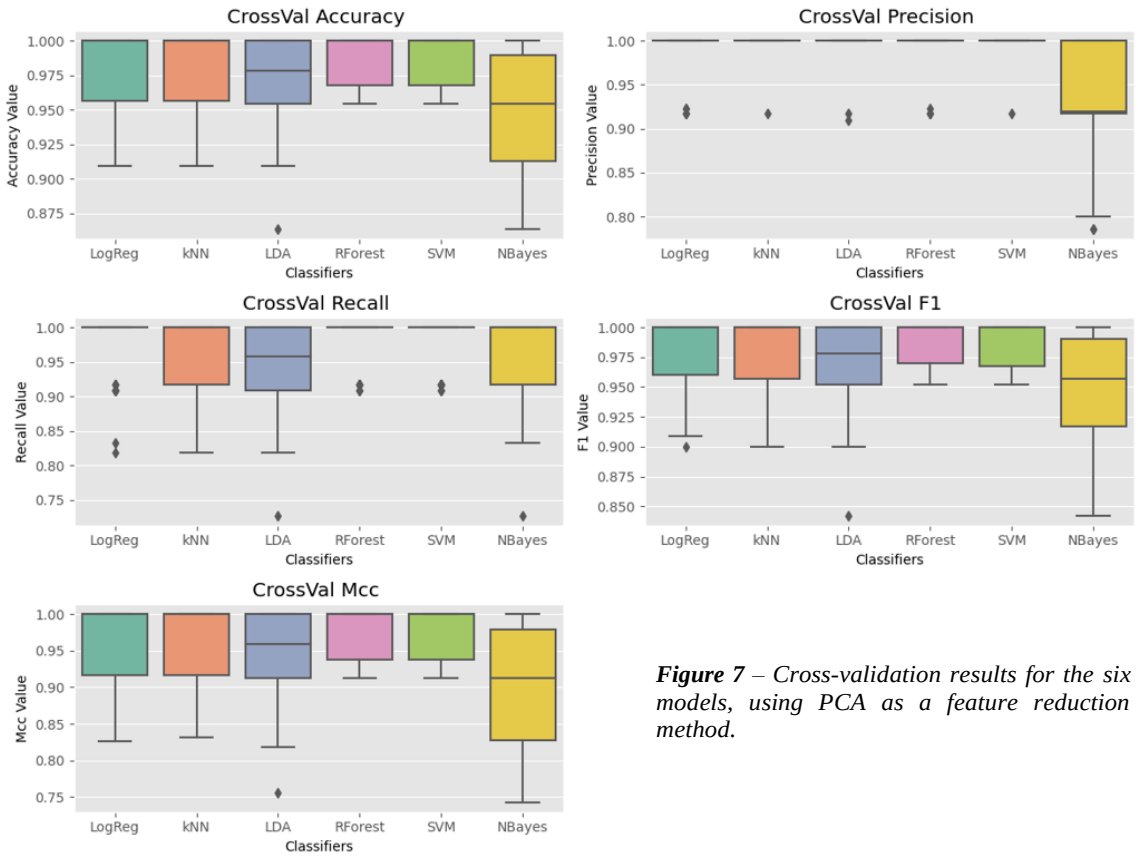


Figure 7 *– Cross-validation results for the six models, using PCA as a feature reduction method.*

9.1.2 Independent testing

Regarding the independent test (**Fig. 8** and **Fig. 9**), once again all the models showed a satisfactory performance. Here, there is no straightforward way to distinguish the overall best model from the group, because they yielded similar results for all the metrics that we used. The SVM model has the worst result for the Mathews Correlation Coefficient (**Table 1**), but this number is not considerably different from the other models' average performance.

Table 1 *- Independent test result for scenario A.*

	LogReg	KNN	LDA	RF	NB	SVM
Accuracy	0.975	0.970	0.950	0.955	0.970	0.940
Precision	0.983	0.983	0.994	0.972	0.988	0.966
Recall	0.989	0.983	0.949	0.977	0.977	0.966
F1	0.986	0.983	0.971	0.974	0.983	0.966
Mcc	0.884	0.863	0.81	0.791	0.868	0.726

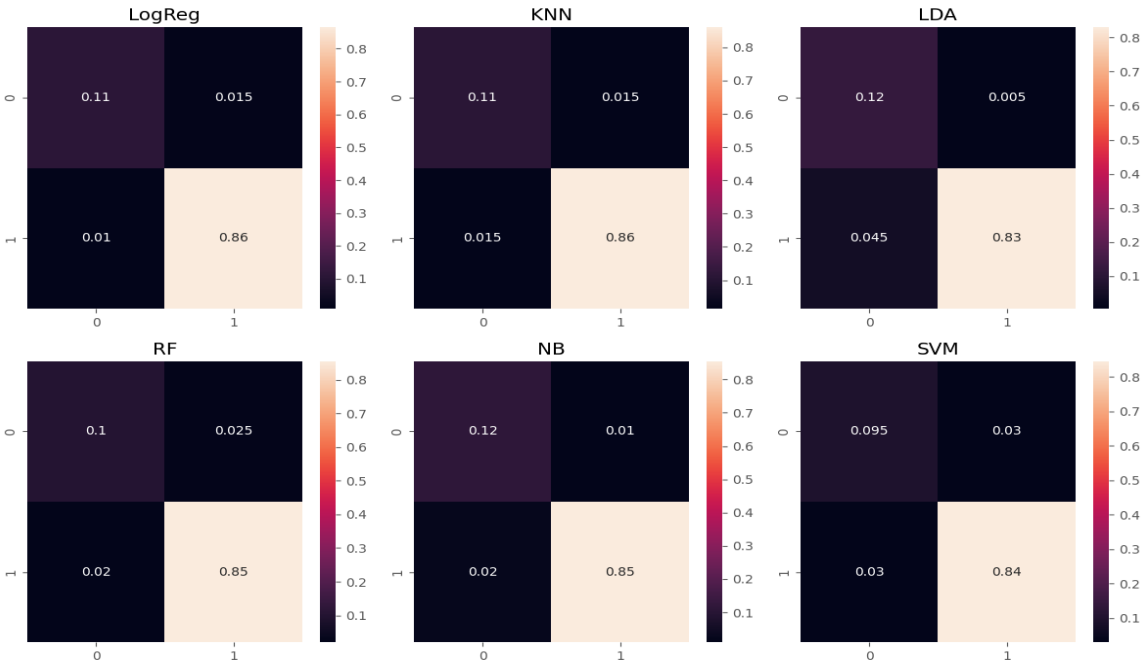


Figure 8 *– Confusion matrix results for the six models, using PCA as the feature reduction method.*

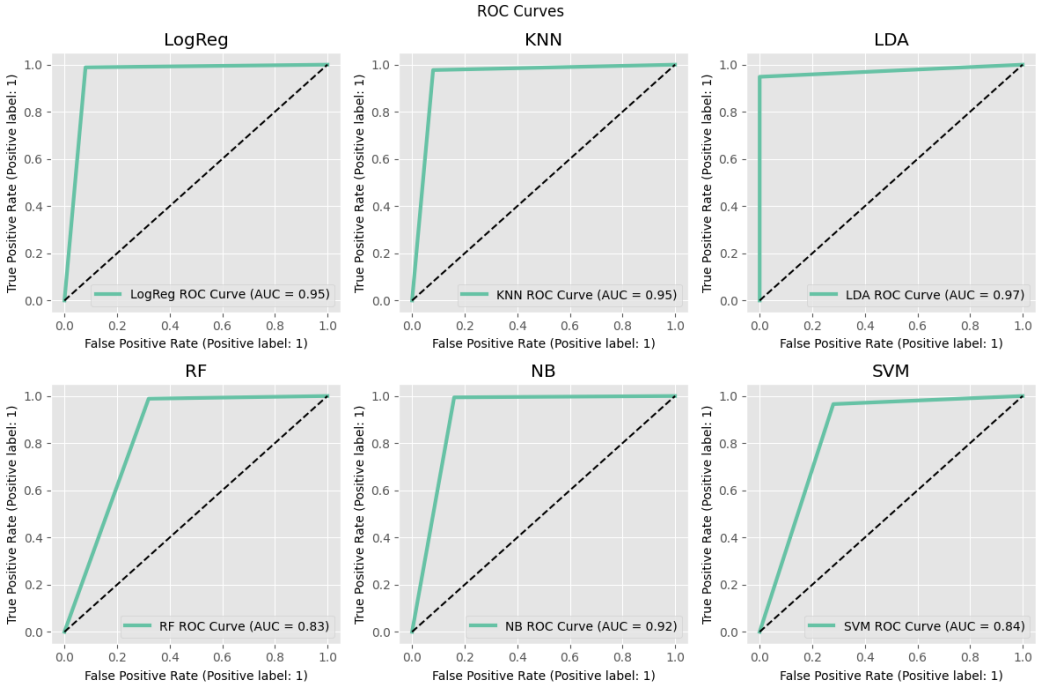


Figure 9 *- ROC curves obtained for scenario A.*

9.2 Scenario B

9.2.1 Cross-validation testing

Regarding the cross-validation test for scenario B (**Fig. 10**), we can see that there has been a significant decrease in the model's performance, with the Random Forest and k-Nearest Neighbors models achieving the overall best results, and contrary to what we saw in the scenario A, the Fisher's LDA was the worst performing model.

We believe that the reason the overall performance of the models decreased was because of the small number of samples for each music genre. After processing of the data each music genre had between 70 to 80 samples, which highlights the need of having a representative number of samples to be able to attain state-of-art results for multiclass classification of music genres.

While the decrease in performance is not desirable, our results are close to the ones obtained by Sturm. B. (Sturm, 2013) that explored the faults of the GTANZ dataset. In their study, they showcased the results of 5 classifiers, 3 implemented by the author (Minimum Distance, Minimum Mahalanobis distance and Nearest Neighbors) and two state-of-the-art Music Genre Recognition (MGR) models (SRCAM and MAPsCAT), and like the author postulated in their work, GTANZ might not be the best benchmark dataset for MGR models.

Another possible solution for this decrease of performance could also be the use of generative deep learning architectures, such as a Generative Adversarial Network (GAN) and its variation or a Variational AutoEncoder (VAE) to generate more samples for each genre based on the few samples that each already has.

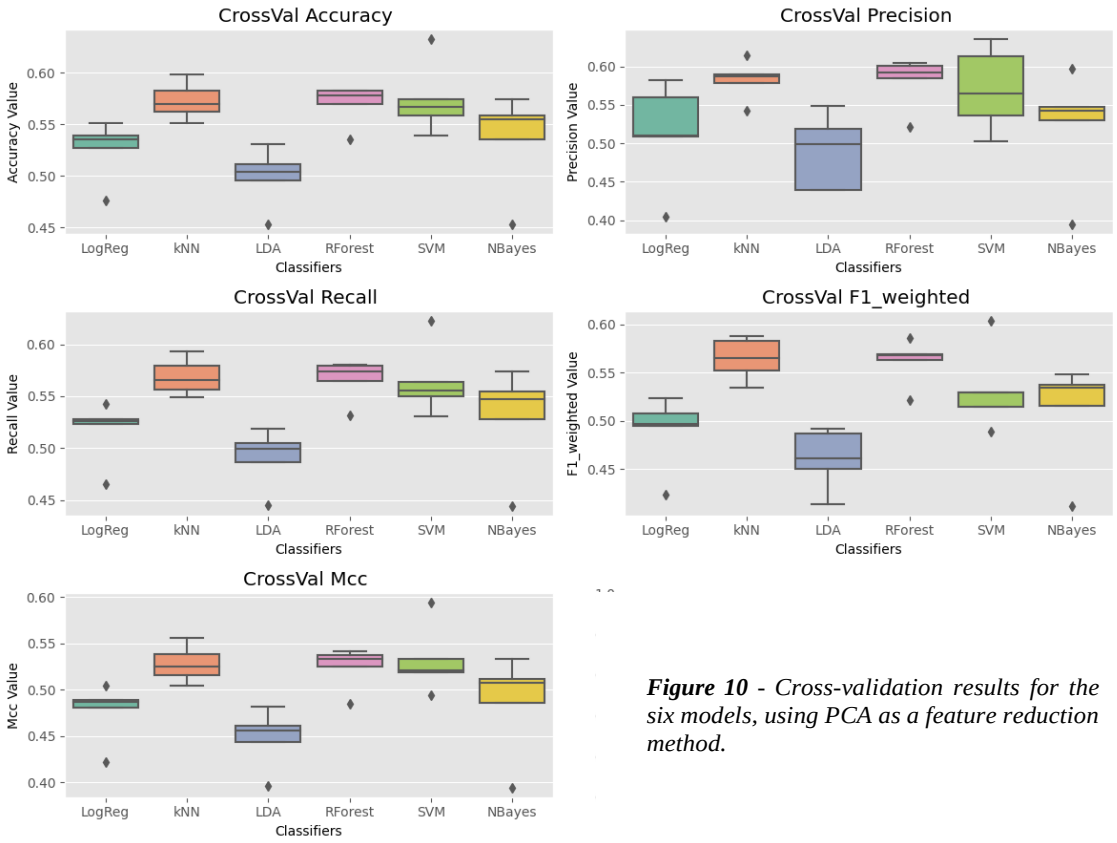


Figure 10 *- Cross-validation results for the six models, using PCA as a feature reduction method.*

9.2.2 Independent testing

Regarding the independent test for scenario B (**Figure 11**), we again see the same decrease in performance, when compared to scenario A. The same viable solutions we described in the cross-validation test section can be applied to this section, the need of a better benchmark dataset and the need of more sample for each genre would be beneficial to the performance of the models across the board.

Table 2 - Independent test for Scenario B.

	LogReg	KNN	LDA	RF	NB	SVM
Accuracy	0.445	0.495	0.43	0.53	0.54	0.445
Precision	0.436	0.518	0.421	0.536	0.569	0.424
Recall	0.445	0.495	0.430	0.530	0.540	0.445
F1 weighted	0.416	0.49	0.395	0.522	0.534	0.421
Mcc	0.391	0.444	0.375	0.48	0.494	0.387

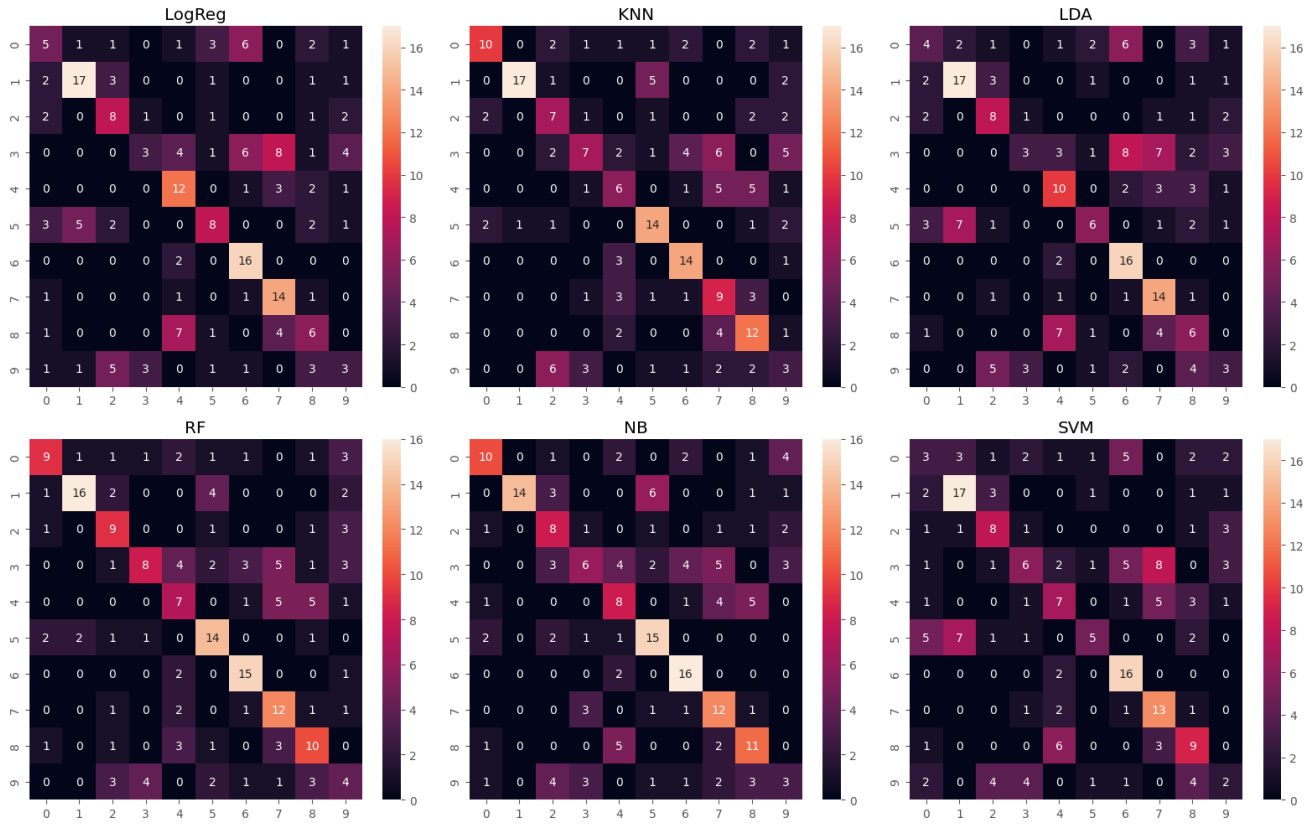


Figure 11 - *Confusion matrix results for the six models, using PCA as the feature reduction method.*

10 Conclusion

This report concludes by displaying our work on machine learning-based musical genre classification. In-depth analyses of binary and multiclass classification scenarios were conducted, and the effectiveness of six distinct models was assessed. The GTANZ dataset was the subject of our analysis.

The models had high accuracy, precision, recall, F1 score, and Matthew's correlation coefficient (MCC), according to our findings, which show that they performed remarkably well in the binary classification job. The results showed that the machine learning techniques used were effective at differentiating between the two classes defined (classical and other).

However, when we switched to the multiclass classification test, we saw a decline in performance as the models found it difficult to correctly categorize music samples from various genres. Despite this drop, it is crucial to remember that our findings were still comparable to other research in the area. This shows that the difficulty of multiclass classification in musical genre analysis is still a complicated issue that calls for additional research and perhaps sophisticated methodologies employing deep learning architectures.

11 References

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic Minority Over-sampling Technique. Journal Of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Choi, K., Fazekas, G., Cho, K., & Sandler, M. (2017). A Tutorial on Deep Learning for Music Information Retrieval. http://arxiv.org/abs/1709.04396

GTZAN (modified) - Music Genre Classification | Kaggle. (n.d.). https://www.kaggle.com/datasets/gabrielopecs/gtzan-modified-music-genre-classification. Accessed 18 March 2023

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. Nature 2020 585:7825, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science and Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 56–61. https://doi.org/10.25080/MAJORA-92BF1922-00A

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html. Accessed 18 March 2023

Schedl, M., Gómez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. Foundations and Trends in Information Retrieval. Now Publishers Inc. https://doi.org/10.1561/1500000042

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), 293–302. https://doi.org/10.1109/TSA.2002.800560

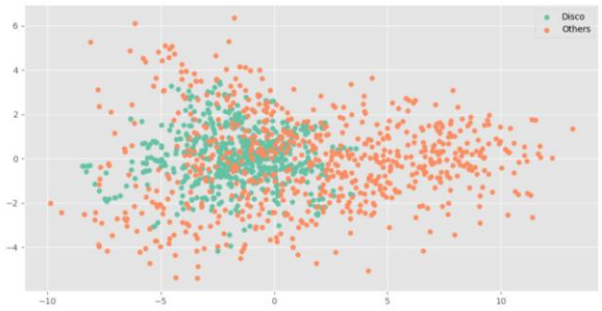
Waskom, M. L. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021. https://doi.org/10.21105/JOSS.03021

Sturm, B. L. (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. arXiv preprint arXiv:1306.1461.

12 Appendix

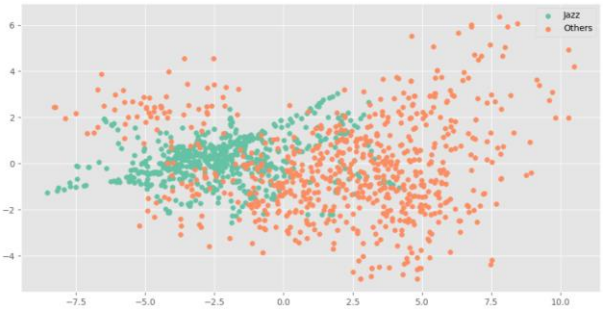
A - PCA plots for scenario A

Appendix 1 *- PCA plot for scenario A ("blues" vs "others").*



Appendix 2 *- PCA plot for scenario A ("disco" vs "others").*

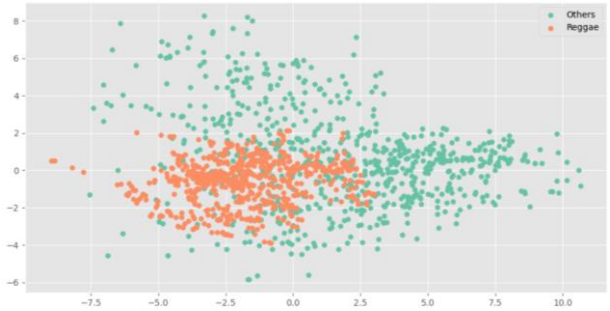
Appendix 3 *- PCA plot for scenario A ("hip-hop" vs "others").*



Appendix 4 *- PCA plot for scenario A ("jazz" vs "others").*

Appendix 2 *- PCA plot for scenario A ("metal" vs "others").*

Appendix 1 *- PCA plot for scenario A ("pop" vs "others").*

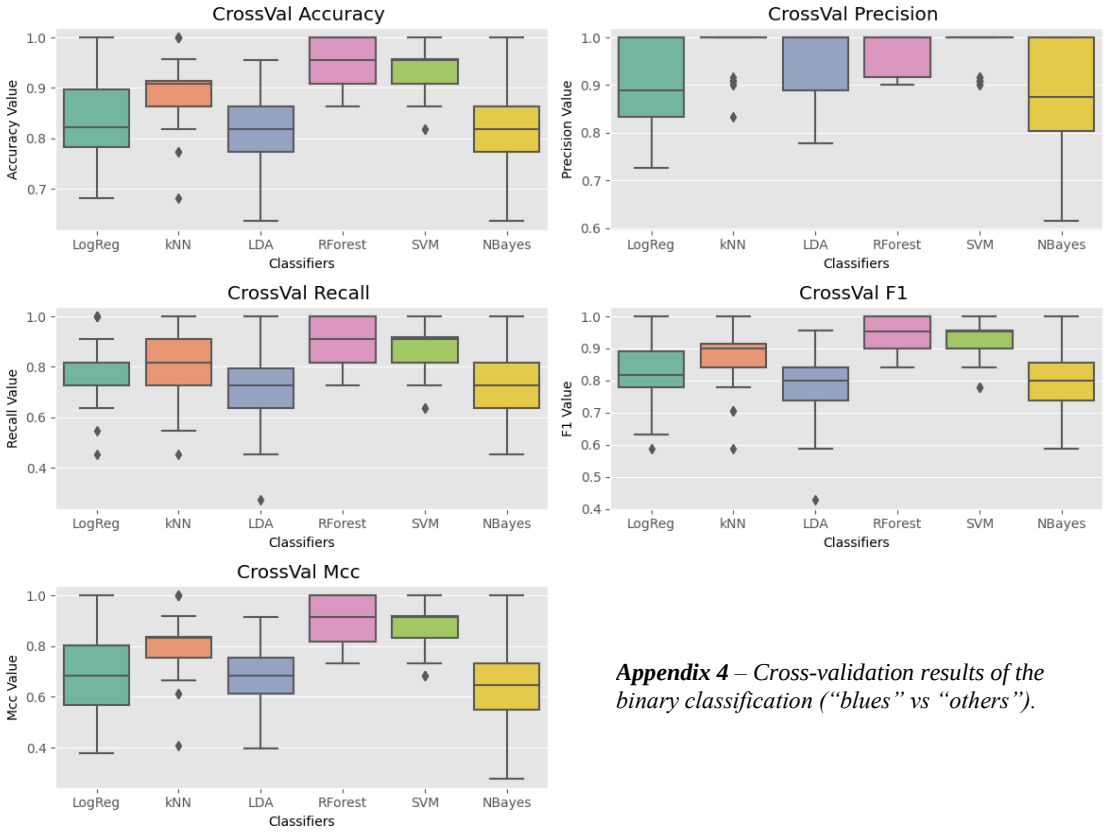


Appendix 7 *- PCA plot for scenario A ("reggae" vs "others").*

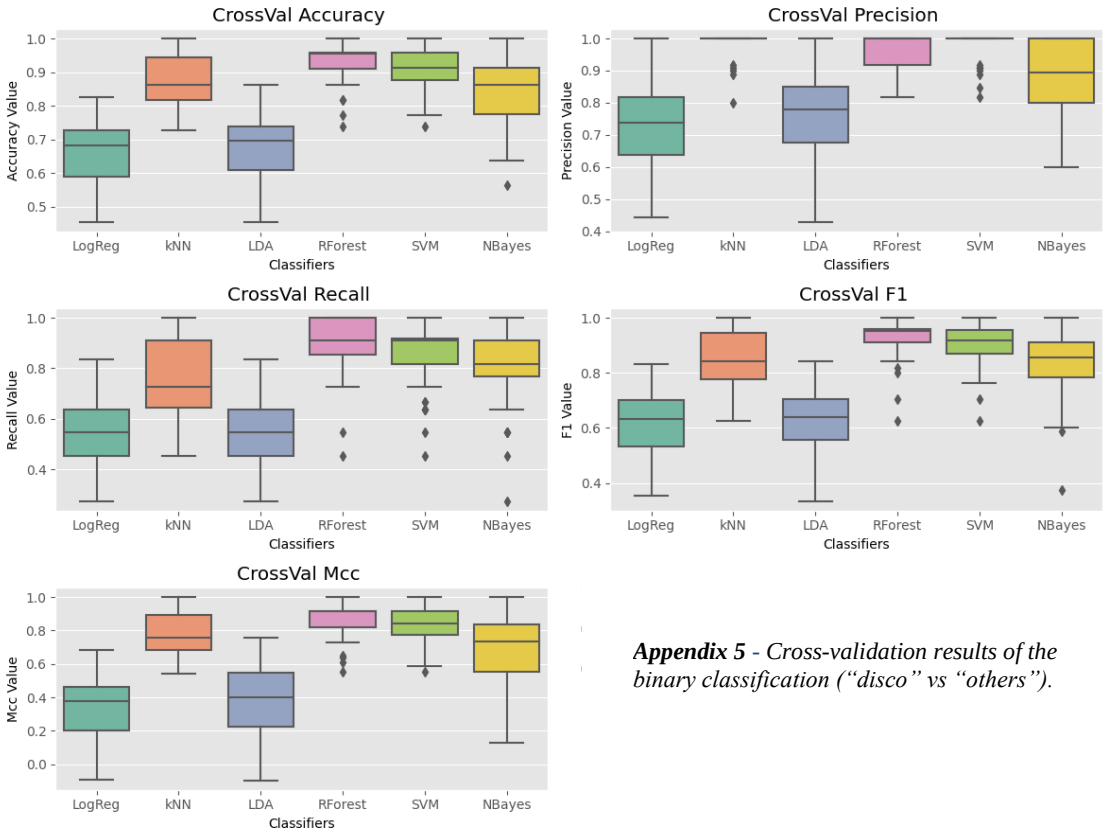
Appendix 8 *- PCA plot for scenario A ("rock" vs "others").*

Appendix 3 *- PCA plot for scenario A ("country" vs "others").*

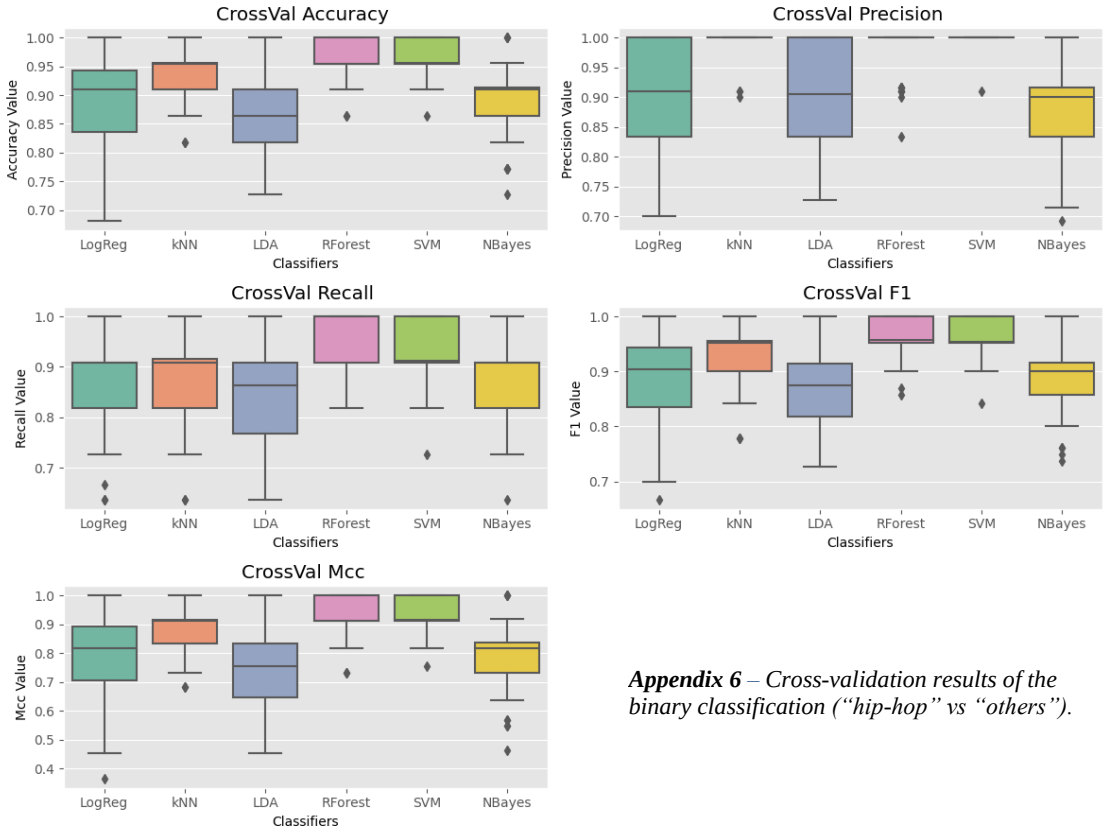
B - Cross-validation results for scenario A



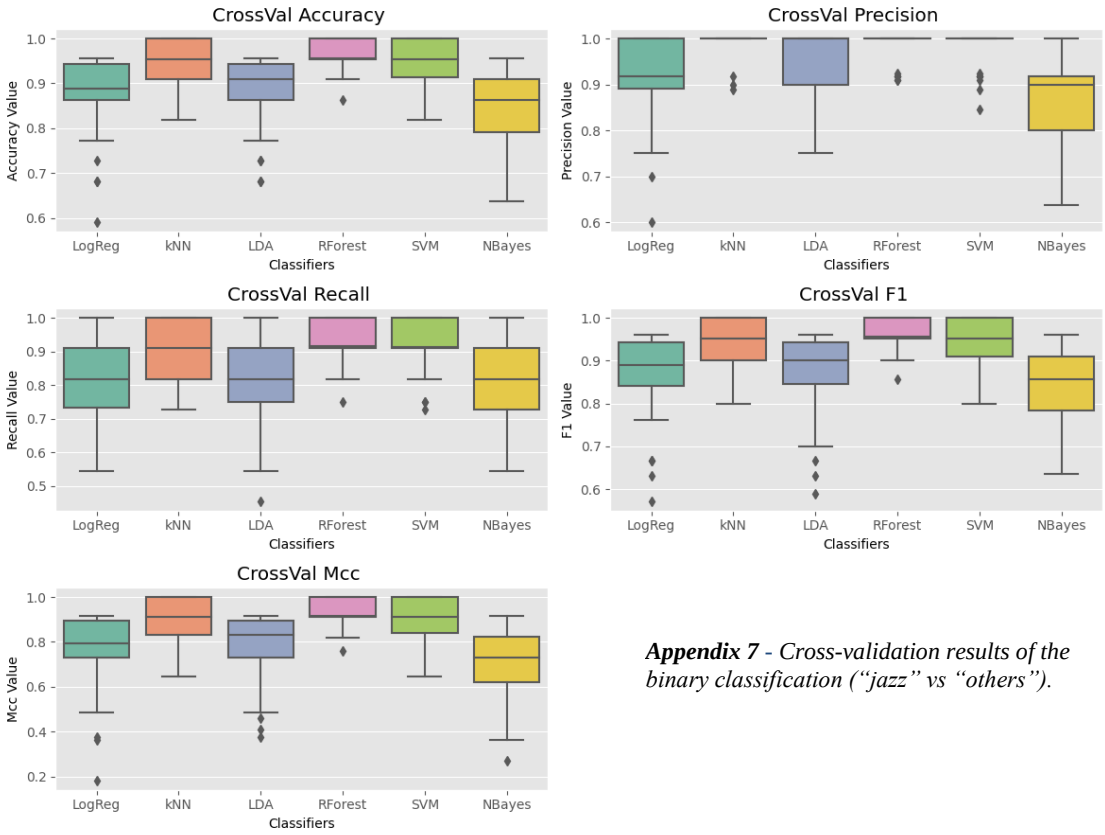
Appendix 4 *– Cross-validation results of the binary classification ("blues" vs "others").*



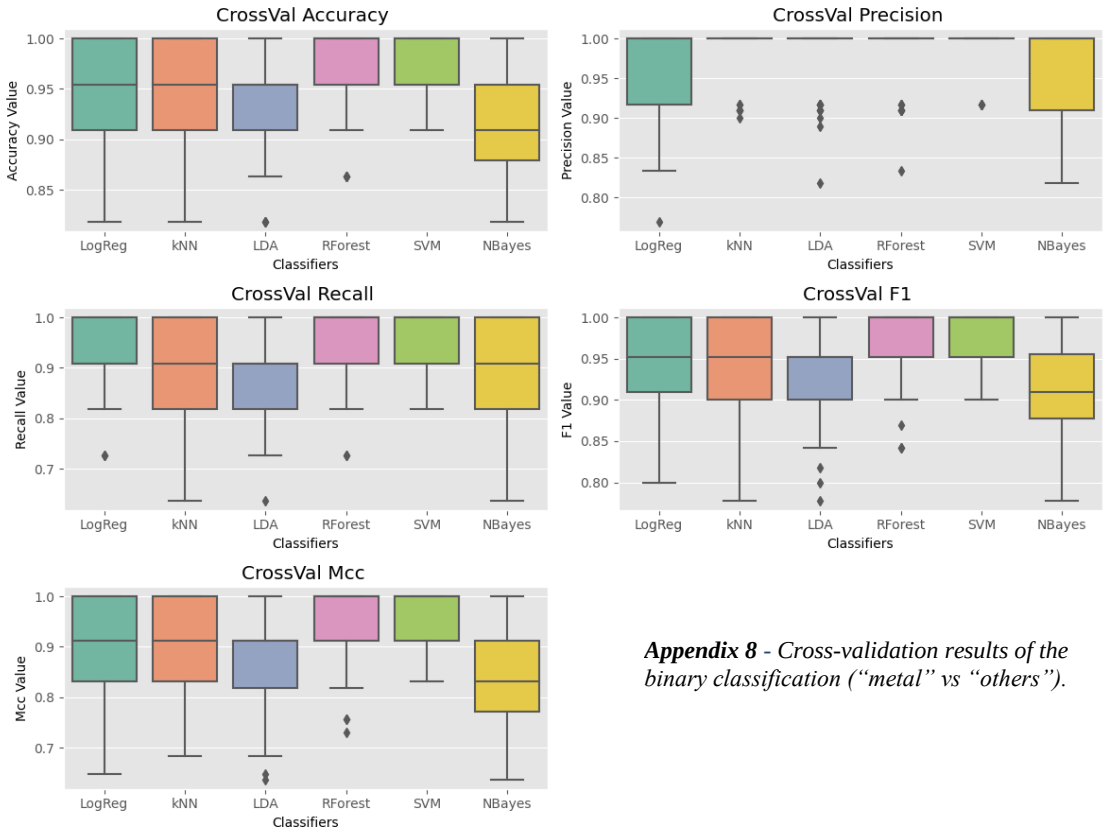
Appendix 5 *- Cross-validation results of the binary classification ("disco" vs "others").*



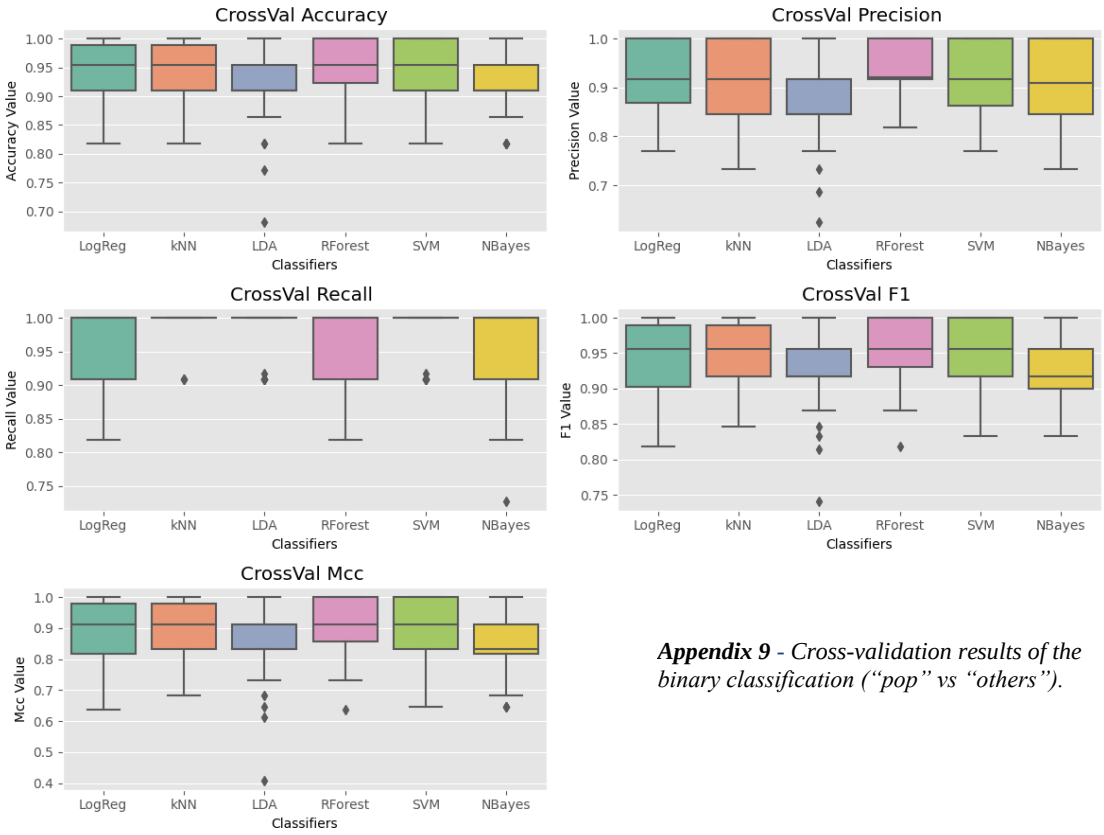
Appendix 6 *– Cross-validation results of the binary classification ("hip-hop" vs "others").*



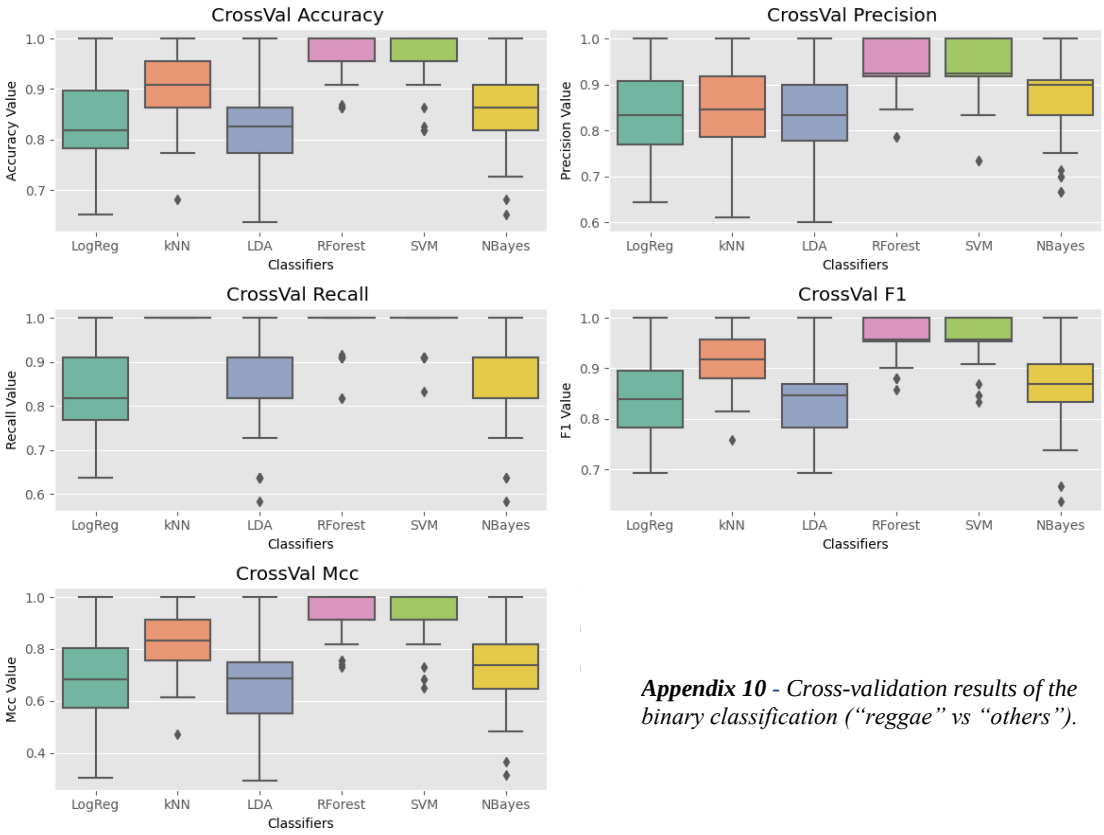
Appendix 7 *- Cross-validation results of the binary classification ("jazz" vs "others").*



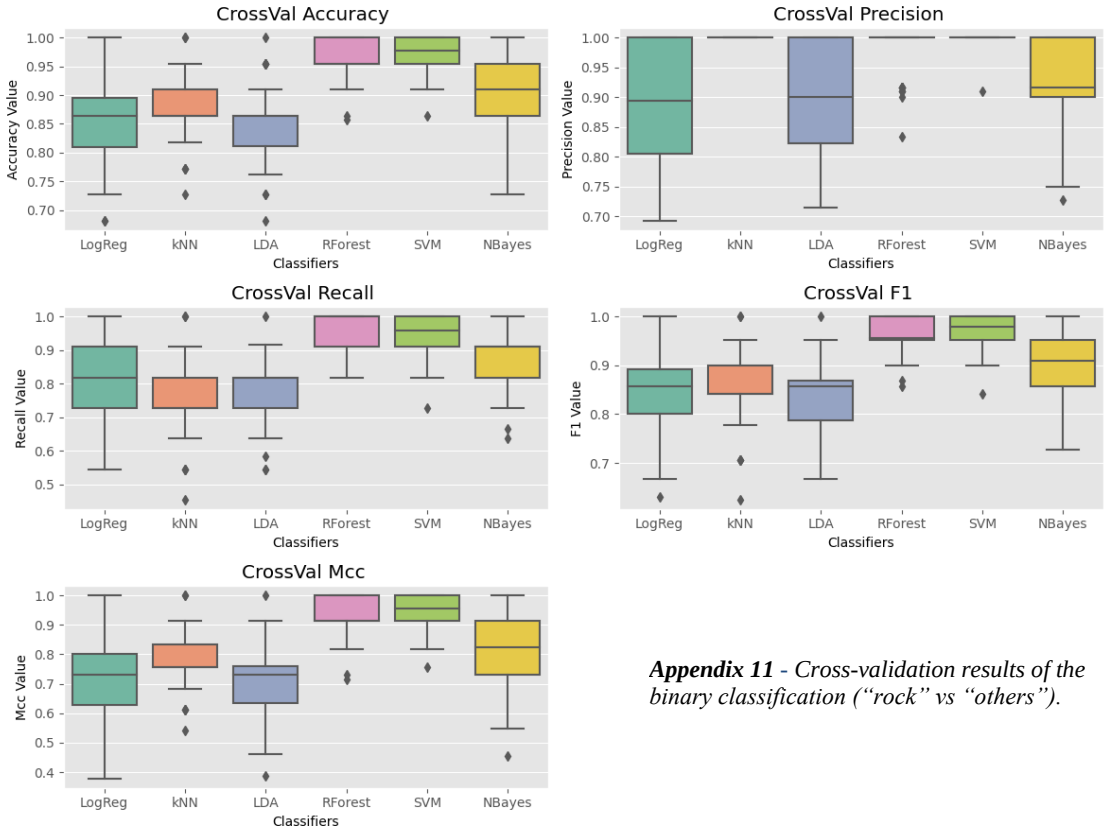
Appendix 8 *- Cross-validation results of the binary classification ("metal" vs "others").*



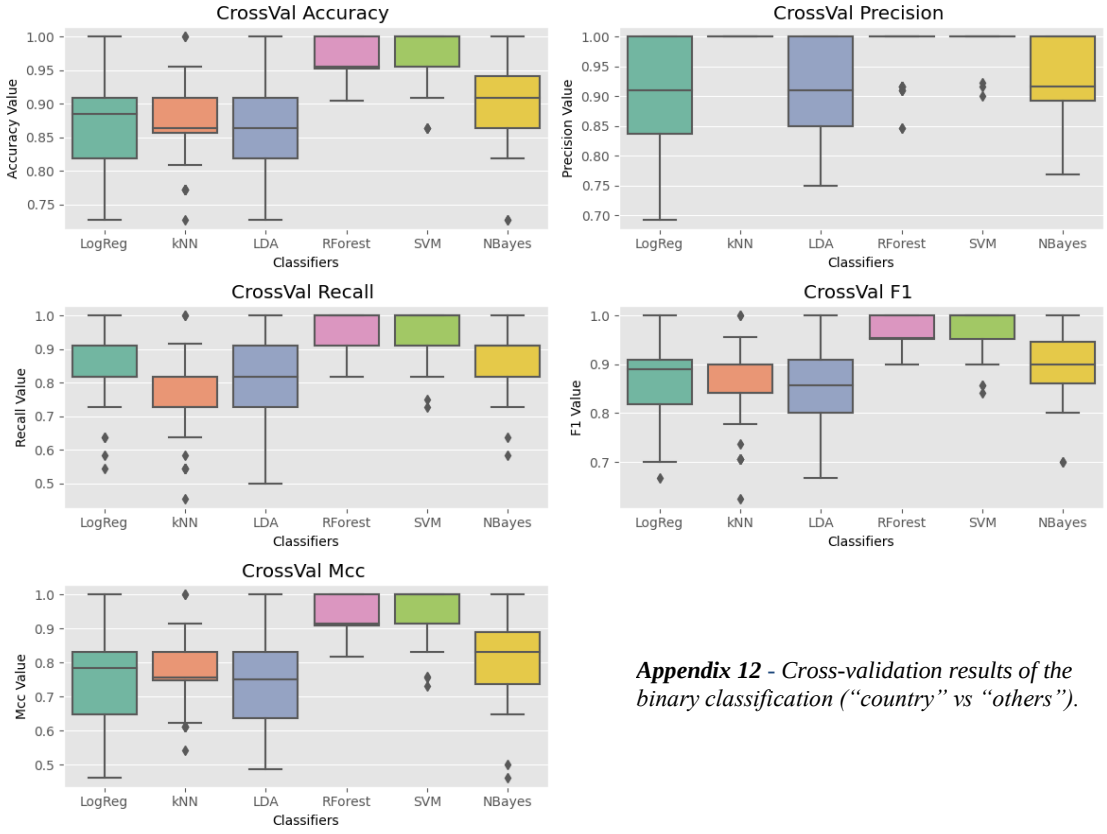
Appendix 9 *- Cross-validation results of the binary classification ("pop" vs "others").*



Appendix 10 *- Cross-validation results of the binary classification ("reggae" vs "others").*



Appendix 11 - *Cross-validation results of the binary classification ("rock" vs "others").*



Appendix 12 - *Cross-validation results of the binary classification ("country" vs "others").*