Comparing Multiple Classifiers for Music Genre Classification

COMP90049 Introduction to Machine Learning Semester 2 2020
University of Melbourne

1 Introduction

Music genre classification is a field that straddles both machine learning and human psychology, and begs the question: is it possible to build a perfect genre classifier, when genre selection may seem entirely subjective? When determining which features to prioritise, it is important to understand that the writer of the song not only drives meaning through the lyrics, but also through non-textual aspects of the song such as tempo, key, or mode (Neuman et al., 2016). While this might imply that the most beneficial approach to features is a general more-is-better, it is important to consider that improper selection of those key features may not only increase the computation time, but may even drag down performance by overpowering other features if they are a majority class (Kikuchi et al., 2020). As we attempt to examine the process of music genre classification, we will be building multiple classifiers using data provided from the Million Song dataset. The features included for analysis include both metadata features of the songs as well as audio vectors for comparison (Bertin-mahieux et al., 2011), (Schindler and Rauber, 2012). This comparison and classification task will be completed using three different types of multinomial classifiers and appropriate data preprocessing functions. Over the course of this paper, analysis will show that preprocessing the data is a necessary component of the classification process, and inadequate preprocessing of the data will lead to inaccuracies that cannot be fixed by modifying the parameters of each classifier when building the learners, especially when the main flaw with the data provided is imbalanced class outcomes.

2 Building the Classifiers

The goal of each classifier is to examine a song from a set of features given, and place it into one of eight genres: soul and reggae, pop, punk, jazz and blues, dance and electronica, folk, classic pop and rock, and metal. The data provided was already split into a training set, validation set, and testing set, so there was no need to implement a sampling method to select a validation set when training and optimising.

2.1 Preprocessing the Data

At first glance, it is apparent that while there exist representatives of each genre in the data set, there are genres that appear much less frequently than others, so we will have to ensure that those genres are not consumed by the more often occurring numbers. This is a case of multiclass imbalance, which can be more difficult to deal with than a classic binary imbalance, as there are multiple majority classes and multiple minority classes (Fernandez et al., 2018). While we might not have the classic problem of a "more interesting" class being underrepresented here, as often is the case for the minority class, we still have the problem that the classifiers will assume a greater significance on the data that is more greatly represented (Krawczyk, 2016).

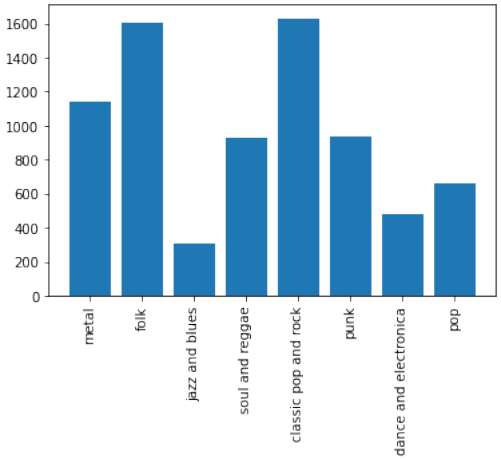

Figure 2.1: Training Labels before Preprocessing

The simplest approach to remedying the underrepresentation of minority classes is to re-

peatedly oversample them in the preprocessing phase – that is, put a copy of the data in to bulk up representation. The data is then undersampled for the majority class, to bring the numbers to be more equal in a rough approximation of the SMOTE method (Bowyer et al., 2011). This method advocates for both undersampling majority classes as well as oversampling of minority classes in order to smooth out the imbalance while minimising data loss. This was done simply by appending the data to the end of the training data and removing a small percentage of the majority classes, creating a more balanced training field, as is shown in the post-processing analysis seen in Figure 2.2.

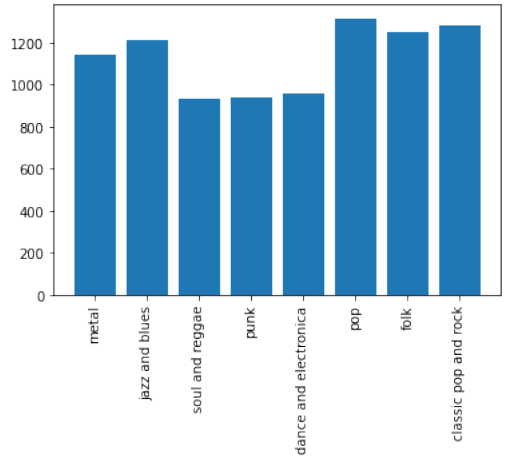

Figure 2.2: Training Labels after Preprocessing

Genre	Count Before	Count After
metal	1143	1143
folk	1601	1251
jazz and blues	303	1212
soul and reggae	930	930
classic pop and rock	1629	1279
punk	937	937
dance and electronica	478	956
pop	657	1314

Table 1: Training Data Distribution Before and After Preprocessing

When examining the training data, it is important to note that there is a mix of data types in the features as well – text classes (title of the song, lyrical tags), integer classes (time signature, key, mode), and also continuous data (loudness, tempo, duration, and song vectors). In order to ensure the classifier was able to interpret all of the data, the textual categorical features were converted to one-hot encoding. When using a classifier like Logistic Regression or K-Nearest-Neighbour, the data was also scaled to ensure comparison was possible between the different feature values (Rençberoğlu, 2019).

2.2 Baseline: Zero-R

All of these models will be compared to the Zero-R baseline, that is, the majority class. In this case, the majority class is the classic pop and rock genre, ultimately giving an accuracy of 12.22% when evaluated over the validation data.

2.3 K-Nearest-Neighbour

For K-Nearest-Neighbour, the two most important algorithmic modifications to consider are how many neighbours to examine, and how much to weight each neighbour depending on distance (Altman, 1992). In order to fine tune the algorithm used, the classifier was built using a varying number of neighbour points (between 1 and 200 for the initial test), and the two possible weighting methods would be to either weight all of the points equally, or weight them by distance. The relative weighting of the data barely effected the accuracy – uniform and distance weighting varied by less than a percent as can be seen in Figure 2.3. Distance weighting had the slight edge with an accuracy of 52% with K=32.

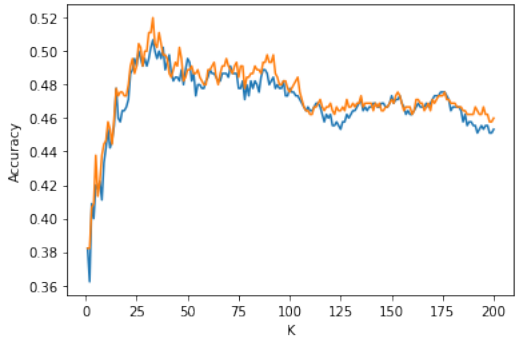

Figure 2.3: Determining Optimal Number of Neighbours

2.4 Random Forest

With the Random Forest algorithm, an ensemble classifier built up of many Decision Trees, the algorithm may be tuned to an optimum number of estimators (Mitchell, 1997)(Tan et al., 2019). The tuning function tested between 10 and 300 going up by intervals of 10. Figure

2.4 shows the change in the accuracy of the tuning function, with the highest accuracy topping out at 54.45% with 170 trees.

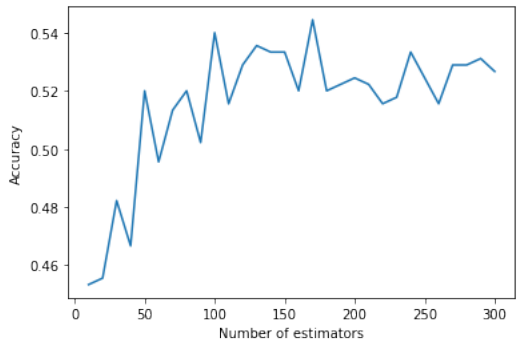

Figure 2.4: Determining Optimal Number of Decision Trees in Forest

2.5 Logistic Regression

Logistic Regression, one of the more general algorithms, was optimised both with the number of iterations, and the base algorithm used for the regression itself (Tan et al., 2019). For a multiclass logistic regression, we will approach it in a very similar manner as a binary logistic regression, but where we will be calculating the likelihood of each class, instead of focusing on a single positive class (Bishop, 2007). When determining the optimum number of iterations for this logistic regression function, it was tested on the lowest number of iterations it would complete and still converge, up until it displayed a similar behaviour to the previous two classifiers and the slope of the accuracy curve flattened to zero. The maximum accuracy reached was 52.45% at 2200 iterations using stochastic gradient descent as the main regression method.

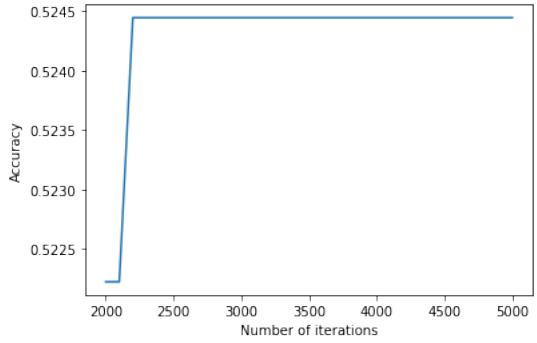

Figure 2.5: Determining Optimal Number of Iterations

3 Behaviour of the Classifiers

Looking at Figures 2.3-2.5 in Section 3, it is apparent that while the arguments used to build the classifiers do effect accuracy, once the optimum argument is found, the accuracy does not vary greatly. In comparison, preprocessing the data in order to make it more palatable for the learner is a greater component in building an accurate classifier. On average, simply preprocessing the data with repeated subsampling improved the maximum recorded accuracy by over 2%.

Classifier	Accuracy with Preprocessing	Accuracy without Preprocessing
Zero-R	12.22	12.22
K-Nearest-Neighbour	52	47.78
Random Forest	54.44	49.78
Logistic Regression	52.45	50

Table 2: Accuracy of Classifiers

Tables 3.1 and 3.2 show the change in accuracy when preprocessing data, while tables 3.3-3.5 show that changing features of the classifiers themselves does not have a similar impact. Once each classifier gets above a threshold value for their training arguments, the curve of the change in accuracy flattens, implying that any change in accuracy is due to fluctuation in the learning itself, not any significant change due to the argument when building the classifier. Each of these models is being tested on the preprocessed data set.

K	Accuracy (percent)
1	38.22
10	44.22
50	49.56
100	47.78
200	45.33

Table 3: Accuracy of K-Nearest-Neighbour varying K Values

3.1 Error Analysis

When analysing the results of the three classifiers built, it is important to note that while accuracy and F1-Score are important, it is also important to measure the bias and variance of the classifiers in order to judge their suitability over varied testing data sets [12].

Comparing the outcome of data with preprocessing and without for the each of the classi-

Number of Estimators	Accuracy (percent)
10	45.33
50	52
100	54
200	52.44
300	52.66

Table 4: Accuracy of Random Forest varying Number of Estimators

Number of Estimators	Accuracy (percent)
2000	52.22
2500	52.45
3000	52.45
4000	52.45
5000	52.45

Table 5: Accuracy of Logistic Regression varying Number of Iterations

fiers, the true impact of preprocessing is clearly shown. When comparing the data between Tables 6 and 7, many of the metrics remain similar, only shifting by a few percentage points. The overall accuracy improves when the data is balanced before processing, but the error rate remains low.

However, one of the more interesting conclusions can be drawn from examining the respective confusion matrices. While the overall accuracy of the classifiers does improve, if only by a couple of percentage points, the more often occurring predictors that the data was imbalanced towards are more often predicted, and more often correctly, but the minority class genres are sometimes completely ignored and entirely misclassified. This can be seen when looking at the "soul and reggae" genre in Figures 3.1 and 3.2 - when the data is preprocessed, the majority of instances are correctly assigned to soul and reggae, whereas when it is not only one instance is correctly assigned and the rest are misclassified.

Another interesting takeaway from examining the confusion matrices of these classifiers is that certain genres have higher percentages of misclassification with other genres, for example metal and dance and electronica, or punk and pop. This could create an opportunity for other interesting courses of study by examining which features of those specific genres greatly overlap (i.e. what they have in common) and why they are especially confusing for these classifiers, but that is outside the scope of this paper.

3.1.1 K-Nearest-Neighbour

Metric	Measurement With	Measurement Without
Accuracy	51	46
Precision	56	66
Recall	51	44
Bias	12.89	12.44
Variance	0.02	0.04
Mean Squared Error	12.5	13.7

Table 6: Metrics for K-Nearest-Neighbour (averaged)

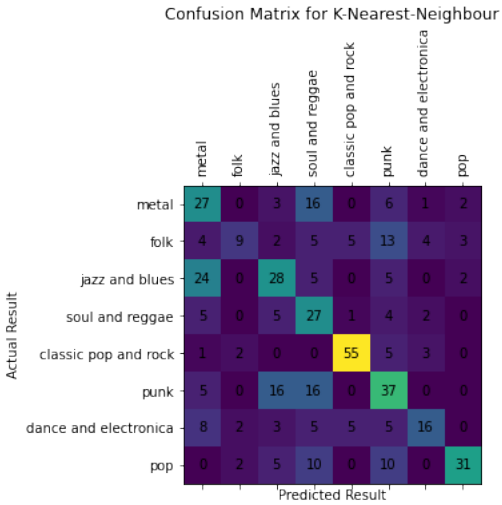

Figure 3.1: Confusion Matrix for K-Nearest-Neighbour with Preprocessed Data

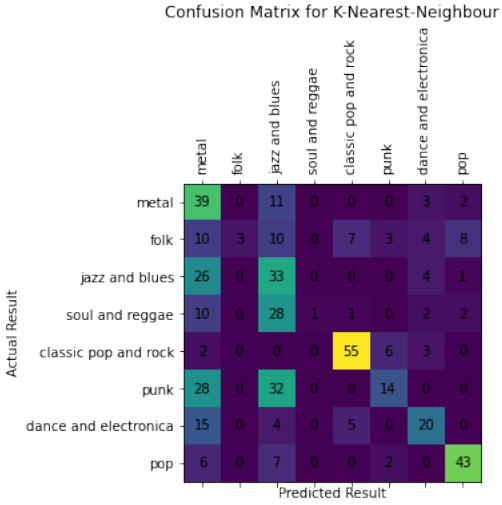

Figure 3.2: Confusion Matrix for K-Nearest-Neighbour without Preprocessed Data

Metric	Measurement With	Measurement Without
Accuracy	51	46
Precision	65	66
Recall	51	44
Bias	13.56	13.78
Variance	0.006	0.01
Mean Squared Error	13.51	13.63

Table 7: Metrics for Random Forest (averaged)

Metric	Measurement With	Measurement Without
Accuracy	52	50
Precision	57	58
Recall	51	50
Bias	16	18.89
Variance	0.05	0.06
Mean Squared Error	17.95	19.14

Table 8: Metrics for Logistic Regression (averaged)

3.1.2 Random Forest

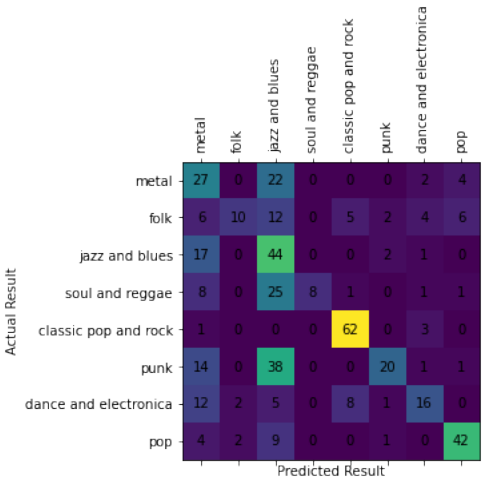

Figure 3.3: Confusion Matrix for Random Forest with Preprocessed Data

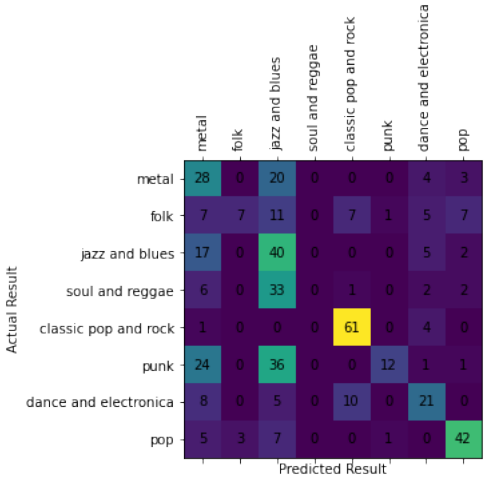

Figure 3.4: Confusion Matrix for Random Forest without Preprocessed Data

3.1.3 Logistic Regression

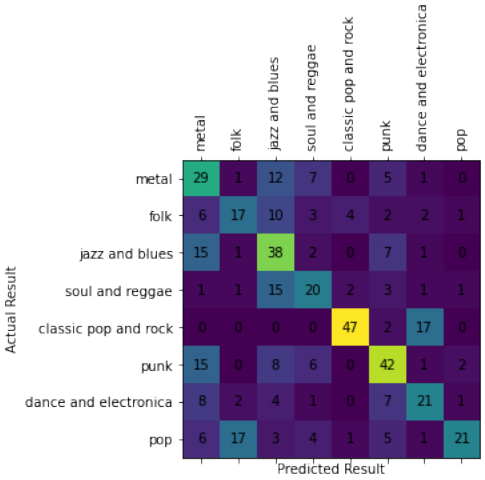

Figure 3.5: Confusion Matrix for Logistic Regression with Preprocessed Data

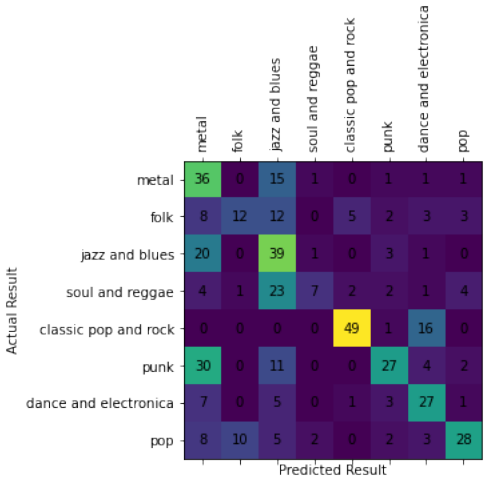

Figure 3.6: Confusion Matrix for Logistic Regression without Preprocessed Data

4 Conclusion

This project emphasised the role of preprocessing data before attempting to build an accurate classifier using that data to train. Even the simplest feature engineering like balancing the data set before classifier training will improve not only the accuracy, but also increases the recall and decreases the overall error. While it is still possible to achieve better results than a strict baseline by using unprocessed data, algorithm engineering alone when building a classifier is not enough to remedy a poorly constructed test data set, and will be combined with feature engineering in order to create a robust multiclass classifier.

References

N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–181.

Thierry Bertin-mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. In *In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR.*

Christopher M. Bishop. 2007. *Pattern Recognition and Machine Learning.* Springer, New York.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority oversampling technique. *CoRR*, abs/1106.1813.

Alberto Fernandez, Salvador Garcia, Mikel Galar, Ronaldo C. Prati, Bartocz Krawczyk, and Francisco Herrera. 2018. *Learning From Imbalanced Data Sets.* Springer.

Y. Kikuchi, N. Aoki, and Y. Dobashi. 2020. A study on automatic music genre classification based on the summarization of music data. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 705–708.

Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

Tom Michael Mitchell. 1997. *Machine learning.* McGraw-Hill.

Yair Neuman, Leonid Perlovsky, Yochai Cohen, and Danny Livshits. 2016. The personality of music genres. *Psychology of Music*, 44(5):1044–1057.

Emre Rençberoğlu. 2019. Fundamental techniques of feature engineering for machine learning, Apr.

Ar Schindler and Andreas Rauber. 2012. Capturing the temporal domain in echonest features for improved classification effectiveness. In *In Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR.*

Pang-Ning Tan, Michael Steinbach, Vipin Kumar, and Anuj Karpatne. 2019. *Introduction to Data Mining, Global Edition.* Pearson Education Limited.