COMP90049 Assignment Two
Music Genre Classification Report

Anonymous

1 Introduction

To listen to every track by every artist to find new music is an insurmountable task. Therefore, we have musical genres that group music based on a “shared tradition or set of conventions” [1]. We can hence filter music on our preferred genre(s) saving time and effort.

Music Genre Classification (MGC) is the task of predicting the genre of a song. Our dataset has eight genres: Soul and Reggae, Pop, Punk, Jazz and Blues, Dance and Electronica, Folk, Classic Pop and Rock, and Metal. In this paper, we find a set of features derived from the full set and use them for genre classification on several models. We aim to show the following: a) a stacking ensemble is valuable for genre classification, and b) it outperforms all its base estimators.

The paper is structured as follows. We give an overview of related work in MGC in Section 2. In Section 3, we explore the dataset. In Section 4, we detail our approach in selecting features and classification models. We evaluate our models in Section 5. Finally, we draw conclusions and preface future work in Section 6.

2 Related Work

MGC is a well-researched area with no shortage of impressive related work. One such example is [2], in which the authors propose a similar audio feature set including timbral texture. They compare different audio features and conclude that timbral texture is most expressive in MGC. They compare statistical pattern recognition (SPR) classifiers with human classifiers and demonstrate that SPR estimators are comparable to human genre classification in performance.

3 Dataset

The dataset is provided by Bertin-Mahieux et al. [3] and A. Schindler and A. Rauber [4]. It consists of 156 features for 8556 tracks, split into training, validation, and test sets, of 7678, 450, and 428 songs, respectively.

Figure 1 reveals the imbalanced class distribution in training and validation sets, which makes for difficult learning of the underrepresented classes. [5]

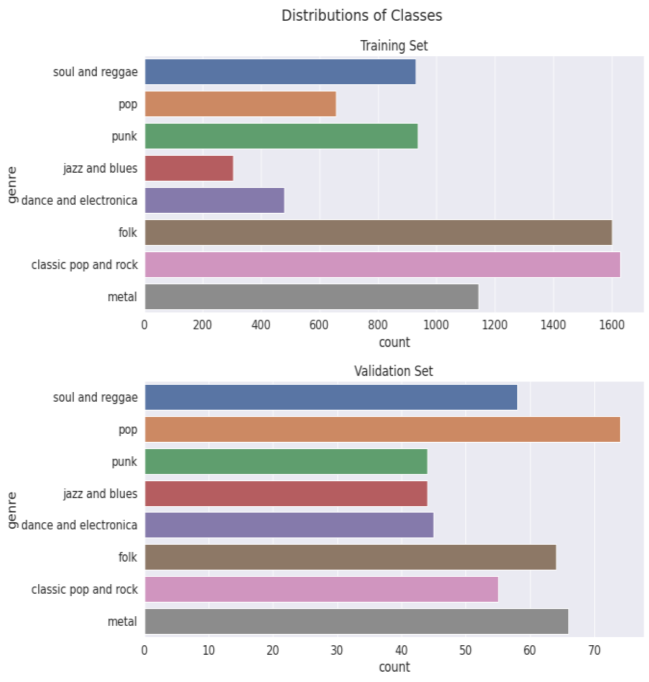


Figure 1- The distributions of the classes.

The feature set is divided into three categories:

Textual: features with values as words, namely lyrics.

Metadata: data that describes the song, such as duration and tempo.

Audio: uninterpretable features pre-extracted from 30-60 second snippets from each track, and capture timbre, chroma and Mel-Frequency-Cepstral-Coefficients (MFCC).

Figure 2 reveals the sampled nominal features do not follow a clear distribution; hence, we have assumed a gaussian distribution. Figure 3 reveals the sampled continuous features are gaussian distributed.

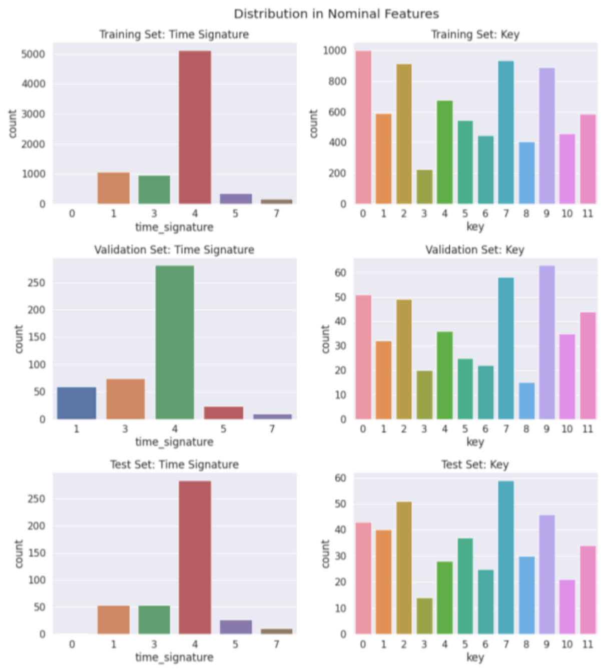


Figure 2- The distributions of a sample of the nominal features.

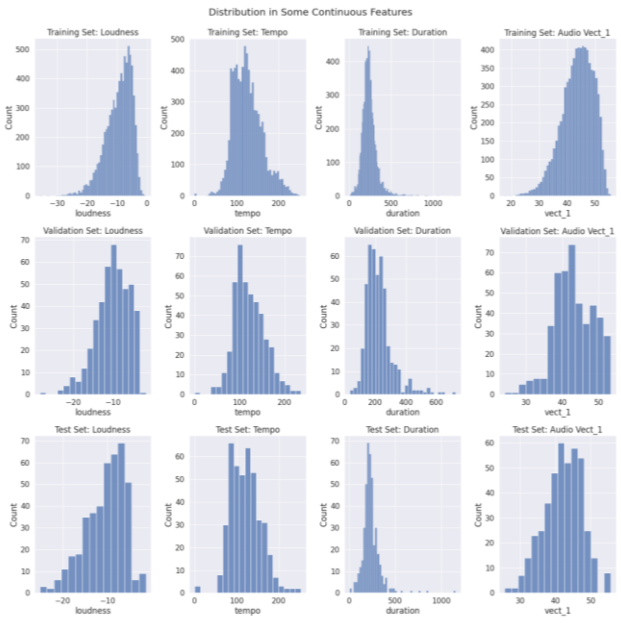


Figure 3- The distributions of a sample of the continuous features.

4 Approach

4.1 Feature Engineering

Since our lyrical tags come stemmed, it is necessary to stem each title to its root form. Stemming is a technique best illustrated with an example. Suppose two songs have lyrical tags “loved” and “love”: stemming would chop “d” from “loved” to make its root form. It affords us better comparison between text.

Next, we concatenate title and lyrical features since they are text-based and apply Term-Frequency-Inverse-Distance-Frequency (Tfidf) vectorization. This gives a normalized statistic (frequency-weight) for each word-root in the training’s textual features, describing how rare a word is in context, factoring in not only its frequency in song, but all songs in the training set (corpus). [6]

We did not remove stopwords, nor punctuation since artists might use punctuation marks in their titles as an identifying feature of their music, for example.

4.2 Feature Selection

4.2.1 Textual Features

The TfidfVectorizer engineers over 7300 continuous features (one each unique word root). The main issue with this feature set is its sheer size (computationally prohibitive).

To reduce dimensionality as well as improve expressiveness of the feature set, we filter textual features according to Mutual Information (MI). MI measures dependency between the target variable and feature. The higher the MI score, the higher the dependency. We screen out all features not meeting a threshold (arbitrarily low) of 0.0001, allowing us to drop over 3400 features. Examples of words we dropped are in Table 1.

Removals		
OO	Mama	Malon

Table 1- Example words removed based on MI.

We also explore which words are most dependent with each class. Examples of words with high MI with respect to Metal genre are in Table 2. Metal music clearly employs highly emotive and dark language.

Highest MI (Metal genre)			
Blood	Love	Dead	Hate

Table 2- Example words with Highest MI in Metal genre.

4.2.2 Metadata Features

We try to identify redundant features by removing any with correlation coefficient (CC) above 0.80 (arbitrarily high) – but remove only one in pair. However, none could be removed. We applied the same MI test from Section 4.2.1; however, none could be removed.

As a final feature selection technique, we simulate subsets of our metadata feature set taking the top K features and plot the error rate from classification. To test we stratify the training instances each time we split into train and test sets and use the robust 5-Fold Cross Validation framework, ensuring all partitions share same proportions of the class labels [7]. Our model was a **Random Forest Classifier (RF)**, which is an ensemble learning algorithm that employs bagging to improve its base algorithm: modified Decision Tree. The results are in Figure 4. We observe selecting all features leads to the lowest error rate.

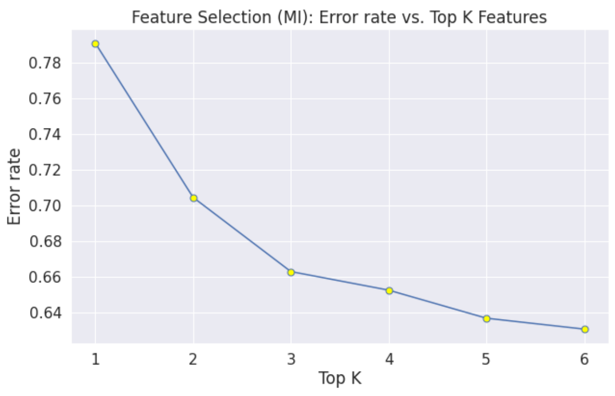


Figure 4- Error rate on different subsets of metadata.

4.2.3 Audio Features

We understood the 30-60 second snippets of audio are time series data. Hence, we did not apply CC screening to avoid losing the expressiveness of such features when taken as a collection.

We apply the same MI screening measures in Section 4.2.1. Like Section 4.2.2, we model the error rate of a **Gaussian Distributed Naïve Bayes Classifier (GNB)** on changing subsets of features. We selected GNB because it scales well to large feature and our features are gaussian distributed. The results from this are shown in Figure 5. We observe selecting all but one of the features leads to the lowest error rate.

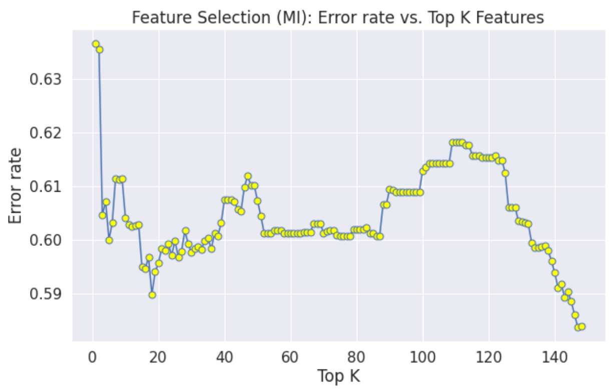


Figure 5- Error rate on different subsets of audio.

4.3 Classification Algorithms

We chose to evaluate the performance of seven supervised classification algorithms.

*assume default parameters in Sklearn unless stated. [8]

Zero-R (O-R): Predicts most frequent class label observed in training set.

Decision Tree (DT): Simple greedy construction strategy that produces set of if-then rules to follow. Measures the quality of a split with Gini impurity, which gets the likelihood of incorrect classification of a new instance of a random variable. Entropy could have equally been used.

Gaussian Naïve Bayes (GNB): Described in Section 4.2.3. Optimises posterior-probability. Although it assumes features are independent given the class, GNB works well when this assumption is somewhat inaccurate. GNB is suitable as we have observed gaussian distributed data in the feature set.

Random Forest (RF): Described in Section 4.2.2. Its base estimator is a modified DT because, at each candidate split it picks a random subset of the features rather than the full set. It applies bagging by fitting several of these modified DTs on various sub-samples of the dataset and uses averaging to improve accuracy and limit overfitting to the training data.

Logistic Regression (LR): Optimises posterior-probability. Transforms its output using the logistic sigmoid function to return a probability. Uses ordinal logistic regression. Does not assume conditional independence in features unlike GNB. Uses cross-entropy loss function. Requires features are scaled to zero mean and unit variance. To improve convergence, we set maximum iterations to 1000.

Multilayer Perceptron (MLP): Optimizes the log-loss function using stochastic gradient-descent-based weight-optimization solver. Learning rate is 0.002 throughout. Uses logistic sigmoid function for activation function for hidden layer. Like LR, we scale features, ensure maximum iterations is 1000, but also set hidden layer sizes to 100 and 50 neurons for the first and second hidden layers.

Stacking Ensemble with LR (ENS): Uses the power of each base estimator by using their output as input to the metaclassifier. Meta algorithm is LR and base estimators are GNB, RF, LR, and MLP.

5 Results and Discussion

5.1 Classification Results

We evaluated each algorithm on the following metrics:

Training Set Stratified 2-Fold Cross Validation Accuracy (TS-2FCV): Accuracy on training set, using a stratified-variant of 2-Fold Cross Validation (2FCV) (described in Section 4.2.2).

Training Set Class-Based Accuracy (TS-CB HO): Simple average of each class's accuracy on training set, using holdout strategy (HO).

Training Set Macro-F1-Score (TS-MF1): F-measure where relative contribution of precision and recall are equal on training set, using HO.

Validation Set Accuracy (VS-HO): Accuracy on validation set, using HO.

Validation Set Class-Based Accuracy (VS-CB HO): Same as item 2, except on validation set.

Validation Set Macro-F1-Score (VS-MF1): Same as item 3, except on validation set.

Our results for each model are in Table 3 and Figure 6.

Model	TS-2FCV	TS-CB HO	TS-MF1 HO	VS-HO	VS-CB HO	VS-MF1 HO
O-R	0.21	0.13	0.04	0.12	0.13	0.03
DT	0.46	1.00	1.00	0.43	0.41	0.42
GNB	0.51	0.55	0.53	0.65	0.61	0.61
RF	0.60	1.00	1.00	0.64	0.59	0.56
LR	0.56	1.00	1.00	0.54	0.51	0.50
MLP	0.53	1.00	1.00	0.57	0.53	0.52
ENS	0.64	1.00	1.00	0.67	0.64	0.62

Table 3- Accuracies/Scores for each classifier.

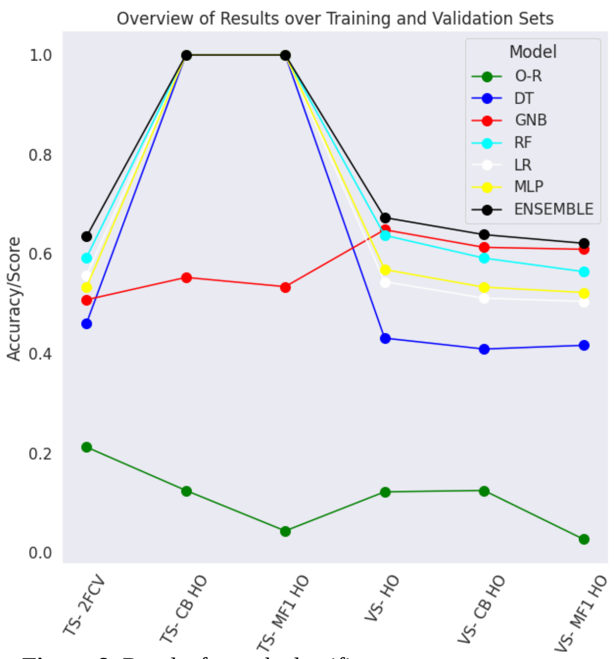
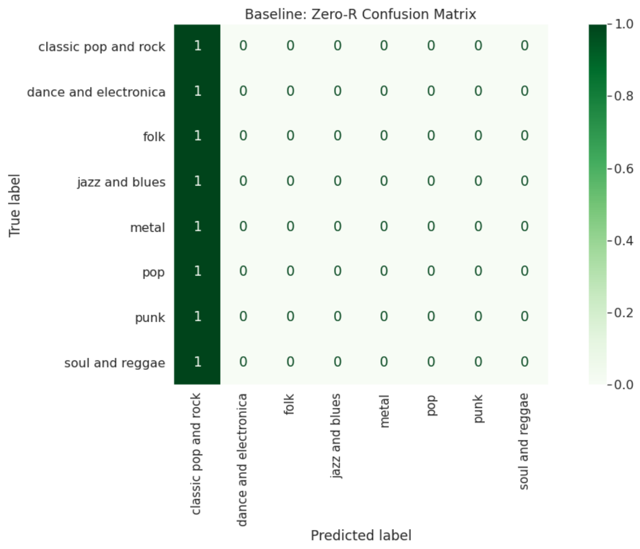


Figure 6- Results for each classifier.

We provide the confusion matrix of our baseline, benchmark and stacking ensemble in Figure 7, 8, and 9, respectively, to gain insight into the predictiveness of the classifiers on a class-by-class basis. The class-classification-accuracies lie in the diagonal of the confusion matrices.



True label	classic pop and rock	dance and electronica	folk	jazz and blues	metal	pop	punk	soul and reggae
classic pop and rock	1	0	0	0	0	0	0	0
dance and electronica	1	0	0	0	0	0	0	0
folk	1	0	0	0	0	0	0	0
jazz and blues	1	0	0	0	0	0	0	0
metal	1	0	0	0	0	0	0	0
pop	1	0	0	0	0	0	0	0
punk	1	0	0	0	0	0	0	0
soul and reggae	1	0	0	0	0	0	0	0

Figure 7- Normalized (row) Confusion Matrix for Zero-R on Validation Set.

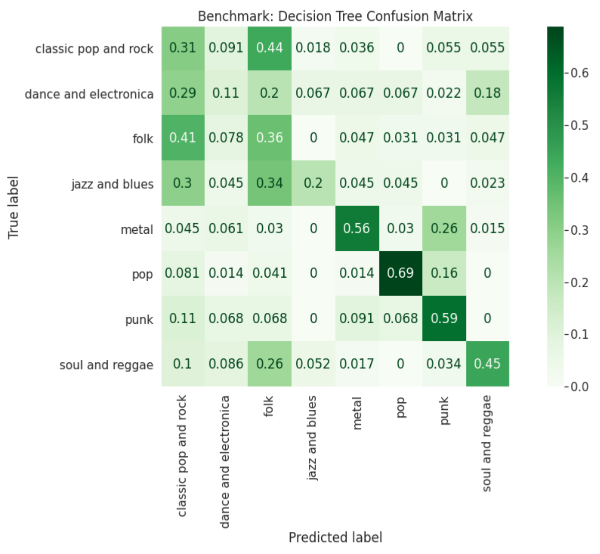


Figure 8- Normalized (row) Confusion Matrix for Decision Tree on Validation Set.

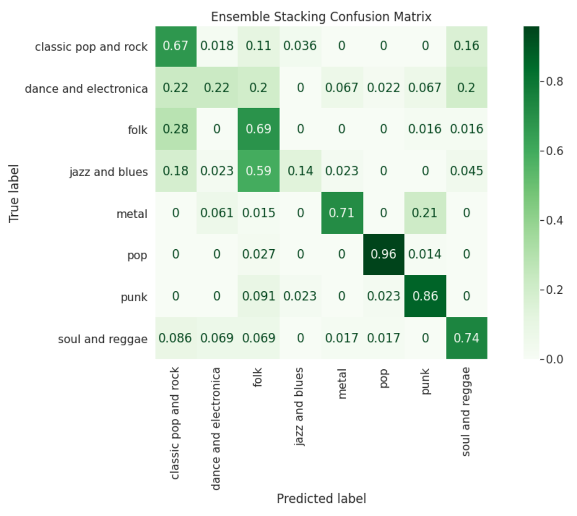


Figure 9- Normalized (row) Confusion Matrix for Stacking Ensemble on Validation Set.

5.2 Evaluation

With a mutli-class-label, we are interested in each class equally and do not want imbalanced data to influence our evaluation, so we apply simple/macro averaging in Metrics 2, 3, 5, 6. Thus, these metrics are most important.

Table 3 reveals ENS yields the highest accuracy/F1-score across all metrics relative to both the standalone models and the O-R baseline and DT benchmark, with scores 0.64, 1.00, 1.00, 0.67, 0.64, and 0.62. Figure 6 reveals the same information with the stacking ensemble's black-line above (or at) the rest.

Here, ENS demonstrates its advantage as a method that harnesses the strength of individual base estimators by taking each of their predictions and running a LR on top. Zero-R is the worst performing, with scores 0.21, 0.13, 0.04, 0.12, 0.13, 0.03.

We observe that five of the seven classifiers enjoy perfect accuracy in TS-CB HO and TS-MF1. However, this does not translate to the same strong performance on validation, demonstrating a case of overfitting. Interestingly, we see RF indeed controls the extent of its overfit compared to the DT, in line with theory (bagging). We also note that DT suffers most from overfitting, diagnosed in Section 5.3.

5.3 Error Analysis

Figure 8 and 9 reveal the least well-predicted genres by both DT and ENS are Jazz and Blues and Dance and Electronica with accuracies 0.11 and 0.2, and 0.22 and 0.14, respectively. This is a direct consequence of the class distribution in the training set being the least frequent genres, mentioned previously, since the ability to learn patterns and reliable probabilities is limited when these genres make up such a small percentage of the training set.

DT clearly overfits to the training data – see dark-blue line with weaker validation scores than training. Although simple and interpretable, DTs approach is sometimes too simple for datasets with complex structures – like ours due to dimensionality. Further, DTs are sensitive to small perturbations in the data and a minor change can result in a radically different tree.

MLP also suffers from overfitting; however, this could be ameliorated by adding a regularization term to the log-loss function to shrink-back the model parameters.

6 Conclusions

In this paper we delve into the dataset to engineer and extract an expressive and tractable set of features. We compare the performance of a stacking ensemble to four standalone methods, as well as Zero-R and DT. We witness how it can be applied to MGC and harness the unique strengths of its base estimators by using their output as input to a Logistic Regression meta classifier.

In future, we might explore Part-of-Speech tagging and Lemmatization on title and tags, as compared to Stemming. We could also explore Principle Component Analysis for engineering orthogonal features.

References

[1] Samson, Jim. “Genre”. In Grove Music Online. Oxford Music Online, 2012.

[2] George Tzanetakis, Student Member, IEEE, and Perry Cook, Member, IEEE, Musical Genre Classification of Audio Signals. In IEEE Transactions on Speech and Audio Processing, Vol. 10. No. 5. 2002.

[3] T. Bertin-Mahieux, D. P.W. Ellis, B. Whitman, and P. Lamere. The million-song dataset. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011.

[4] A. Schindler and A. Rauber. Capturing the temporal domain in Echonest Features for improved classification effectiveness. In Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR) , 2012.

[5] M. A. Munoz, L. Villanova, D. Baatar and K. Smith-Miles. Instance spaces for machine learning classification. Machine Learning, volume. 107, pp. 109-147, 2018.

[6] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and Style Features For Musical Genre Classification By Song Lyrics. In Conference Paper, 2008.

[7] Stone, M. An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike’s Criterion.” In Journal of the Royal Statistical Society: Series B (Methodological), 1977.

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.