COMP90049 Assignment 2 Report

Anonymous

1 Introduction

The rise of music-streaming services has made music ubiquitous. Recommending suitable music for different users is the goal of every music software. The genre of music is one of the main characteristics of music and is also an important factor in whether listeners choose to listen to a song. Being able to automatically assign music to the genre to which it belongs could be a huge benefit to music software.

The goal of this project is to build a machine learning model that predicts music genres based on the audio, text, and metadata features of music. Each song will be labelled from one of eight genres, including Soul and Reggae, Pop, punk, Jazz and Blues, dance and electronica, folk, classic pop and rock, and metal.

The Dataset used in this paper mainly came from the Million Song Dataset (MSD) and extracted the audio content with time characteristics. Techniques that have been explored in MSD include the use of KNN in recommendation systems and Vowpal Wabbit (VM) algorithms designed for large-scale learning. (Thierry Bertin-Mahieux et al., 2011) In the audio content, the time feature is extracted based on the timing alignment vector sequence. (Schindler & Rauber, 2012) This method is significantly better than its predecessors in terms of the classification of music types, namely simple time-varying statistics.

Therefore, the assumption I will make in this paper is that models capable of learning complex linear inseparable data sets will perform well in this classification problem. To test this hypothesis, I will present three machine learning models as solutions in this article.

2 Dataset and Features

The features used in this dataset are divided into three categories: audio, metadata, and text features. Metadata features include title, loudness, speed, key, mode, duration, and time signature. Text features include the lyrics of each song, while audio features represent audio features such as timbre, chromaticity, and "Mel frequency ceptrum factor" (MFCC) aspects.

All the data is divided into predefined training set (7678 songs), development set (450 songs) and test set (428 songs). Among them, training set and development set are used to evaluate the model after pre-processing, while test set is used to test the accuracy of the model. For the pre-processing of text and digital features, the appropriate methods are used.

For data with numerical characteristics, standard scaler is used for processing. The standardized data is obtained by fitting the training set with the scaler and then converting the training data and validation data. This is done using the StandardScaler() function of Sci Kit Learn. This process also involves normalizing the data to prevent overfitting.

For data on text features (lyrics and titles), pre-processing includes normalizing the first letter and removing stop-words to normalize the data. Here, we applied the Term Frequency Inverse Document-Frequency algorithm (TFIDF) to convert the text data into numerical data. Firstly, the training class is used to complete the transformer, and then the transformer is used to transform the training set, validation set and test set, so as to filter out the common words and retain the important words. However, after such pre-processing, because the words in the title and tags change greatly, many sparse filling characteristics occur, which makes it difficult to find the data characteristics of the model. Therefore, we use principal component analysis (PCA) algorithm to reduce the dimension of trait matrix. Unfortunately, the initial results showed a significant decrease in accuracy after the inclusion of title features. This is mainly because SVM and MLP models are sensitive to data, and some feature values of the data are too large, leading to a great increase in the weight of the feature. The training data needs to be standardized. Therefore, I chose to standardize the pre-processed text features together, and the accuracy was 0.16 and 0.64 before and after standardization.

3 Models

In this project, the baseline model used is the ScikitLearn's DummyClassifier. This classifier is the most basic and widely used classifier in machine learning as the minimum baseline of classifier accuracy to determine whether the selected classifier is suitable or not. It also provides some ideas about what conclusions can be drawn from using simple models for complex problems.

In addition, I chose the three main models for classifying types of music, one is a multiplayer feedforward network perceptron (MLP) classifier, another is support vector Machine (SVM) classifier, and also the use of weak classifier (decision tree) iterative training to get the optimal model of Light Gradient Boosting Machine (LGBM) model. These three models have strong classification ability and can solve the very complex classification problem of pattern distribution, but there are some differences in structure and function.

As mentioned earlier, the data set used in this project contains characteristic data with high latitude and linear inseparability. MlP is based on neural network, while SVM is based on linear computation, but both have the ability to learn complex linear non-fractional data sets. (Stanislaw Osowski et al., 2004) The two models are often compared. The advantage of LGBM is that they combine multiple weak models into a strong model by boosting, which can effectively prevent the overfitting effect. I adjusted the parameters of each model to get the best performance and compared the results.

4 Model Analysis

For the three models that have been trained, a learning curve is drawn to reflect their performance.

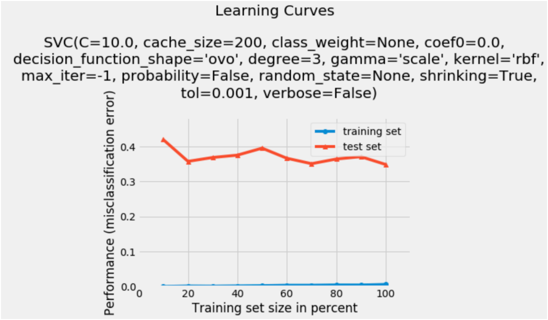


Figure 1. Learning curve of the SVM model.

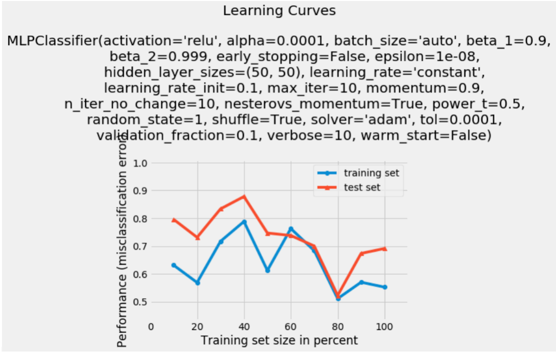


Figure 2. Learning curve of the MLP model.

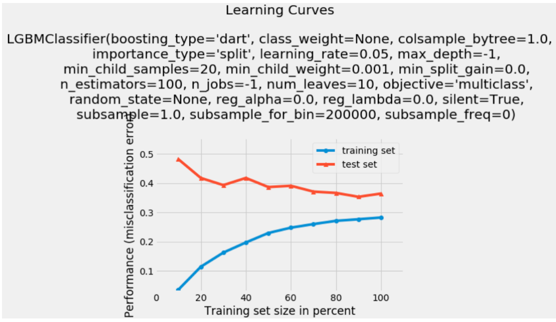


Figure 3. Learning curve of the LGBM model.

The learning curves of all three models show signs of underfitting. (Figure 1, 2, 3) To make the model fit the training data better, we need to adjust and optimize the parameters.

4.1 Baseline classifier

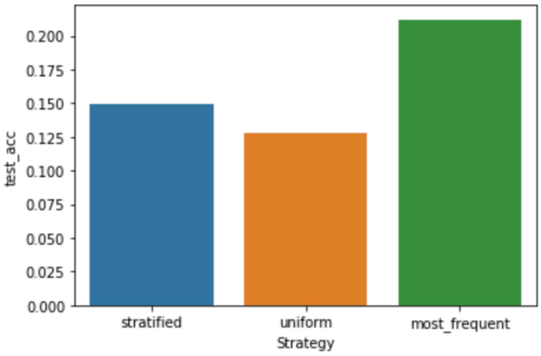


Figure 4. Accuracy of baseline model

The baseline model uses ScikitLearn's DummyClassifier to construct, and uses three strategies of "stratified", "uniform", "most_frequent" respectively, to predicts the label of a new song by learning the label distribution of the training set. The accuracy is 14% for Stratified, 10% for uniform strategy and 12% for most frequent strategy (zero-Rule). (Figure 4)

4.2 SVM classifier

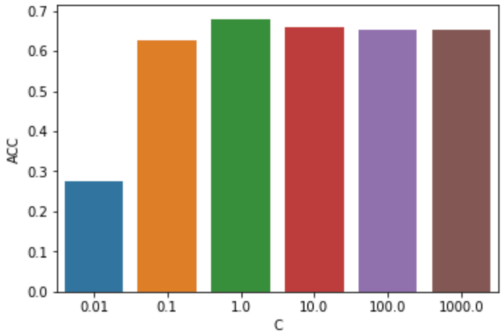


Figure 5. SVM Accuracy vs C parameter.

The C parameters of the SVM classifier is the main function of balance and the complexity of support vector classification error rate between the two relations, used for handling margin between classes. When the coefficient C is larger, there will be more support vectors, resulting in lower bias and variance and easy overfitting. And the smaller C is, the less fitting is likely to be. So, we need to find an optimal value to keep a good balance between variance and bias, which is 1.0 from Figure 5.

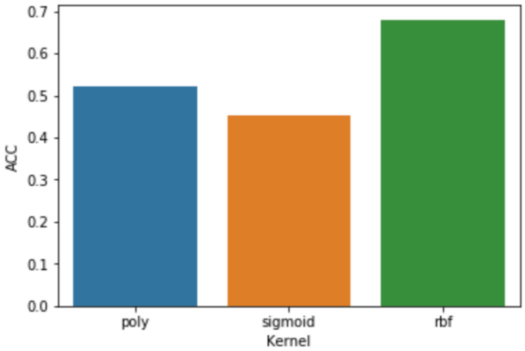


Figure 6. SVM Accuracy vs Kernel type.

The kernel type will affect the performance of the model, and kernel function can improve the feature dimension of the model, so that SVM has better nonlinear fitting ability. Among them, RBF Kernel performed best in terms of performance. (Figure 6)

The gamma coefficient can control some of the complexity of the model. A larger gamma means that the mapping is more dimensional, and the model is more complex, which will increase the bias and reduce the variance. A lower gamma means that the model is more compact, which will reduce the deviation and increase the variance, which will lead to a smoother result. The gamma parameter selection optimizes "scale" to make it lower to increase its generalization ability and utility value.

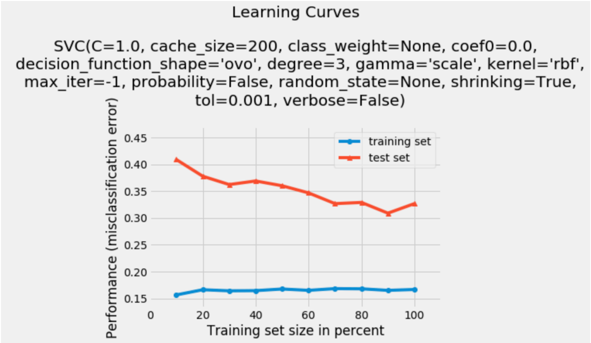


Figure 7. Learning curve of the re-tuned SVM model.

As can be seen from the figure 6, the training error of the SVM model after adjustment has decreased, while the verification error remains almost unchanged.

4.3 MLP Classifier

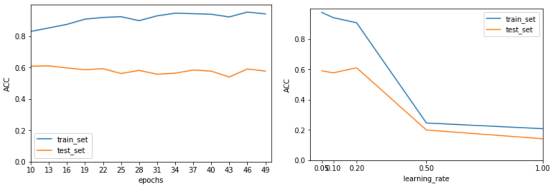


Figure 8. Acc vs Epochs and learning rate. SGD solver.

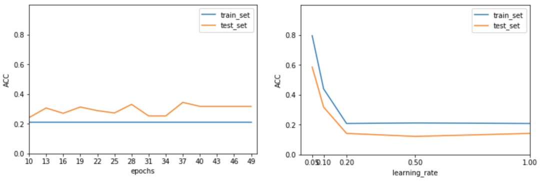
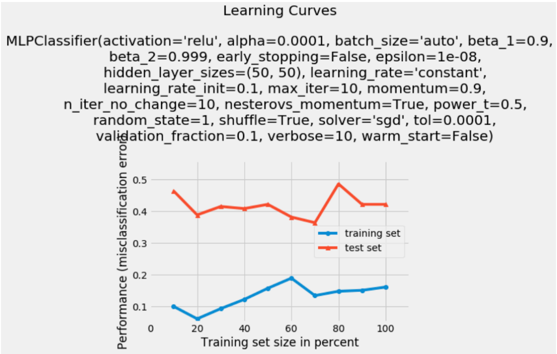


Figure 9. Accuracy vs Epochs and learning rate. Adam solver.

For the MLP classifier, the parameters tuned includes the network configuration (size of the hidden layers) the solver and activation function used in the training process, the learning rate.

Figure 8 and 9 shows that the behaviour of solvers used for MLP has a small number of epochs required for fitting the model, but has a large impact on the learning rate. For Adam and

SGD classifiers, the number of iterations required to achieve the best validation accuracy in the Adam solver was similar to that in the SGD solver, about 37-40 epoches were required. But in general, SGD solvers are more accurate than Adam solvers. At the same learning rate, SGD solvers also have higher accuracy than Adam solvers.



```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
beta_2=0.999, early_stopping=False, epsilon=1e-08,
hidden_layer_sizes=(50, 50), learning_rate='constant',
learning_rate_init=0.1, max_iter=10, momentum=0.9,
n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
random_state=1, shuffle=True, solver='sgd', tol=0.0001,
validation_fraction=0.1, verbose=10, warm_start=False)
```

Figure 10. Learning curve of the re-tuned MLP model.

In order to optimize the model, the SGD solver with constant learning rate and Relu activation function was used, and the optimal super parameter found for MLP classifier was network configuration (50, 50). At the same time, the regularization intensity is reduced, and the maximum number of iterations is increased. After parameter readjustment (Figure 10), the training error and validation error of MLP model are both decreased.

4.4 LGBM Classifier

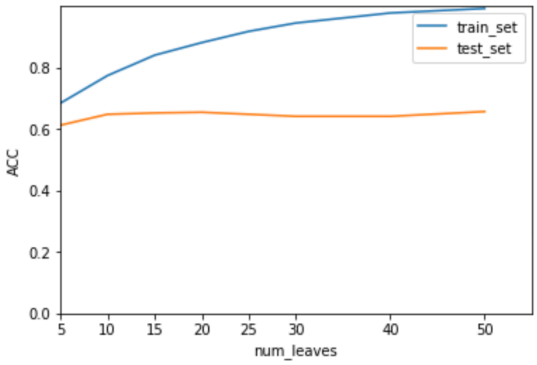


Figure 11. Accuracy vs num_leaves

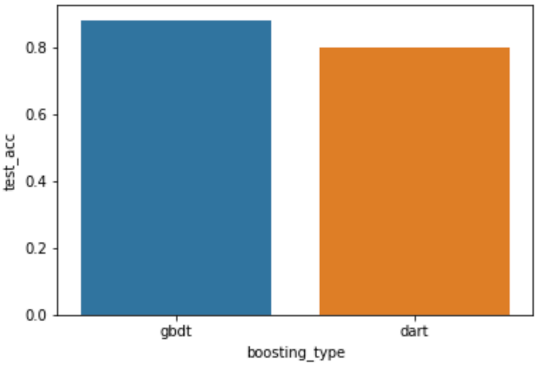
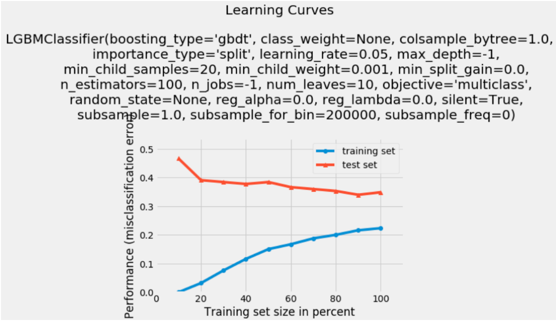


Figure 12. Accuracy of gbdt vs dart type.

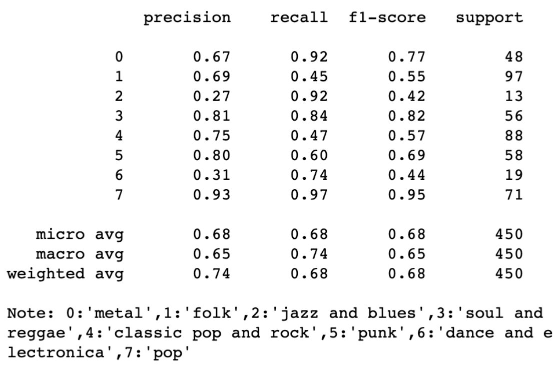


```
LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
importance_type='split', learning_rate=0.05, max_depth=-1,
min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
n_estimators=100, n_jobs=-1, num_leaves=10, objective='multiclass',
random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True,
subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

Figure 13. Learning curve of the re-tuned LGBM model.

For LGBM model, the parameter to be adjusted is the number of leaf nodes, which is the main parameter to control the complexity of tree model. The problem of overfitting can be alleviated by reducing the number of leaf nodes. However, as shown in Figure 11, the number of leaves does not change much for the accuracy of the test set. As shown in figure 12, boosting Gbdt type has higher accuracy than Dart. After adjusting the parameters, the training error of LGBM model decreases, but the verification error basically remains unchanged.

4.5 Measurement



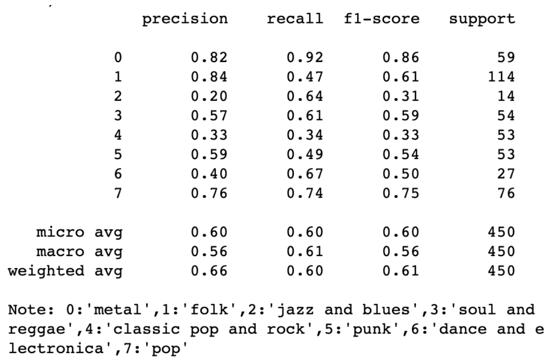
```
              precision    recall  f1-score   support

           0       0.67      0.92      0.77        48
           1       0.69      0.45      0.55        97
           2       0.27      0.92      0.42        13
           3       0.81      0.84      0.82        56
           4       0.75      0.47      0.57        88
           5       0.80      0.60      0.69        58
           6       0.31      0.74      0.44        19
           7       0.93      0.97      0.95        71

   micro avg       0.68      0.68      0.68       450
   macro avg       0.65      0.74      0.65       450
weighted avg       0.74      0.68      0.68       450

Note: 0:'metal',1:'folk',2:'jazz and blues',3:'soul and
reggae',4:'classic pop and rock',5:'punk',6:'dance and e
lectronica',7:'pop'
```

Figure 14. Classification report of the SVM model.



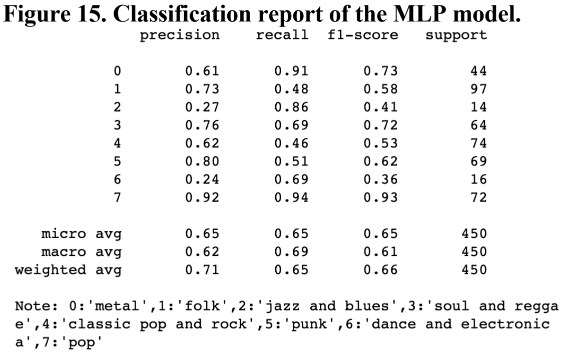
```
              precision    recall  f1-score   support

           0       0.82      0.92      0.86        59
           1       0.84      0.47      0.61       114
           2       0.20      0.64      0.31        14
           3       0.57      0.61      0.59        54
           4       0.33      0.34      0.33        53
           5       0.59      0.49      0.54        53
           6       0.40      0.67      0.50        27
           7       0.76      0.74      0.75        76

   micro avg       0.60      0.60      0.60       450
   macro avg       0.56      0.61      0.56       450
weighted avg       0.66      0.60      0.61       450

Note: 0:'metal',1:'folk',2:'jazz and blues',3:'soul and
reggae',4:'classic pop and rock',5:'punk',6:'dance and e
lectronica',7:'pop'
```

Figure 15. Classification report of the MLP model.



```
              precision    recall  f1-score   support

           0       0.61      0.91      0.73        44
           1       0.73      0.48      0.58        97
           2       0.27      0.86      0.41        14
           3       0.76      0.69      0.72        64
           4       0.62      0.46      0.53        74
           5       0.80      0.51      0.62        69
           6       0.24      0.69      0.36        16
           7       0.92      0.94      0.93        72

   micro avg       0.65      0.65      0.65       450
   macro avg       0.62      0.69      0.61       450
weighted avg       0.71      0.65      0.66       450

Note: 0:'metal',1:'folk',2:'jazz and blues',3:'soul and regga
e',4:'classic pop and rock',5:'punk',6:'dance and electronic
a',7:'pop'
```

Figure 16. Classification report of the LGBM model.

The tuned model was used to verify the processed data. The SVM model verified 68% accuracy, MLP model 60% accuracy, and LGBM verification 65% accuracy (figure 14, 15, 16). Among them, SVM linear model and LGBM model performed well in predicting pop, soul and Reggae, metal and other music genres, while MLP model performed well in predicting pop, Punk, soul and Reggae, folk, metal and other music genres, and the F1 value of these types was all above 0.6. They're a little less good at predicting genres like Jazz and Blues, dance and electronica, possibly because they have less of a training set. Another possible reason is musical commonality. For example, a song can be labelled either jazz or electronic at the same time. This common feature between the music indicates that the tags are not mutually exclusive, and the problem of multiple tags may be one of the reasons that the performance of the model is not as good.

5 Conclusion

Overall, the performance of the three models was similar, with support vector machines scoring slightly higher than MLP and LGBM models in the test index. All three models are superior to the baseline model, suggesting that they are suitable candidates to solve this problem. My assumption is that models that can learn complex linear nonfractional data perform well in this experiment. This assumption has been verified in SVM and MLP models, but LGBM models also perform well. During the validation phase, the learning curve shows signs of inappropriate fitting. Some models improve performance through parameter tuning and feature redesign.

Limitations of this project may include that music has problems with multi-genres, genre labels are not mutually exclusive, and some similar types of music may have common characteristics, such as jazz and electronics may have similar audio characteristics, which will lead to the degradation of the classification performance of the model. Another disadvantage is that I did not screen the features in this experiment. If the features could be further screened, the model may be more optimized.

6 Reference

Schindler, A., & Rauber, A. (2012, October 29). *Capturing the Temporal Domain in Echonest Features for Improved Classification Effectiveness*. ResearchGate; unknown. https://www.researchgate.net/publication/266171053_Capturing_the_Temporal_Domain_in_Echonest_Features_for_Improved_Classification_Effectiveness

Thierry Bertin-Mahieux, Daniel, Whitman, B., & Lamere, P. (2011). *The Million Song Dataset*. ResearchGate; unknown. https://www.researchgate.net/publication/220723656_The_Million_Song_Dataset

Stanislaw Osowski, Krzysztof Siwek, & T. Markiewicz. (2004, February). *MLP and SVM networks - a comparative study*. ResearchGate; unknown. https://www.researchgate.net/publication/4095905_MLP_and_SVM_networks_-_a_comparative_study