Music Genre Classification with Machine Learning Algorithms

COMP90049

1. Introduction

This is a report for COMP90043 Introduction of Machine Learning. In this project, students are required to build and analyse machine learning algorithms to predict the genre of music with the datasets compiled by Schindler et al. (2011). Students are given datasets for training, validating and testing classification models. In the datasets, there are three types of features; audio, metadata and text. The audio features and metadata are in forms of numeric values, including binary, integer and floating points numbers, whereas the text and one of the metadata features are textual values. This report will analyse how to classify genre with basis on audio, metadata and text features.

1. 1 Hypothesis

Prior to building and evaluating classification models on the genre, it is crucial to formulate a hypothesis on what aspects of data would influence the productivity of the prediction. As mentioned in the introduction, the datasets consist of audio, metadata and text data, including categorical, continuous and binary attributes. Text data contains tags that represent keywords appeared in the lyrics. The tags could be viewed as the most significant predictor of the music genre since lyrics have a direct relationship with the characteristic of music. While some of the tags are ordinary and do not have implications on the music genre, many words, such as 'love', could provide meaningful distinctions on types of the genres. The audio data, on the other hand, indicates chroma, timbre and Mel-frequency cepstral coefficients (MFCC) aspects, that are extracted from the 30 to 60 snippets. The music genres, such as metal and jazz, have a clear difference in timbre and other audio related components. Therefore, the audio data could also be a factor to distinguish the genres. The metadata has the title, loudness, tempo, time signature, key, mode and duration of the music. Some of the metadata could be useful for the prediction as correlated with the audio of music, e.g. loudness and time signature. The title of the music, however, might not be a large factor of prediction, because the title is a concise summary of music and different genres of music would have similar titles.

Thus, audio or text features could be considered as significant factors of prediction on the genre classification.

2. Related Work

There have been a number of attempts in music genre classification tasks using machine learning algorithms in academic communities. In music genre classification task, not only instruction information but also rhythmic information is considered to be important (Tsunoo et al. 2009). Annesi et al. (2007) took an approach to distinguish the instrument that

operates in the lower frequency, such as the drum and bass, by extracting the number of beats and apply this for the music genre prediction. The graph below shows the result of Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and Multi-category Proximal Support Vector Machine (MPSVM) conducted on various features by Li et al. (2003). As can be seen from the graph, the accuracy of each classifier on audio features and the best combination of them that output the best results. Also, the beat and pitch are not as prominent predictors as FFT and MFCC. SVM shows more than 60 per cent of accuracy on FFT, while MPSVM appears to be the lowest accuracy on FFT and MFCC. Overall, the combination with the beat, FFT, MFCC and Pitch manage the best accuracy score over 70 per cent with LDA. That being said, the combination of features could bring better results than using an individual feature.

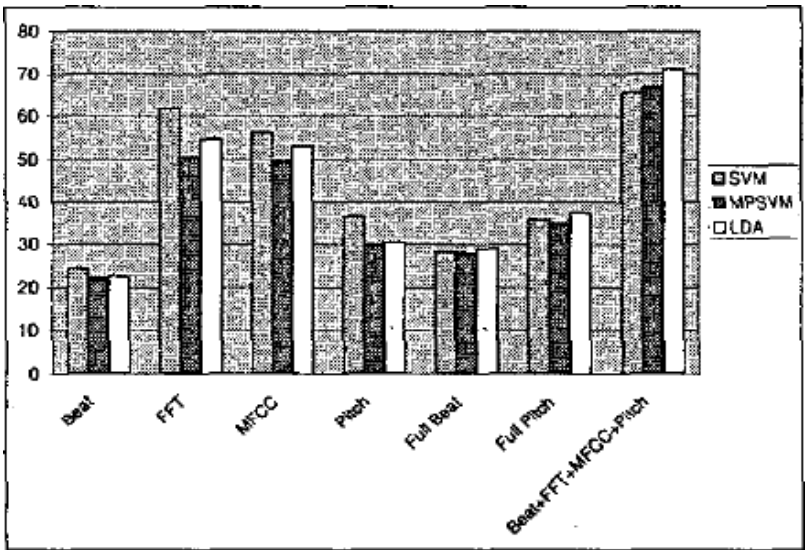

Figure 1. Accuracy Comparison of Different Methods

3. Model Training Methodology

In this project, we are required to handle different data types, including floating points number, integer, binary and text. In order to fit into models, it is necessary to convert them into appropriate forms. Moreover, the datasets are to be separated into audio, text, metadata for the project purpose. Since the datasets are in CSV files, Pandas library was used to read the files and transformed into Pandas data frame. TrackID in the datasets is for mapping songs to their labels so that it is to be removed after merging labels and datasets. Following sections will discuss analysis on data and how the data was handled.

3.1 Dataset Check and Correlation between Features

In the training files, there are more than 7K instances, and the analysis of the training data is the first step to solve the multi-class classification challenge. For visualising distribution and correlation between features, Pandas scatters matrix was used. Figure 2 illustrates the

correlation and distributions of metadata features in datasets in the matrix. As can be seen from Figure 2, the genre class occurrence distribution is unbalanced, and ‘classic pop and rock’ presents 1629 times in the data, while ‘jazz and blues’ appears 303 times. The relationship between genre and each metadata feature demonstrates that the loudness and duration have a relatively large correlation with the genre. The key, mode and key are evenly distributed in the dataset and seem to have a weak relationship to the genre.

Also, checking null in the datasets is important as it may produce unforeseen errors in model prediction. It has been checked if the data contains null and no null has been found.

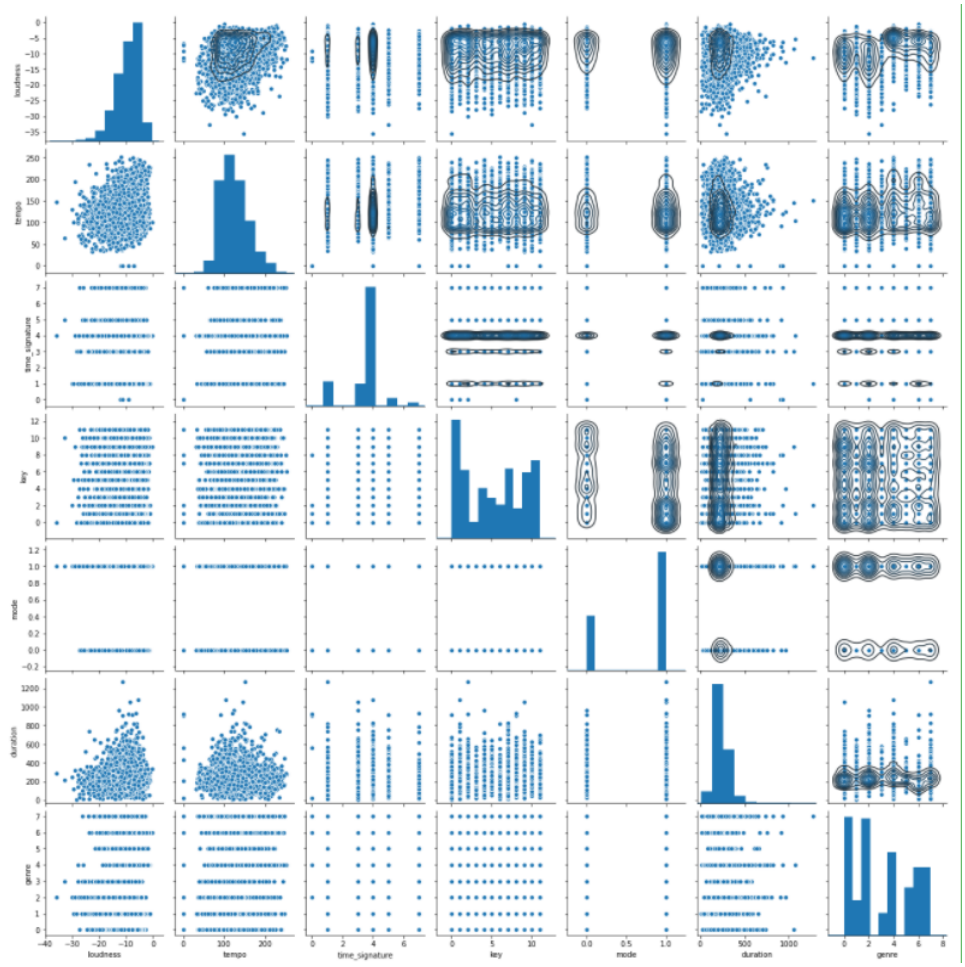

Figure 2. Panda Scatter Plots on the Training Data

3.2 Handling Floating-point Values

Two approaches were taken to handle floating-point value; put them into bins and utilise min-max scalar from Sklearn pre-processing library to rescale all value to the range of 0 to 1. Binning allows some models that accept only categorical values to handle floating values. Converting floating values to the range of 0 to 1 eliminates negative values for some functions that accept only positive values.

3.3 Handling Text values

The tags and title features contain a comma-separated list of words and TfidfVectorizer was employed to extract lists and convert them to a matrix of TF-IDF features with scores for each word. TF-IDF stands for Term Frequency Inverse Document Frequency and measures importance of a word. Looking into Figure 3, it displays the top 100 words with its number of appearances and some words appear more than 6000 times. Words occurred in only a few titles are easier to predict compared to high-frequency words. TF-IDF scores important words higher and common words lower. The TfidfVectorizer has features to ignore list of defined words for avoiding ordinal stop-words that carries only little meaning and might increase noise on the prediction (e.g. 'the', 'to' and 'a'). From Figure 3, the most frequent words are stop-words, and they might have a negative impact on the performance of prediction.

```
the     6590
to      6258
a       6179
and     6126
i       5904
        ...
well    1186
then    1166
or      1166
some    1157
mind    1152
Length: 100, dtype: int64
```

Figure 3. Frequency of Words in Tags

3.4 KBest Feature-Selection

Feature selection is a technique to eliminate features that are not highly relevant to the target variable. SelectKBest library in Sklearn automatically chooses features that contribute the most to the prediction according to k-highest score. To remove the number of irrelevant features results in not only less noise on the prediction, but also improvement of execution time as fewer attributes to fit in classifiers. When applying KBest feature selection, it is essential to choose an appropriate scoring function. Chi-square is a scoring function that measures the difference between the expected and observed frequencies of a set of event or variables. Figure 4 shows how Chi-square calculates the score.

$$\chi^2 = \frac{(O(W) - E(W))^2}{E(W)} + \frac{(O(X) - E(X))^2}{E(X)} + \frac{(O(Y) - E(Y))^2}{E(Y)} + \frac{(O(Z) - E(Z))^2}{E(Z)}$$

Figure 4. The Formula of Chi-Square

4. Results

Seven different classifiers and one baseline are tested on multiple features and the combination over the validation datasets. Figure 5 and 6 below perform the results of the testing; the x-coordinate axis is accuracy, and the y-coordinate axis is a list of feature-sets tested on the classifiers. The hypothesis made at the early stage of the report seems to be correct. It can be seen that the metadata and title are relatively low accuracy indicating they are a poor predictor, while text and audio features show high accuracy on the prediction. The highest score (0.6822) was achieved by Logistic Regression classifier with the combination of binned audio and text. Compared to Kbest-selected features, the features without KBest-selection yield little better scores in overall classifiers. This might be because inappropriate k was chosen for the KBest feature-selection, and some important features were missed. Also, the binned numerical features achieved higher scores than the rescaled numerical features for all models. The reason is that the binned features groups continuous variables to the number of the intervals, and it becomes easier to be interpreted for the models.

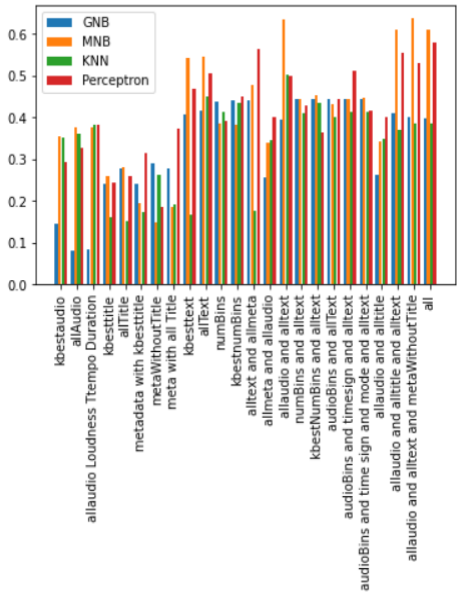

Figure 5. Accuracy on Attribute Sets

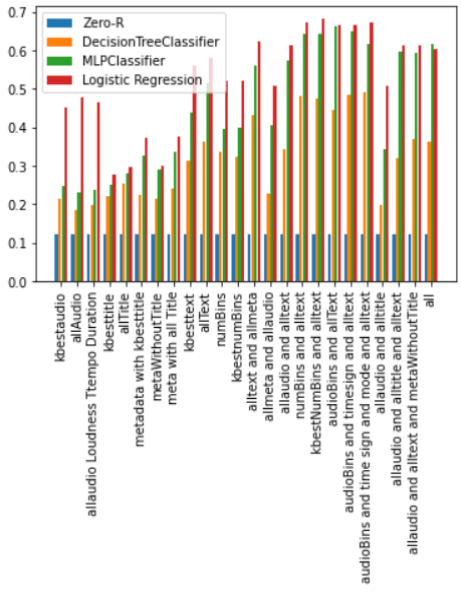

Figure 6. Accuracy on Attribute Sets

4.1 Classifier Analysis with Base Line

Zero-R baseline was employed to measure the performance of the classifiers. Zero-R is the simplest classifier method, which ignores all predictors and predicts only based on a target variable. In this test, the Zero-R gave a baseline of 0.1222 accuracies. From Figure 6, it is evident that all the classifiers with sets of features achieved higher scores than the baseline.

4.2 Error Analysis

Figure 7 describes confusion matrices of the binned numerical and text features for Zero-R, Logistic Regression, MLP and Decision Tree. Zero-R predicted the most correct-number at 55 in the first column. However, Zero-R failed to predict correctly for other columns because Zero-R is a naïve method and dependent on only a label. Decision Tree had low accuracy, especially in the first column. The reason might be overfitting of training data since Decision Tree is deterministic and the small noise could significantly impact on the result. Logistic regression had a lower error rate than other classifiers. This might be because the size of the training data is large enough, and the binned variable reduced the noise.

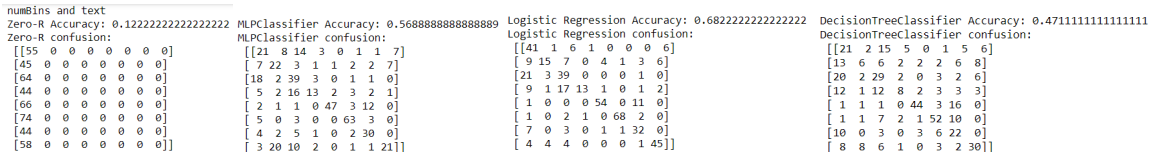

Figure 7. Confusion Matrices

4.3 Classifier Tuning

Often time, we do not immediately know what the optimal hyperparameter should be given to the model structure. In this project, logistic regression, multi-layer perception and decision tree classifiers were selected for the parameter tuning with Grid Search CV from Sklearn library. Grid Search CV is an exhaustive cross-validated search to find optimal hyperparameters with ‘fit’ and ‘score’ methods. Logistic regression has several parameters for tuning, e.g. penalty and C. The C is an inverse of regularisation strength and must be a positive float. Multi-layer perception has been set with grid search for the following hyperparameters: activation, hidden layer sizes, solver, alpha, and learning rate. The hidden layer size is the main parameter for the neural network, which defines the number of neurons in the hidden layers. Lastly, the best hyperparameters of the decision tree were examined by grid search: max-leaf nodes, min sample split, max depth and criterion.

Figure 7 is the table of the Grid Search’s result on the multi-layer perceptron with the pair of binned numerical and text features. As can be seen from the table, Grid Search automatically tried all pairs of the targeted hyperparameters and chose the pair with the highest score. In this case, 0.001 and (50,50,50) were chosen for alpha and hidden layer sizes.

	alpha	hidden_layer_sizes	Accuracy
0	0.001	(50, 25, 10)	0.627639
1	0.001	(50, 50, 50)	0.640791
2	0.002	(50, 25, 10)	0.582572
3	0.002	(50, 50, 50)	0.603412
4	0.003	(50, 25, 10)	0.585436
5	0.003	(50, 50, 50)	0.615395

Figure 8. The Table of Grid Search Results

5. Conclusions

The classification task on the music genre is complex and still requires improvement with machine learning algorithms. The related researches on this classification task have focused on classifier optimisation, model comparison, and optimal features for the prediction. The analysis of this project demonstrates that text and audio features are highly correlated with the music genre prediction, and the appropriate classifier ought to be chosen on the basis of data types and performance. The combination of different features could bring a better result. Also, it is important that the data type in the datasets should be handled carefully and transform into a different form if necessary. At the pre-processing stage, analysing data and cleaning unnecessary data, such as stop-words, could reduce the noise in the result. Classifier tuning is helpful to produce an optimal result out of the model.

In future research, it might be helpful to find different combinations of features that have not been discovered. Tzanetakis and Cook (2002) said that emotion and voice style could be investigated as possible categories for classification tasks. It is likely that the undiscovered features and combinations would yield better result on the prediction on the music genre.

References

Annesi, P, Basili, R, Gitto, R, Moschitti, A & Petitti, R 2007, *Audio Feature Engineering for Automatic Music Genre Classification*, 1 June, viewed 16 October 2020, <http://casa.disi.unitn.it/moschitti/articles/RIAO2007.pdf>.

Bertin-Mahieux, T, Daniel P W Ellis, Whitman, B & Lamere, P 2011, *THE MILLION SONG DATASET*, viewed 14 October 2020, <http://ismir2011.ismir.net/papers/OS6-1.pdf>.

Schindler, E & Rauber, A 2012, *Capturing the temporal domain in echonest features for*

improved classification effectiveness, CiteSeer, viewed 14 October 2020, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.474.2673>.

Tao Li & Tzanetakis, G 2003, *Factorsinautomaticmusicalgenreclassificationofaudiosignals*, IEEE Xplore, pp. 143–146, viewed 14 October 2020, <https://ieeexplore.ieee.org/abstract/document/1285840>.

Tsunoo, E, Tzanetakis, G, Ono, N & Sagayama, S 2009, *Audio genre classification using percussive pattern clustering combined with timbral features*, IEEE Xplore, pp. 382–385, viewed 16 October 2020, <https://ieeexplore.ieee.org/abstract/document/5202514>.

Tzanetakis, G & Cook, P 2002, *'Musical genre classification of audio signals'*, IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293–302.

Appendix

Appendix A

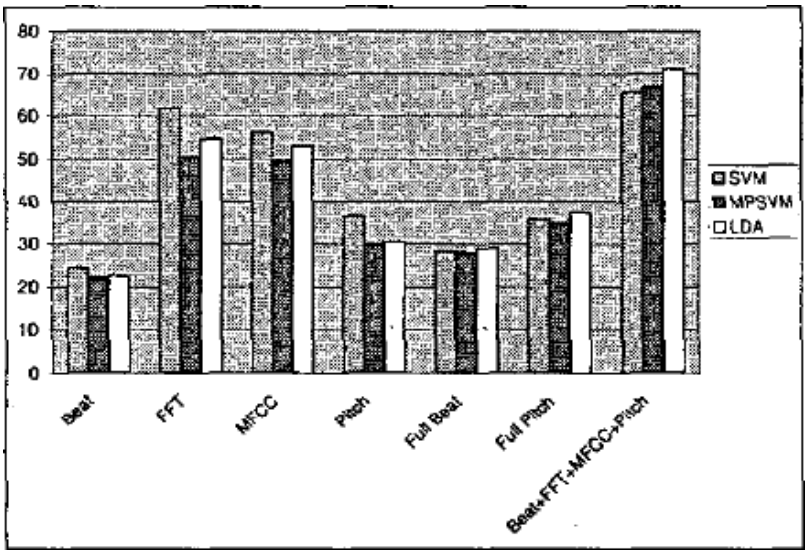

Figure 1. Accuracy Comparison of Different Methods

Appendix B

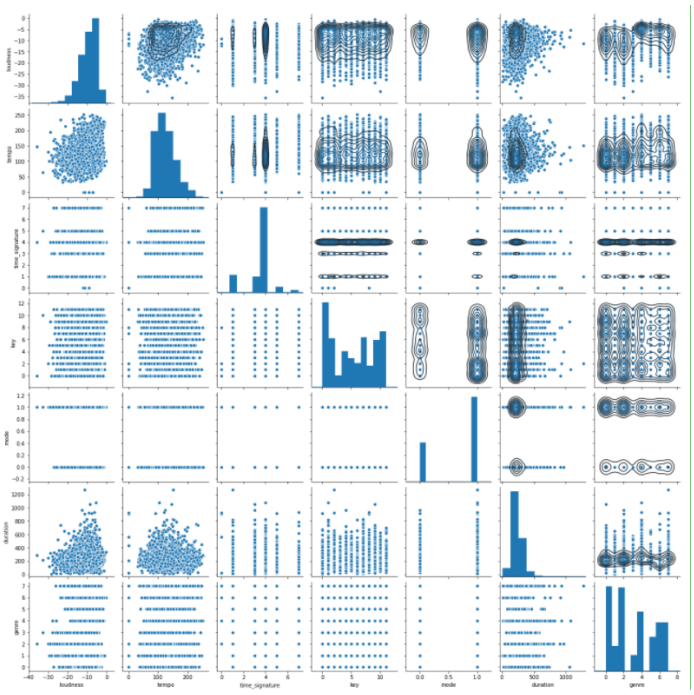

Figure 2. Panda Scatter Plots on the Training Data

Appendix C

```
the       6590
to        6258
a         6179
and       6126
i         5904
          ...
well      1186
then      1166
or        1166
some      1157
mind      1152
Length: 100, dtype: int64
```

Figure 3. Frequency of Words in Tags

Appendix D

$$\chi^2 = \frac{(O(W) - E(W))^2}{E(W)} + \frac{(O(X) - E(X))^2}{E(X)} + \frac{(O(Y) - E(Y))^2}{E(Y)} + \frac{(O(Z) - E(Z))^2}{E(Z)}$$

Figure 4. The Formula of Chi-Square

Appendix E

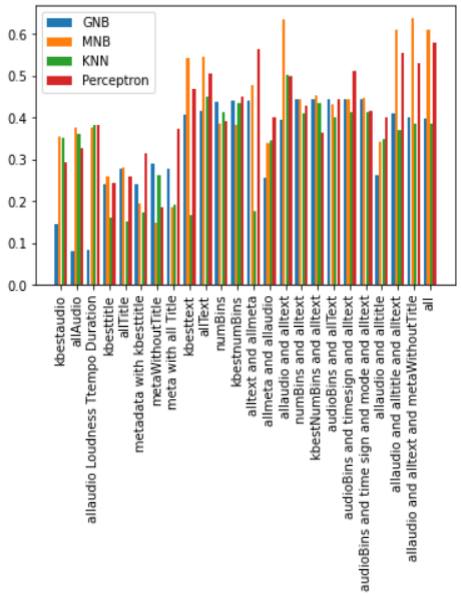

Figure 5. Accuracy on Attribute Sets

Appendix F

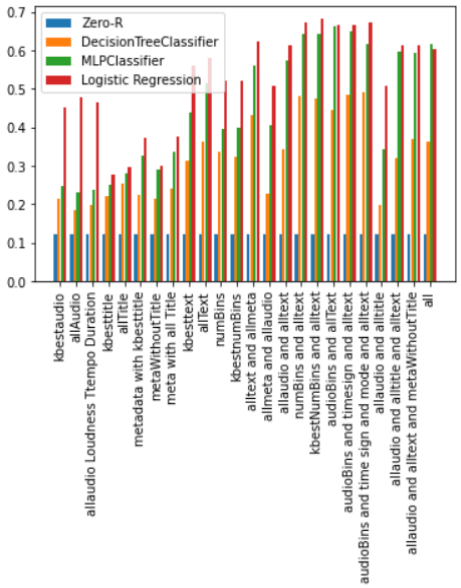

Figure 6. Accuracy on Attribute Sets

Appendix G

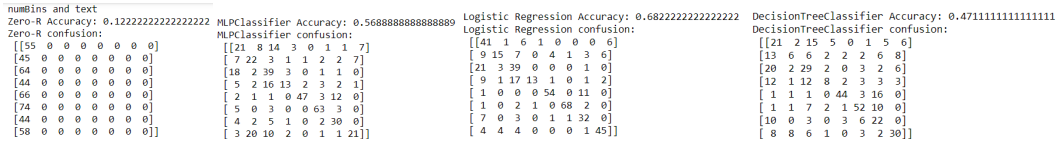

Figure 7. Confusion Matrices

Appendix H

	alpha	hidden_layer_sizes	Accuracy
0	0.001	(50, 25, 10)	0.627639
1	0.001	(50, 50, 50)	0.640791
2	0.002	(50, 25, 10)	0.582572
3	0.002	(50, 50, 50)	0.603412
4	0.003	(50, 25, 10)	0.585436
5	0.003	(50, 50, 50)	0.615395

Figure 8. The Table of Grid Search Results