Lab Report for CW2 DATA CLASSIFICATION

***Abstract*—This report should be considered as a continuation of CW1 report. Several models have been trained to classify samples from different programmes, including Decision Tree, Random Forest, Linear and Kernel Support Vector Machine, and Gaussian Naive Bayes. In the end, 'Programme3' is well distinguished while 'Programme2' is not.**

***Index Terms*—Data Preprocessing, Decision Tree, Gaussian Naive Bayes, Kernel SVM, Linear SVM, Random Forest, Support Vector Machine**

I. INTRODUCTION

THIS experiment aims to classify samples from different programmes based on the given dataset from INT104 final exam in 2023. Multiple classifiers have been applied, and the accuracy and confusion matrix are considered as performance evaluation.

II. CLASSIFY 'PROGRAMME 1&2&3&4'

Apply several methods to classify students from different programmes, and test model performance in various ways.

A. Data Pre-processing

Before training any model, the raw data should be pre-processed as follows:

• Remove 'Index' and 'Programme' from raw data as DataFrame X and let DataFrame Y contain a single column 'Programme'. The reason we remove 'Index' is that the indexed numbers do not contain any practical meaning, and might lead to overfitting, for instance, giving each index a branch, which does not represent any generalization performance.

• Separate data into 80% training set and 20% test set. Within the training set, K-fold cross validation can be performed to obtain the optimal parameter combination. Now the data is ready for Decision Tree and Random Forest, naming the data here 'data_dt' and 'data_rf'.

• Perform Z-score normalization on the data for Support Vector Machine (SVM) and Naive Bayes (NB), since the principles of training these two models. Another operation is to perform one-hot encoding on the 'Gender' column since it is represented by numbers '1' and '2', which do not have a natural order. So far the data is ready for SVM and NB, naming the data here 'data_svm' and 'data_nb'.

B. Decision Tree & Random Forest

Random Forest (RF) can be considered as an ensemble classifier of multiple Decision Tree (DT) models.

1) Decision Tree: To determine the optimal parameter combination for DT classifier, perform grid search and 5-fold cross-validation on 'data_dt'. The criterion for the evaluation is also chosen in this process between 'gini' and 'entropy'. With this parameter set, train a DT model with the training set and achieve an accuracy of 0.65 on the test set. Due to space constraints, the confusion matrix is not displayed here.

2) Random Forest: Before performing RF, the runtime should be considered as the computational complexity of RF is quite high. To avoid sacrificing the quality of parameter combinations, Bayesian optimization is introduced to leapfrog in search of the optimal parameter combinations, significantly reducing runtime. Besides, adjust the cross-validation into 10-fold. This RF model achieved an accuracy of 0.69 on the test set. The confusion matrix is shown in Fig. 1..

Fig. 1. Confusion Matrix for Random Forest

Another attempt is made to improve the model's generalization performance through post-pruning, but the accuracy slightly decreases to 0.66. This suggests that inappropriate pruning led to the loss of valuable information.

3) Extract Features: Various different feature sets have been extracted to train RF model, but there was slight decreases in accuracy. Since the operation of DT and RF inherently includes a feature selection process, removing some features from the training set fails to significantly improve the model's performance.

C. Support Vector Machine

Given the high-dimensional nature of this dataset, SVM shows great potential. The 'data_svm' is used to train SVM models in this section.

1) Linear SVM: In the search space, the regularization parameter 'C' was set to a large range to explore the most suitable parameter combinations. The result of Bayesian optimization selected C=0.04, indicating relatively weak penalty and generalization capability enhancement. The final model achieves an accuracy of 0.70, and the result can be represented in Fig. 2..

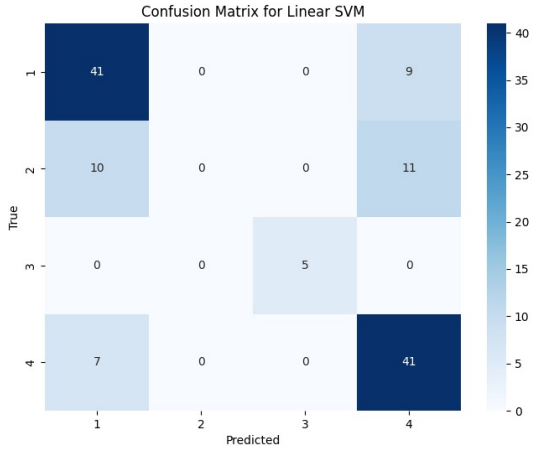

Fig. 2. Confusion Matrix for Linear SVM

2) Kernel SVM: As for the choice of kernel type, it is included along with the relevant parameters of different models, as part of the search space. Ultimately, Bayesian optimization selects the 'sigmoid' kernel along with a considerably large regularization parameter. The final accuracy decreases slightly to 0.69 compared to linear SVM model.

3) Extract Features: For feature selection, different SVM models should be considered with different criteria. For linear SVM, the linear correlation of different features with 'Programme' is calculated and the strongest feature set is chosen. For kernel SVM, it utilizes parameters that represent non-linear correlations, including Chi-square Test, Analysis of Variance, and Mutual Information. Specifically, even though the overall correlation between 'Grade' and 'Programme' is not strong, based on its particular association with 'Programme3' as stated in CW1, it should be retained. Several attempts have been performed on both linear and kernel SVM, the resulting accuracy has never reached up to the models trained with full feature set. This suggests that some features with relatively weak correlations still contain valuable information.

D. Naive Bayes

Naive Bayes (NB) is a mathematic algorithm based on probablity theory. The so-called "Naive" can be interpreted as the indifference to order, which can be neglected due to the relative independence between different features. Different types of NB should be considered based on their principles and the type of data being processed. The 'data_nb' is used for training NB models.

1) Polynomial & Bernoulli NB: Polynomial NB is suitable for frequency-type data, so the current dataset needs extensive transformation. Moreover, due to the wide and scattered range of scores within many features, and the possibility of some scores not appearing in the training set, Polynomial NB is not an appropriate choice. As for Bernoulli NB, the features of the dataset fails to meet the requirements for binary data. Thereby, Gaussian NB seems like a good choice.

2) Gaussian NB: Gaussian NB does not require any parameter, and thus skip the Bayesian optimization. The resulting model achieves an accuracy of 0.67 and the result is shown in Fig. 3.. The classification of 'Programme4' is evidently not as strong as previous models.

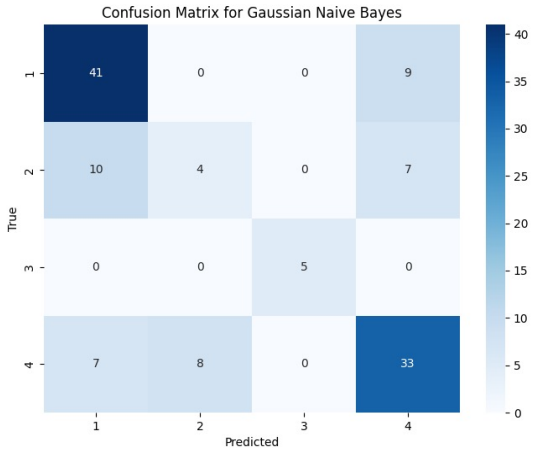

Fig. 3. Confusion Matrix for Gaussian Naive Bayes

3) Extract Features: Since Gaussian Naive Bayes assumes that features follow a Gaussian distribution, histograms of each feature have been plotted for data observation. Here, take feature 'MCQ' as an example, and the histogram below in Fig. 4. has shown evident yet not exact Gaussian distribution. Again, feature 'Grade' has been reserved for the same reason as stated in SVM feature extraction. Train several Gaussian NB models with different feature sets, and the best one has reached up to an accuracy of 0.65.

E. Ensemble Learning

Multiple models have been trained and the next attempt is to combine models into an ensemble learning model. The most common approach is to use the 'VotingClassifier' package. However, since different models have different data preprocessing methods and search space compositions, this approach is not suitable here. Instead, integrate the predictions from each model into a single DataFrame and determine the most frequently occurrence as the final prediction.

1) Ensemble Classifier Of 3 Best Models: The first attempt is to integrate 3 best models, one from each category of classifier, which are exactly the 3 confusion matrices shown above. The individual accuracies are 0.69, 0.70 and 0.67 respectively to Random Forest, Linear SVM and Gaussian Naive Bayes. The ensemble classifier achives an accuracy of

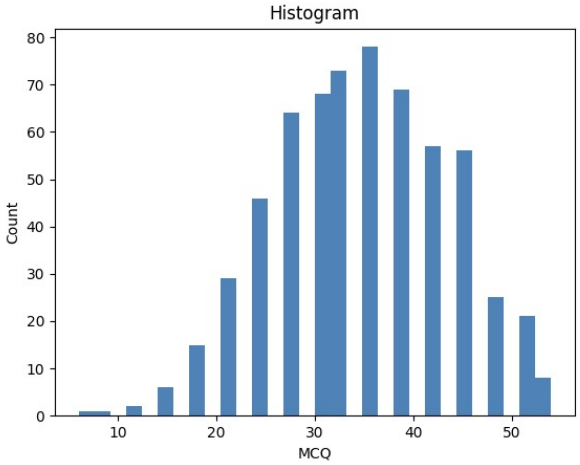

Fig. 4. Histogram of 'MCQ'

0.70, which is exactly the accuracy of Linear SVM model. The confusion matrix shown in Fig. 5. has demonstrates that the other 2 models offer mostly similar predictions compared to Fig. 2., and the ensemble classifier fails to enhance the overall performance.

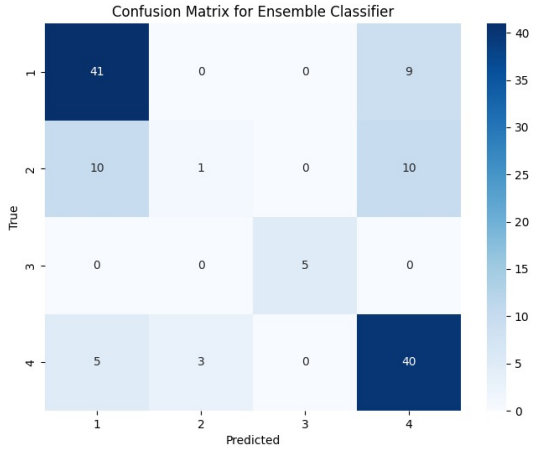

Fig. 5. Confusion Matrix for Ensemble Classifier

2) Ensemble Classifier Of More Models: Ensemble learning does not imply exclusive inclusion of high-performing models. Occasionally, models with lower aggregate accuracy may effectively rectify predictions for specific samples. Thereby, set the accuracy threshold at 0.65, and integrate all models with accuracy exceeding this threshold together. The ensemble learning model composed of eight individual models achieves a final accuracy of 0.69.

III. Classify 'Programme 1&2&4'

The several confusion matrices all perfectly classify students in 'Programme3'. Further classification on the rest of 'Programme' is done in this section. Before training any model, remove all the samples in 'Programme3' from the dataset, and perform similar data preprocessing for each kind of model.

A. Gaussian NB

Surprisingly, this Gaussian NB model has an accuracy of 0.21. From the confusion matrix shown in Fig. 6., it can be interpreted that this model indiscriminately predicts almost all samples as 'Programme2'. In the classification report, the recall for 'Programme2' is 0.96 while precision and f-1 score are 0.22 and 0.35 respectively. This indicates that the model exhibits extreme imbalance between precision and recall, and lacks generalization capability.

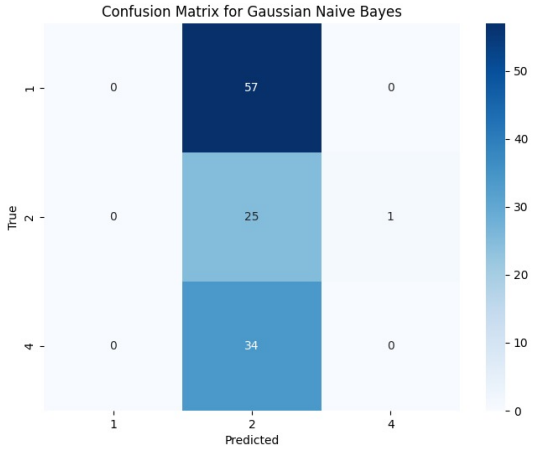

Fig. 6. Confusion Matrix for Gaussian Naive Bayes ('Programmes 1&2&4')

B. Other Classifier

4 more models have been trained by Random Forest, Random Forest with Post-pruning, Linear SVM and Kernel SVM. The accuracies are 0.57, 0.60, 0.58 and 0.62. The similarity between these models is that only a few samples from 'Programme2' have been successfully classified.

IV. Classify 'Programme 1&4'

Further remove samples in 'Programme2' from the dataset, and train an ensemble learning model of Random Forest and Linear SVM, which have great classification on 'Programme1&4'. This model has improved the accuracy to 0.74. However, this means all 'Programme2' will be classified into 'Programme1&4' if performing this model on samples of 'Programme1&2&4'.

V. Result

Overall, 'Programme3' has been perfectly classified, and the classification between 'Programme1' and 'Programme4' can be classified with sacrificing 'Programme2'.

VI. Recommendation

Due to the limited number of samples for 'Programme3', there still exists a risk in the model's ability to predict unknown samples accurately. More samples can be collected to further enhance the performance.