INT104 CW2 Lab Report

Mingxuan Hu 2252534
TA: Ziqian Yang

***Abstract*—This lab report presents the results of an experiment using supervised learning classifiers to classify students' programmes. With the assistance of Python, decision tree, random forest, support vector machine, Naïve Bayes, and an ensemble classifier were employed to analyze the dataset and evaluate their classification performances. Feature selection and model evaluation metrics were highlighted. The findings demonstrate the effectiveness of ensemble classifiers in achieving improved classification performance. Additionally, the potential for further optimization using grid search was suggested.**

***Index Terms*—Supervised Learning, Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, Stacking, Grid Search**

I. Classifiers in Machine Learning

In machine learning, a classifier is an algorithm or model that is a part of supervised learning. It learns from known data with pre-selected features for training, and classifies unlabeled data based on these features.

A good classifier is usually evaluated based on several key metrics, including accuracy, precision, recall, and F1 score. Accuracy measures the proportion of correct predictions over all predictions. Precision refers to the ratio of true positive predictions to the total number of positive predictions. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that are correctly identified. And the F1 score is the harmonic mean of precision and recall, which provides a balanced measure for models with imbalanced class distribution.

The purpose of this experiment is to use different supervised learning classifiers to classify students' programmes, try to use different features to optimize the classification performance, and comprehensively compare the classification effects of multiple classifiers.

II. Decision Tree

The first classifier uses the decision tree method.

Decision tree is a model that displays decision rules and classification results with a tree-like data structure. As an inductive learning algorithm, its focus is to transform seemingly disordered and chaotic known data into a tree-like model that can predict unknown data through some technical means.

In order to find the attributes that contribute more to the final classification result of the decision tree, it is necessary to evaluate the information carried by the features. Entropy, information gain, and Gini impurity are several commonly used metrics for selecting features.

Entropy measures the degree of randomness in data, where a higher value means more disordered data. Information gain indicates how much entropy can be reduced by splitting on a feature, with a higher value suggesting a more informative feature. Gini impurity measures the probability of misclassifying a randomly chosen sample from the dataset if it were randomly labeled according to the class distribution, which can be used as an alternative to entropy for selecting attributes.

Based on the above knowledge, Figure 1 shows the entropy, information gain, and Gini impurity of each feature in the given data set. The figure illustrates that the Total, Grade, MCQ and Q2 columns have a higher information gain, meaning that classification using these features may be better. However, it is noted that the entropy and Gini impurity of the Total column are relatively high, indicating that the data for this feature may not be pure enough, and the decision tree may need more child nodes for classification using this feature. Moreover, the Gender and Q1 columns are also useful features for classification, given their low entropy and Gini impurity, if problems such as computational complexity and overfitting are taken into account.

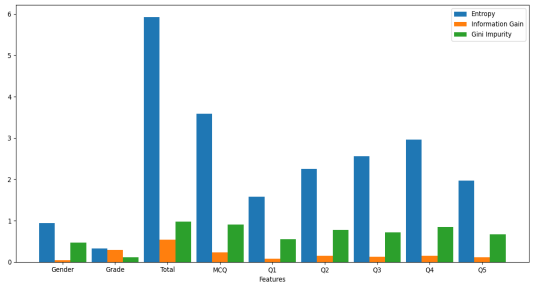

Fig. 1. Entropy, Information Gain and Gini Impurity of Each Feature

In addition, Figure 2 shows the similarity between entropy and Gini impurity. When the entropy and triple Gini impurity of the nine features in the data set are plotted as a line plot, it is found that except for the Total column, the two values corresponding to the other eight features are very close, which indicates that entropy and Gini impurity give similar results in practice.

The above analysis is helpful for feature selection when constructing decision tree. According to the analysis, the six features Total, Grade, MCQ, Q2, Gender and Q1 will be used to build the decision tree classifier. The classification results of the classifier can be represented by a four-by-four confusion matrix. The rows of this matrix represent the real categories, and the columns represent the predicted categories. Each cell represents the number of cases in which the true category intersects with the predicted category. The confusion matrix of the constructed decision tree classifier can be visualized

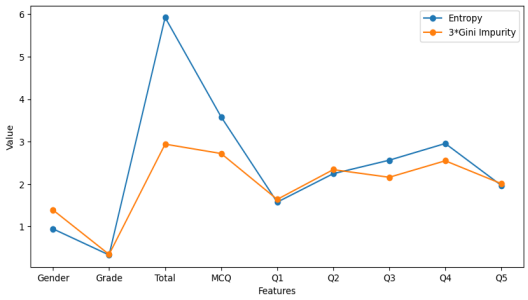

Fig. 2. Entropy vs. 3*Gini Impurity

using heat maps, and the results are shown in Figure 3.

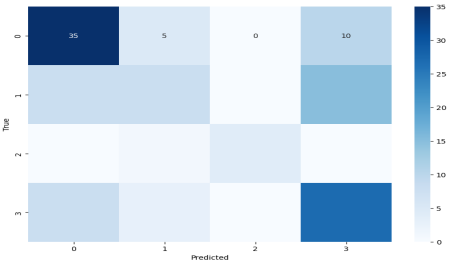

Fig. 3. Confusion Matrix of Decision Tree

However, due to the randomness in the selection of training and testing samples (80% and 20% respectively) in the data division, any single classification by the classifier is accidental. Therefore, the same data set can be divided into different training sets and test sets for multiple classification, and the average F1 score of multiple classification can be calculated to judge the performance of the classifier. Figure 4 shows the F1 score of each classification after 10 decision tree classifications of the data set. Through calculation, the average F1 score value of 10 classifications is 0.57.

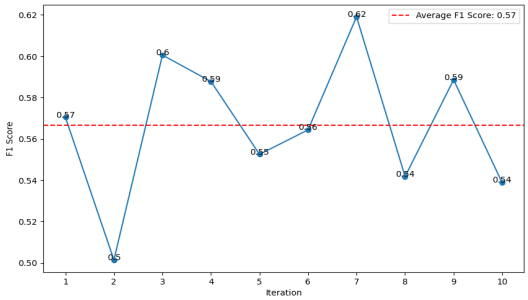

Fig. 4. Average F1 Score of Decision Tree over 10 Iterations

III. Random Forest

Obviously, the classification effect of decision tree is not up to the ideal state. Therefore, a more powerful machine learning model can be used - random forests.

Random forest is an ensemble learning method that integrates multiple decision trees to improve classification. The basic principle of a random forest is to create multiple decision trees and vote on their predictions. Specifically, each tree is trained on a portion of randomly selected training data as well as randomly selected features. When making predictions, the random forest lets each tree make predictions independently, and then selects the category with the most votes as the final prediction result.

Compared with a single decision tree, the random forest model can not only deal with the complex situation of high dimension and large amount of data effectively, but also has good robustness and generalization ability, so it can be used as our second classifier.

Since there is no need to select features in advance, the entire dataset (excluding the Index column and selecting the Programme column as the target column) can be randomly forested directly. Figure 5 shows F1 scores for 10 random forest classifications of the dataset, compared to the decision tree.

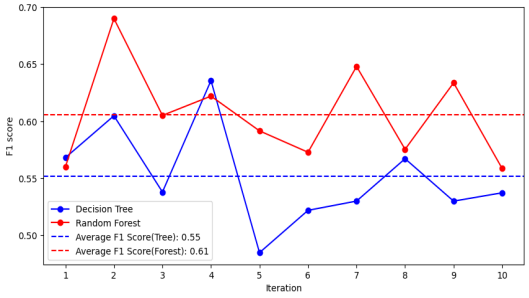

Fig. 5. F1 Scores for Decision Tree and Random Forest

As can be seen from the figure, F1 scores for random forests are generally more stable and have a higher average compared to decision trees. This shows that random forest has higher average performance and more stable prediction ability over the course of multiple iterations.

IV. Support Vector Machine

The third classifier is the SVM classifier.

SVM, which stands for support vector machine, is a supervised learning algorithm. Its core idea is to find an optimal decision boundary (or hyperplane) that effectively separates different classes of data points.

In two-dimensional space, the decision boundary is usually a line that separates two categories of data points. This line is called the decision boundary of a linear SVM. However, when the data cannot be separated by a single line, a non-linear SVM is needed. By introducing kernel function, non-linear SVM maps data from the original feature space to a higher-dimensional space, making the data linearly separable in this higher-dimensional space. In this way, a linear hyperplane can be found in the high-dimensional space to separate the data, thus achieving nonlinear classification.

However, SVM classification may present some challenges when faced with high-dimensional data, such as the 11-dimensional data set that needs to be processed in this ex-

periment. As the data dimension increases, the complexity of the data space increases exponentially. In this case, the distance between the data points becomes sparse, resulting in the difficulty of finding a clear, representative decision boundary in the high-dimensional space. In addition, the computational complexity of SVM increases with the increase of data dimensions. Feature selection and processing in high-dimensional data also becomes more difficult.

Therefore, before SVM classification of this dataset, the dataset was first normalized, then dimensionally reduced by PCA (Principal Component Analysis), and the dimensionally reduced data was mapped to a two-dimensional plane. In terms of feature selection, features that contribute the most to the first two principal components newly generated after PCA are selected, including Total, MCQ, Q2 and Q4 (as mentioned in the previous CW1 Report).

After verification, the average F1 score of the 10 classifications of the SVM classifier is about 0.55. In addition, the distribution of data after PCA and the decision boundary found by SVM algorithm can also be visually observed on a two-dimensional plane, as shown in Figure 6.

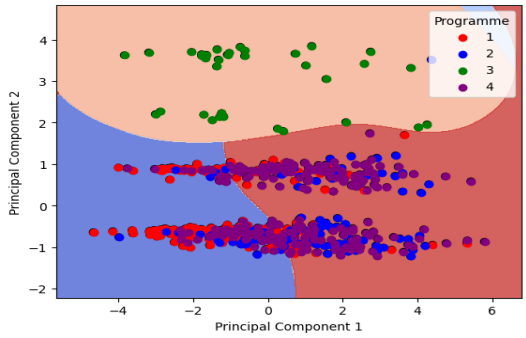

Fig. 6. PCA with SVM Decision Boundary

V. Naïve Bayes

The fourth classifier uses a Naïve Bayes method.

Naïve Bayes method is a classification algorithm based on Bayes' Rule. Its core idea is to estimate the posterior probability of the samples to be classified to belong to each category through Bayes' Rule, and then select the category with the highest posterior probability as the prediction result. Bayes' Rule describes how to update our understanding of the probability of an event based on new observational data, given the known prior probability. The formula of Bayes' Rule is as follows:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

In Naïve Bayes, the assumption is that the features are independent of each other, which makes it simple to calculate a posterior probability by multiplying the conditional probability of each feature and then multiplying it by the prior probability. In this way, the Naïve Bayes method enables fast and efficient classification, especially in the case of small data sets.

To sum up, since all features are independent of each other in Naïve Bayes, feature selection is usually not performed in Bayesian classifiers, but all available features are used. Through repeated verification, the average F1 score of the Bayesian classifier can reach 0.58, which indicates a relatively satisfactory classification effect.

VI. Ensemble Classifier

Based on all the above findings and the four classifiers that have been constructed, the final ensemble classifier can be constructed with the idea of stacking.

Stacking is an ensemble learning technology. The core idea of stacking is to input the prediction results of multiple base classifiers into a meta-model as new features to obtain more accurate final prediction results. This approach allows the strengths of different classifiers to complement each other, thereby improving overall prediction performance. Through the multi-layer architecture, stacking captures the observations and assumptions of different classifiers, improving the generalization ability of the model.

Therefore, the classification effect of the ensemble classifier should be better than all the previous classifiers. The facts are as shown in Figure 7. By randomly classifying 10 times, Figure 7 shows the F1 score of all 5 classifiers constructed in this experiment each time and the average F1 score of each classifier. It can be found that the ensemble classifier has the best score among all classifiers.

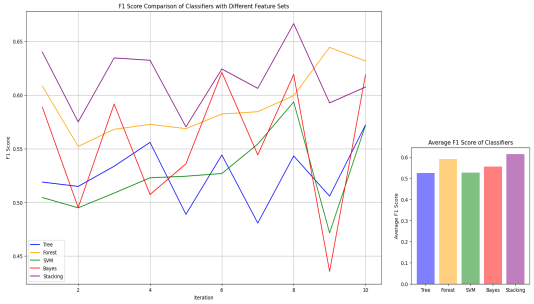

Fig. 7. Classifier Performance Analysis

In addition, the receiver operating characteristic (ROC) curve can also be used to analyze the classification effect of the classifier. The ROC curve illustrates the relationship between the model's true positive rate (TPR) and false positive rate (FPR) under different threshold Settings, which summarizes all of the confusion matrices that each threshold produced. Due to spatial constraints, no further demonstration will be provided here.

It should also be noted that in order to optimize the classifier, grid search can be used. Grid search is a method used in machine learning to systematically traverse multiple combinations of model parameters to determine the best model. By cross-validating within a predefined parameter space, grid search can help determine the optimal combination of parameters to optimize the performance and generalization of the model. This is where the experiment can be improved.