CW2 report

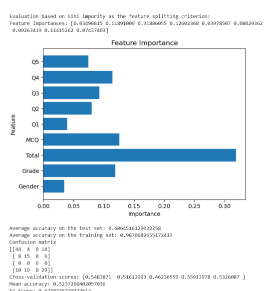
Enze Zhou, Student ID:2254411
TA:Zhejun.Yang

I. Decision tree and decision forest

After the data and packages were imported, I first removed the index and Programme columns, divided the data set with a ratio of 3:1 and normalized them. I predict that random forests will work better than decision trees.First I decide what algorithm to use for the decision tree model,After comparison, I used CART algorithm to train the decision tree model.As shown in figure [1].

To evaluate my model, I define the model evaluation function, which includes accuracy, confusion matrix, cross-validation, and F1 score. The figure [2] presents that i also included feature importance in the evaluation of the decision tree model so that I could observe the contribution of the features.

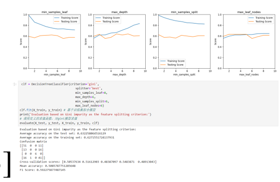
[1] 

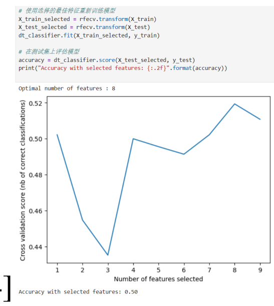
[2] 

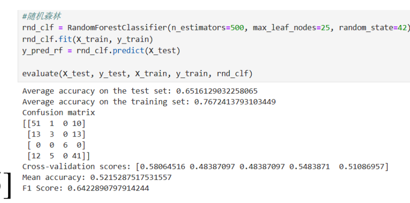
In addition, in order to make the decision tree more accurate, the scores of the decision tree model under different parameters are visualized, and the best combination is selected to retrain the model for comparisonAs illustrated by the figure[3].

Then, I use recursive feature elimination, which can improve the model performance and reduce the risk of overfitting, so as to obtain the relationship between the best feature and the target variable and the feature importance ranking

Finally, since the previous effects were not satisfactory, I used random forest and tested various feature combinations, which partially improved the effect.From the figure[5] that random forest is more suitable for the analysis of this data.Just as I predicted,This may be because it reduce overfitting and improve generalization.

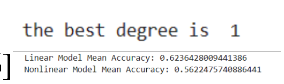
[3] 

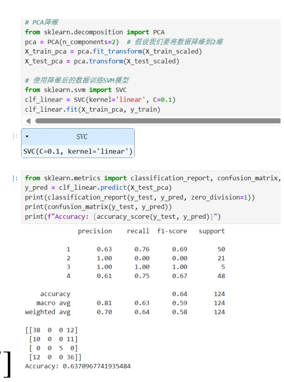
[4] 

[5] 

II. SVM

Since there were four types of programmes to be analyzed, I first used nonlinear SVM to check the optimal order of polynomial kernel, and found that the result was actually 1, as shown in the figure[6]. Therefore, I trained the linear SVM model I adjusted the c parameter and adjusted the size of the polynomial kernel, i also compared it with the nonlinear SVM model,I predicted that the nonlinear SVM would be better than the linear SVM, but surprisingly, the linear SVM was clearly better. I found that the linear SVM model was better, so I adopted the linear SVM model.

[6] 

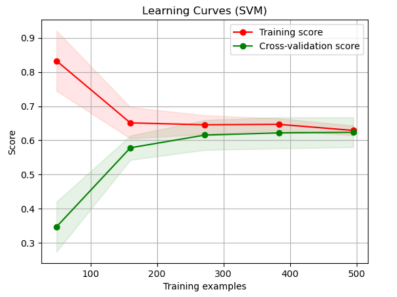
[7] 

Then, in order to help analyze the performance

and generalization ability of the model, As shown in figure [8],I drew a learning curve to evaluate the model, and found that my linear SVM model fit well, and the size of the training set was also just right. The main problem should be the raw data and the selection of my feature combination.

```
# 使用上面定义的函数绘制学习曲线
title = "Learning Curves (SVM)"
plot_learning_curve(clf_linear, title, X, y, cv=5, n_jobs=-1)

plt.show()
```

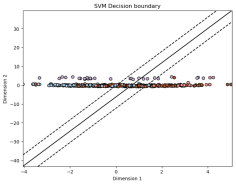


[8]

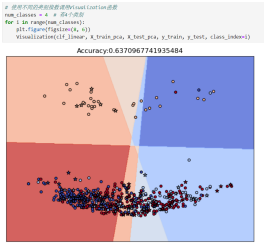
Then, in order to visualize the SVM decision boundary, I first conducted PCA dimensionality reduction on the original data like figure[9], and evaluated the model. When I found that the effect was good, I began to visualize it. However, as can be seen from the figure[10], the visualization results were not satisfactory.

So I visualized the decision boundaries of different class indices, as shown in the figure, where the star is the test sample, the circle is the training sample, and the color of the background shows the classifier's prediction results for each position. However, the effect is still poor, which may be due to inappropriate feature selection or overfitting problems.

[9]



[10]



III. Naïve Bayes

For naive Bayes,I first made a model of Gaussian naive Bayes and a model of polynomial naive Bayes, and compared and tried to combine various features after many comparisons, I chose Gaussian naive Bayes model, and conducted training and evaluation.Originally, I predicted that the feature combination would be the same as the best feature combination obtained after PCA and tsne, but unexpectedly, the result was completely different. I tried multiple feature combinations and found that 'Q3', 'Q4 ', 'MCQ' did not provide a positive contribution, especially 'Q3' and 'Q4 ', as shown in the figure[11],[12].

According to the figure[13], in order to more clearly explain how these features affect the classification process, I obtained the weight of each feature and made a comparison.Still got the same result.

[11]

[12]

```
# 模型评估
y_pred = gnb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

[13]

```
Gender: 1.7542857142857142
Grade: 2.0
Total: 73.16857142857143
MCQ: 39.34285714285714
Q1: 7.291428571428572
Q2: 5.04
Q3: 12.274285714285714
Q4: 7.5685714285714285
Q5: 1.6628571428571428
Class: 1, Probability: 0.40415704387990764
Class: 2, Probability: 0.18244803695150116
Class: 3, Probability: 0.057736720554272515
Class: 4, Probability: 0.3556581986143187
Accuracy: 0.6129032258064516
```

IV. Ensemble classifier

In the final integrated classification, I first trained and evaluated this model by Bagging method, and

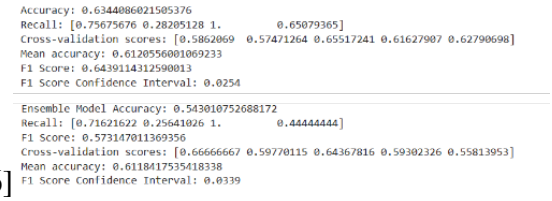
then I used Boosting methodIt's not in the picture.Finally, I also use model fusion technology to combine the prediction results of multiple basic models to improve the overall prediction performance, hoping to get more accurate and robust prediction results by combining different types of models and making use of their advantages.As these figure[14],[15] shows.

[14]

[15]

However,After training and evaluation, I found that Boosting model and model fusion technology has very poor effect, so I used the soft vote in voting classifier to train and evaluate again, and compared multiple feature combinations, as shown in the figure[16] below(The top is the Bagging assessment and the bottom is the Voting assessment). Again, the Bagging method and the combination of features ['Gender', 'Grade', 'Total', 'MCQ', 'Q1', 'Q2', 'Q5'] were found to be superior.

[16]



V. Conclusion

In this experiment, I deeply studied the application of various machine learning algorithms such as decision trees, support vector machines, naive Bayes and integrated classifiers in data analysis and classification tasks.

First, I used the decision tree model and chose the CART algorithm by comparison. I then evaluated model performance, including accuracy, confusion matrix, cross-validation, and F1 scores, and visualized the importance of features. In order to further improve the accuracy of the decision tree, I tried different parameter combinations and adopted recursive feature elimination method to optimize the feature selection.

However, since the decision tree model did not work well, I tried the random forest model instead and tested various combinations of features. The results show that the random forest model is a better fit for this dataset, which is in line with my expectations.

Next, I turned to support vector machine models and found that linear SVM models worked better than nonlinear SVM models. Through the learning curve and decision boundary visualization, I further analyzed the model's performance and generalization ability.

In the naive Bayes model, I ran several experiments and found that certain features contributed poorly to the model's performance, which was not what I expected, but still got 'Q3', and the feature combination with 'Q4' removed was the best.

Finally, I tried an integrated classifier and found that the Bagging method worked best with a combination of specific features.