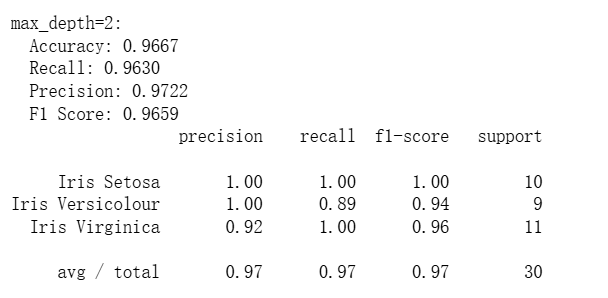
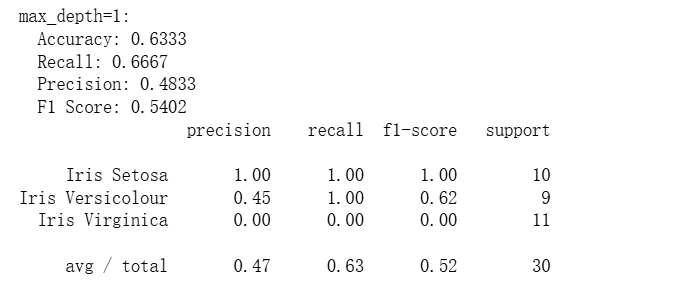
IIT number: A20563467

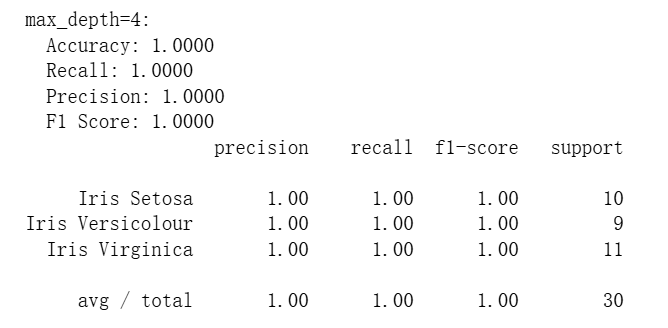
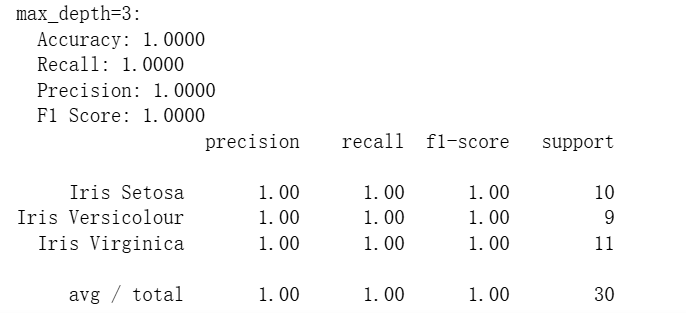
Name: Zongkun Qin

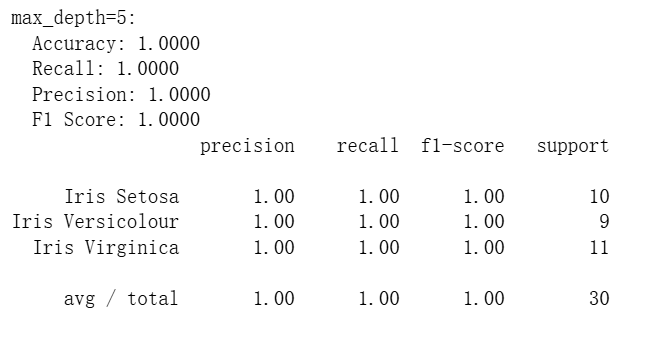
1. Problem 1

Load the Iris sample dataset from sklearn (using load\_iris()) into Python with a Pandas DataFrame. Induce a set of binary decision trees with a minimum of 2 instances in the leaves (min\_samples\_leaf=2), no splits of subsets below 5 (min\_samples\_split=5), and a maximum tree depth ranging from 1 to 5 (max\_depth=1 to 5). You can leave other parameters at their default values. Which depth values result in the highest Recall? Why? Which value resulted in the lowest Precision? Why? Which value results in the best F1 score? Also, explain the difference between the micro, macro, and weighted methods of score calculation.





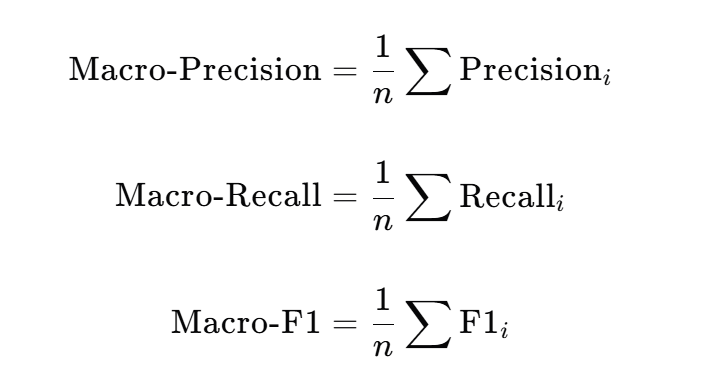
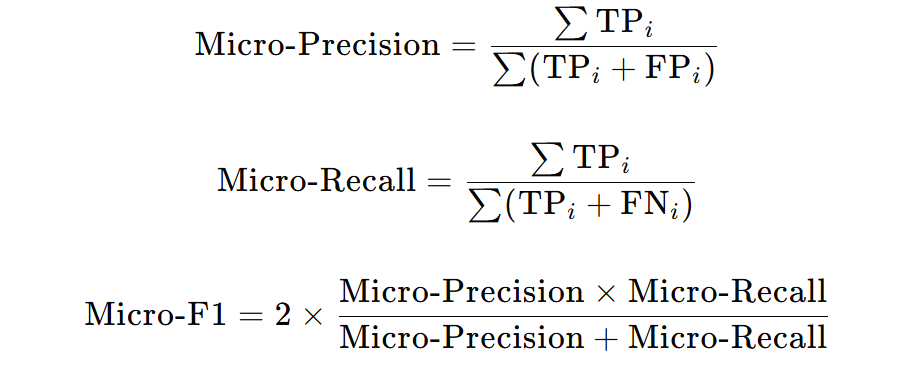


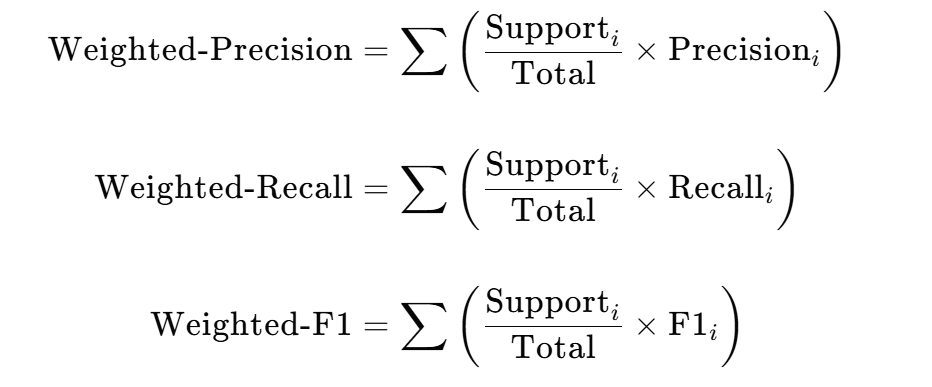


When max\_depth=1, the model is too simple to capture the complex patterns in the data, so its performance is poor. At max\_depth=2, the model becomes more complex and its performance improves significantly.When max\_depth=3, the model can fit the data better, achieving the best performance.

Deeper trees (like max\_depth=4 and 5) might cause overfitting, where the model works well on the training data but doesn't improve or even gets worse on the test data. However, on this dataset, the performance doesn't drop at max\_depth=4 and 5, indicating the data is relatively simple and these deeper models don't overfit. Deeper trees need more computational resources and time for training and prediction.

Although the performance is similar for max\_depth=3, 4, and 5, the model with max\_depth=3 is more computationally efficient while maintaining high performance.





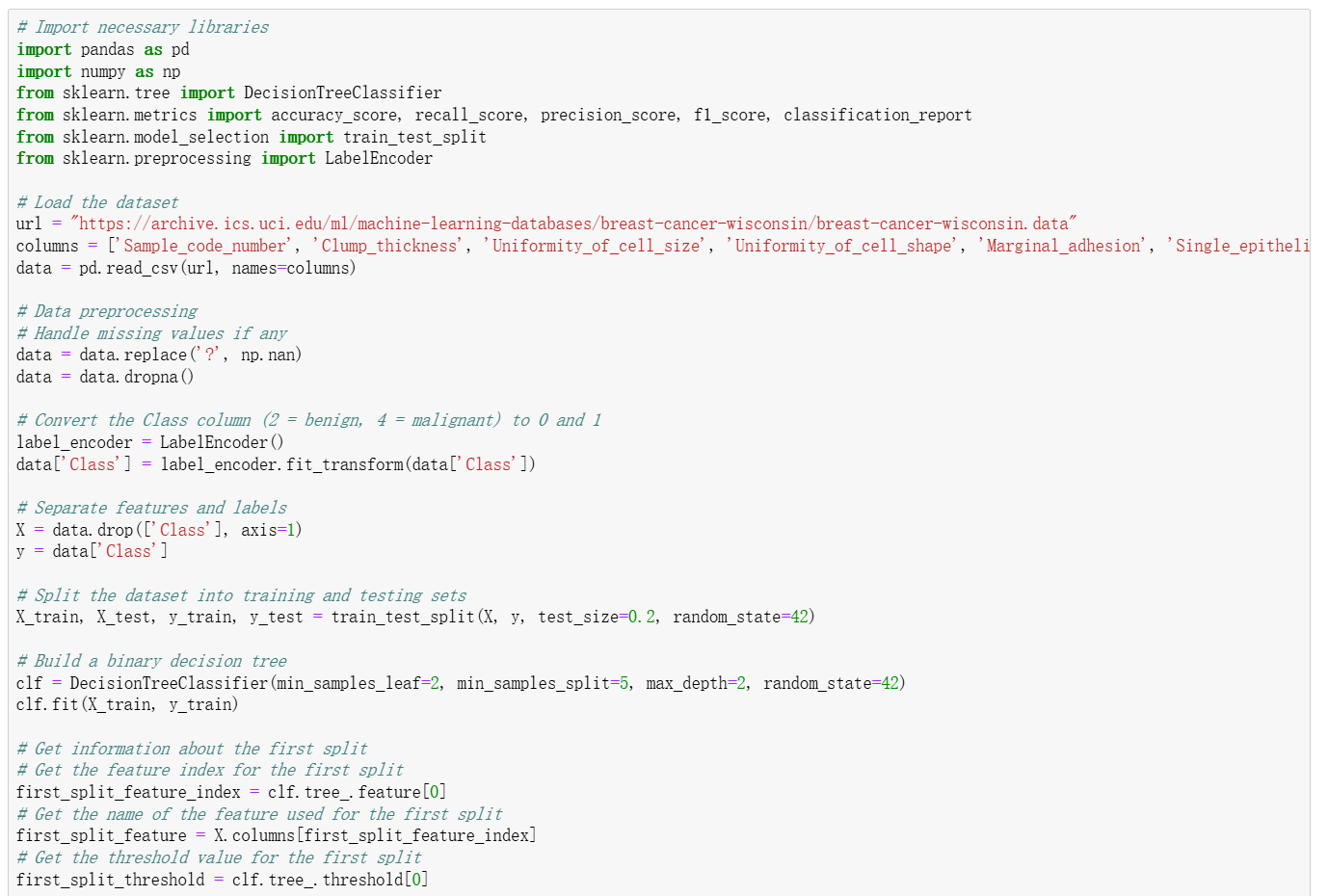
Micro-averaging sums all true positives (TP), false positives (FP), and false negatives (FN) across classes, then calculates precision, recall, and the F1 score. It focuses on overall prediction performance rather than individual class performance when class distribution is imbalanced, reflecting the model's overall data performance.

Macro-averaging calculates precision, recall, and the F1 score for each class separately and then averages these values. It fairly assesses each class's performance when classes are balanced. However, it may yield inaccurate evaluations with imbalanced classes since it doesn't account for sample size differences.

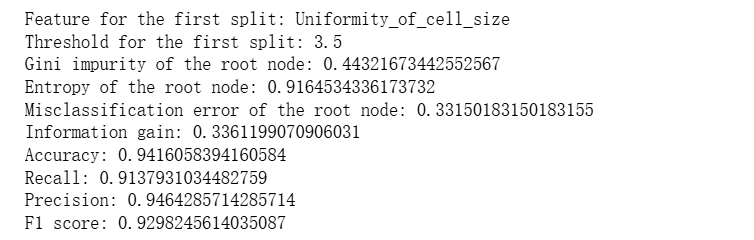
Weighted-averaging calculates precision, recall, and the F1 score for each class separately and then weights and sums these values based on the proportion of samples in each class relative to the total. It considers both class performance and sample size with imbalanced classes, providing a more comprehensive reflection of the model's overall performance.

2.Problem 2

Load the Breast Cancer Wisconsin (Diagnostic) sample dataset from the UCI Machine Learning Repository (the discrete version at breast-cancer-wisconsin.data) into Python using a Pandas DataFrame. Induce a binary Decision Tree with a minimum of 2 instances in the leaves, no splits of subsets below 5, and a maximum tree depth of 2 (using the default Gini criterion). Calculate the Entropy, Gini, and Misclassification Error of the first split. What is the Information Gain? Which feature is selected for the first split, and what value determines the decision boundary?



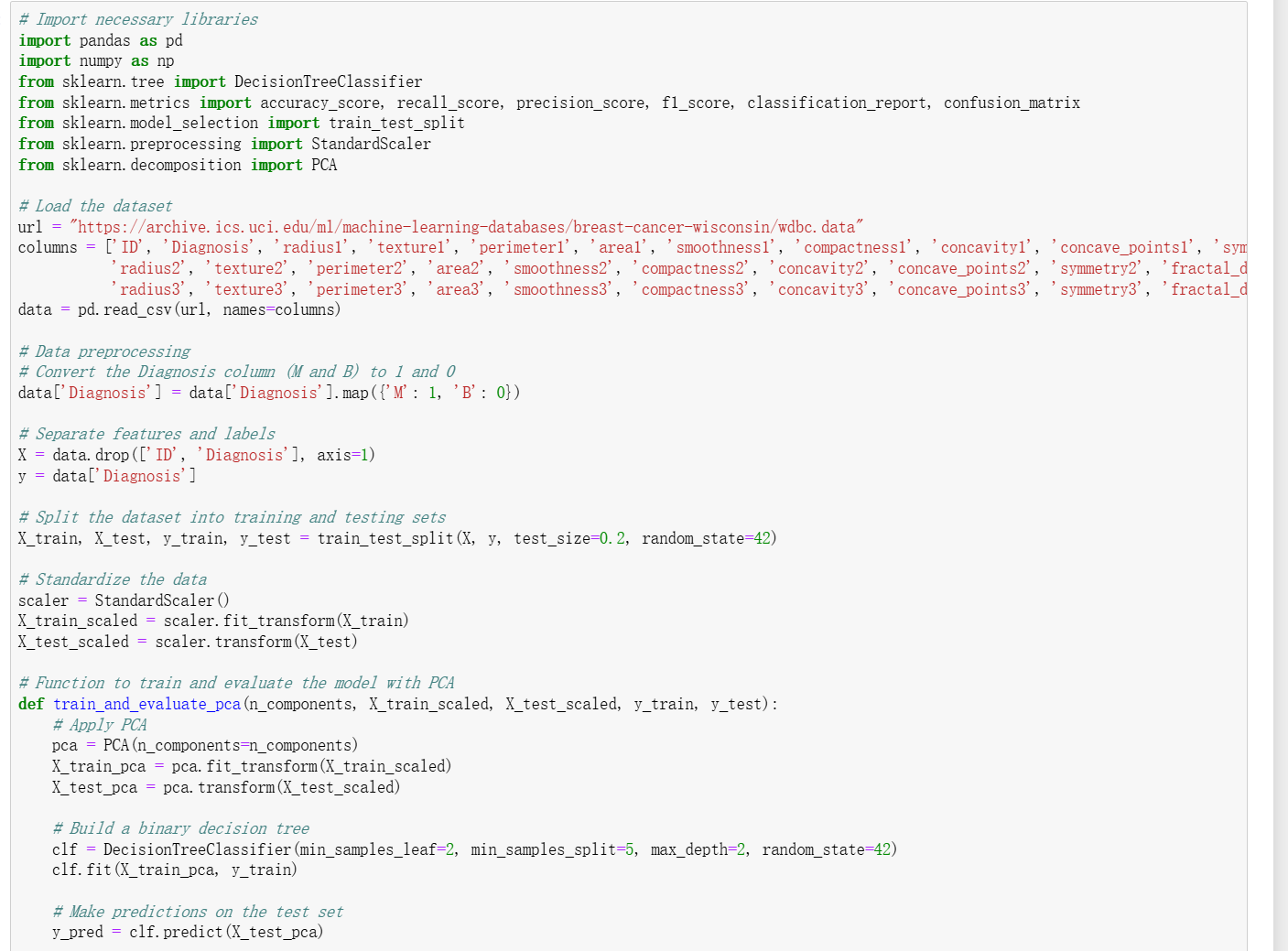


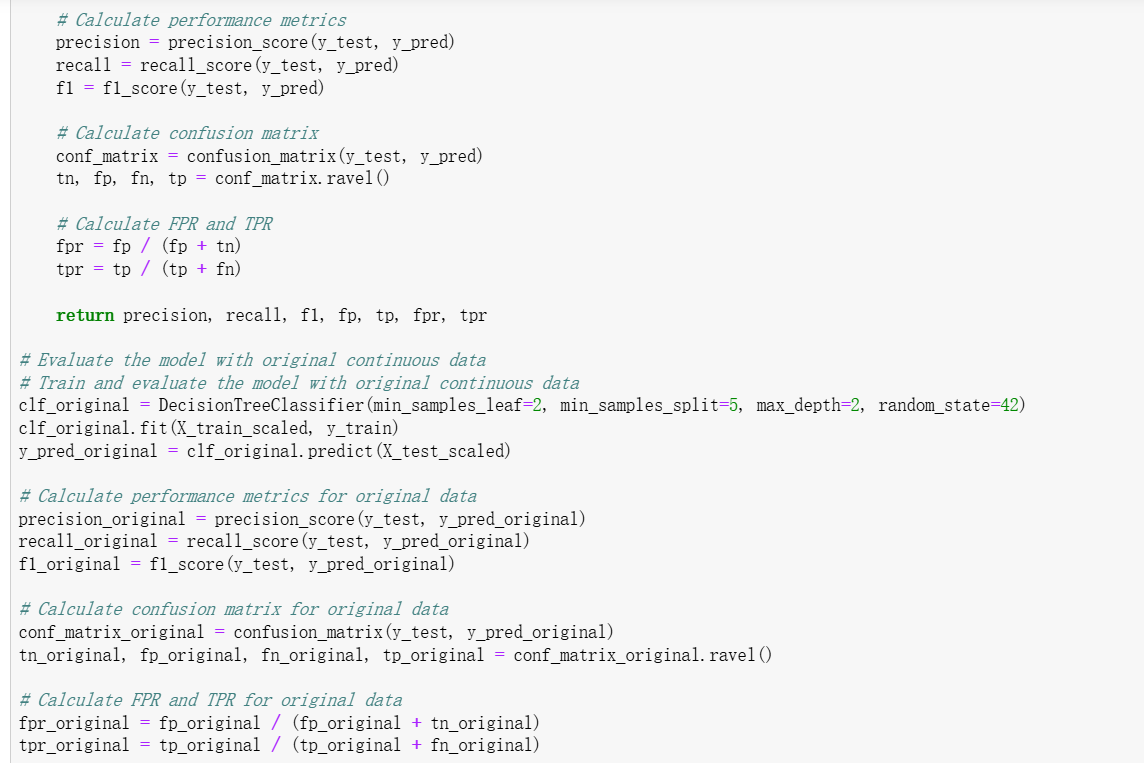


The variable first\_split\_feature stores the feature chosen for the first split—"Uniformity\_of\_size”, and the variable first\_split\_threshold stores the threshold value used for the first split—“3.5”.

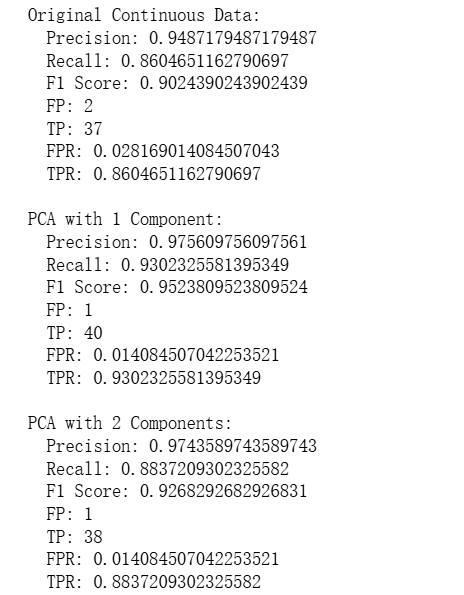
3.Problem 3

Load the Breast Cancer Wisconsin (Diagnostic) sample dataset from the UCI Machine Learning Repository (the continuous version at wdbc.data) into Python using a Pandas DataFrame. Induce the same binary Decision Tree as above (now using the continuous data), but perform PCA dimensionality reduction beforehand. Using only the first principal component of the data for model fitting, what are the F1 score, Precision, and Recall of the PCA-based single factor model compared to the original (continuous) data? Repeat the process using the first and second principal components. Using the Confusion Matrix, what are the values for False Positives (FP) and True Positives (TP), as well as the False Positive Rate (FPR) and True Positive Rate (TPR)? Is using continuous data beneficial for the model in this case? How?









Although F1 score of the continuous data is lower, but it’s more accuracy in precision, especially after the PCA operations. In this dataset, using PCA to reduce the data dimensions, especially to just 1 component, makes the model perform better in most ways compared to using the original continuous data. The precision and F1 score become higher after PCA. The false positive rate drops a lot, which means the model is better at spotting negative cases. The true positive rate also goes up noticeably when using 1 PCA component. So, in this case, using PCA - reduced data is way more helpful for the model. It keeps the model's performance, but with fewer dimensions, making computations faster and more efficient.