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

By traversing different maximum depths of decision trees, multiple decision tree models are constructed, and then each model is evaluated using different evaluation metrics (recall, accuracy, and F1 score) and different averaging methods (micro averaging, macro averaging, and weighted averaging). Finally, the decision tree maximum depth value that performs best or worst under different evaluation metrics and averaging methods is identified.

```
The depth value that results in the highest recall : 3
The depth value that results in the lowest precision : 1
The depth value that results in the best F1 score : 3
```

$$Recall(r) = \frac{TP}{TP + FN} \tag{1}$$

Recall rate: The proportion of samples that are actually positive cases that are correctly predicted as positive cases.

A depth value of 3 leads to the highest recall rate: When the depth is 3, **the model complexity is moderate**, which can effectively learn the feature patterns of positive samples in the data while **avoiding overfitting**. This enables the model to correctly identify more samples that are actually positive examples during prediction.

$$Precision(p) = \frac{TP}{TP + FP} \tag{2}$$

Accuracy: The probability of the model predicting the correctness of a positive sample.

Depth value 1 leads to the lowest accuracy: **the model is too simple** and the learning of data features is insufficient. The model is prone to incorrectly predicting negative cases as positive cases (**increasing false positive cases FP**)

$$F - measure(F) = \frac{2rp}{r+p} \tag{3}$$

The F1 score is the harmonic mean of recall and accuracy, used to balance the importance of both.

The depth value of 3 leads to the optimal F1 score: This balances recall and accuracy and makes the F1 score optimal.

$$Micro - Recall = \frac{\sum TP_i}{\sum (TP_i + FN_i)} \tag{4}$$

Micro First, calculate the sum of true cases (TP), false positive cases (FP), and false negative cases (FN) for all categories, and then calculate the recall rate, accuracy, and F1 score based on these sums. This method focuses more on overall performance and is more sensitive to categories with a large sample size.

$$Macro - Recall = \frac{1}{C} \sum_{i=1}^{C} Recall_i \tag{5}$$

Macro Calculate the recall rate, accuracy, and F1 score for each category separately, and then take the average of these scores. This method treats each category equally, without considering differences in sample size between categories.

$$Weighed - Recall = \frac{\sum_{i=1}^{C} N_i \times Recall_i}{\sum_{i=1}^{C} N_i} \tag{6}$$

Weighted Calculate the recall rate, accuracy, and F1 score for each category separately, and then weight average these scores based on the sample size of each category. This method considers the difference in sample size between categories

and assigns higher weights to categories with larger sample sizes.

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Problem 2

Load the Breast Cancer Wisconsin (Diagnostic) sample dataset from the UCI Machine Learning Repository (the discrete version at: breast-cancerwisconsin.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?

```
Entropy of the left subset after splitting: 0.1879
Entropy of the right subset after splitting: 0.5931
Gini index of the left subset after splitting: 0.0558
Gini index of the right subset after splitting: 0.2457
Misclassification error of the left subset after splitting: 0.0287
Misclassification error of the right subset after splitting: 0.1434
Information Gain: 0.5889

 best feature : Uniformity of Cell Size, threshold: 2
```

$$Gini = 1 - \sum_{i=0}^{c-1} p_i(t)^2 \tag{7}$$

The Gini coefficient for the left subset is 0.0558, and for the right subset it is 0.2457. The Gini coefficient measures data impurity, and the closer the value is to 0, the higher the data purity

$$Entropy = -\sum_{i=0}^{c-1} p_i(t) log_2 p_i(t) \tag{8}$$

The entropy of the left subset is 0.1879, and the entropy of the right subset is 0.5931. Entropy measures the degree of chaos in data, with lower values indicating higher data purity (more concentrated categories)

$$Classificationerror = 1 - max[p_i(t)] \tag{9}$$

The misclassification error of the left subset is 0.0287, and the misclassification error of the right subset is 0.1434. This value represents the error probability when predicting with the majority class, and the lower the value, the better the classification performance

$$Gain_{split} = Entropy(p) - \sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \tag{10}$$

The information gain is 0.5889.This reflects the degree of improvement in data purity after splitting, with higher values indicating more effective splitting.

Features selected for the first segmentation: Uniformity of Cell Size

Decision boundary threshold: 2, eigenvalue ≤ 2 is the left subset, eigenvalue>2 is the right subset.The eigenvalues and corresponding splitting thresholds jointly determine the decision boundary.

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? 1.the F1 score, Precision, and Recall of the PCA-based single factor model are 0.9286,0.8966,0.9630. the F1 score, Precision, and Recall of the model using original data are 0.8889,0.8889,0.8889.

2.When using the first and second principal components ,the F1 score, Precision, and Recall of are 0.9286,0.8966,0.9630.

3.When using the Confusion Matrix

```
Original Data:
F1 Score: 0.8889
Precision: 0.8889
Recall: 0.8889
TP: 48.0000
FP: 6.0000
TPR: 0.8889
FPR: 0.0674

Using 1st Principal Component:
F1 Score: 0.9286
Precision: 0.8966
Recall: 0.9630
TP: 52.0000
FP: 6.0000
TPR: 0.9630
FPR: 0.0674

Using 1st and 2nd Principal Components:
F1 Score: 0.9286
Precision: 0.8966
Recall: 0.9630
TP: 52.0000
FP: 6.0000
TPR: 0.9630
FPR: 0.0674

Using PCA-transformed data might be beneficial as it has a higher combined score of F1,
Precision, and Recall.
```

Table 1: Original Data Confusion Matrix

ACTUALCLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	48	6
Class=No	6	83

Table 2: Using 1st Principal Component Confusion Matrix

ACTUALCLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	52	2
Class=No	6	83

```
Confusion Matrices:
Original Data:
[[83  6]
 [ 6 48]]
Using 1st Principal Component:
[[83  6]
 [ 2 52]]
Using 1st and 2nd Principal Components:
[[83  6]
 [ 2 52]]
```

4.From the improvement in F1 score and the decrease in FP and FPR, it can be seen that using continuous data is beneficial for the model in this case.

Continuous data allows the model to dynamically select splitting thresholds, which will determine more accurate decision boundaries. To avoid loss of details caused by discretization and preserve the complete distribution of the original data.

PCA is more effective in dimensionality reduction on continuous data, and principal components can better preserve discriminative information.

With richer features, the model can better adapt to the distribution of test data and improve its generalization ability.

$$FPR = \frac{FP}{TN + FP} \tag{11}$$

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

Table 3: Using 1st and 2nd Principal Components Confusion Matrix

ACTUALCLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	52	2
Class=No	6	83