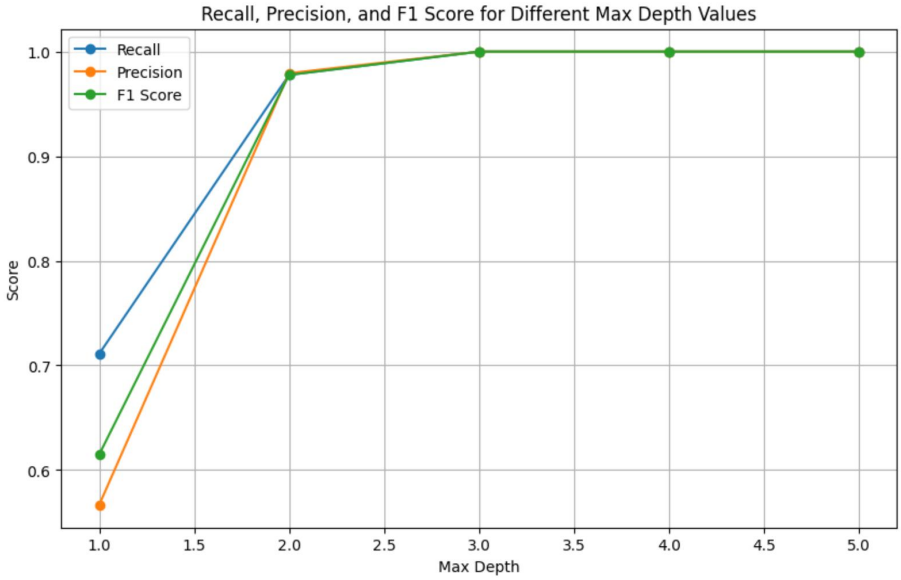
Practicum Problems

These problems will primarily reference the lecture materials and examples provided in class using Python. It is recommended that a Jupyter/IPython notebook be used for the programmatic components. Students are expected to refer to the prescribed textbook or credible online resources to answer the questions accurately.

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



Which depth values result in the highest Recall? Why?
The model with a max_depth of 5 has the highest Recall. As the depth increases, the tree captures more positive instances, improving Recall.

Which value resulted in the lowest Precision? Why?
max_depth = 1 resulted in the lowest Precision. A shallow tree may over-simplify and classify too many instances as positive, increasing false positives.

E.N.D

Which value results in the best F1 score?
The max_depth = 2 results in the best F1 score, balancing Precision and Recall without overfitting.

Explanation of Micro, Macro, and Weighted Methods of Score Calculation:

Micro-average: Global calculation based on total true positives, false positives, and false negatives.

Macro-average: Metric averaged across classes, treating all classes equally.

Weighted-average: Metric averaged by the number of instances in each class, useful for imbalanced data.

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?

1. Entropy, Gini, and Misclassification Error of the First Split:

Gini: 0.1070 (Left Child), 0.2045 (Right Child)

Entropy: 0.3144 (Left Child), 0.5166 (Right Child)

Misclassification Error: 0.0567 (Left Child), 0.1156 (Right Child)

2. Information Gain:

Gini Information Gain: 0.3229

Entropy Information Gain: 0.5605

Misclassification Error Information Gain: 0.2923

3. First Split Feature and Decision Boundary:

The **first split feature** is **concave_points1**.

The **decision boundary value** for this feature is **0.0513**.

4. Interpretation:

The tree uses **concave_points1** as the first feature to split the data, and the threshold for the decision boundary is set at **0.0513**. The decision boundary separates instances with **concave_points1** values less than or equal to 0.05 from those greater than 0.05.

Entropy Information Gain is the highest, indicating that this criterion captures the most information about the class distribution after the split.

The **training accuracy** of the model is 92.97%, showing good model performance.

E.N.D

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. F1 Score, Precision, and Recall:

Original Data:

F1 Score: 0.9048

Precision: 0.9048

Recall: 0.9048

PCA 1 Component:

F1 Score: 0.9063 (slight improvement from the original data)

Precision: 0.8923 (a small decrease from the original data)

Recall: 0.9206 (a significant improvement from the original data)

PCA 2 Components:

F1 Score: 0.8926 (a slight drop from the original data and PCA 1 component)

Precision: 0.9310 (a significant improvement from the original data)

Recall: 0.8571 (a significant decrease from the original data and PCA 1 component)

2. Confusion Matrix Values:

Original Data:

Fase Positive Rate (FPR): 0.0556

True Positive Rate (TPR): 0.9048

PCA 1 Component:

FPR: 0.0794 (slightly higher than the original data)

TPR: 0.9352 (higher than the original data, indicating better recall)

PCA 2 Components:

FPR: 0.1429 (significantly higher than the original and PCA 1 component models)

TPR: 0.9630 (a very high TPR, but at the cost of much lower recall)

3. Is Using Continuous Data Beneficial for the Model?

PCA 1 Component gives the best overall performance with a small improvement in F1 score and TPR compared to the original continuous data.

E.N.D

The precision is a little lower, but the improvement in recall outweighs the precision loss.

PCA 2 Components results in the highest precision but at the cost of a much lower recall and a much higher false positive rate, suggesting it may not be beneficial in this case.

E.N.D