A black background with grey leaves

Description automatically generated

[Document title]

[Document subtitle]

A grey logo on a black background

Description automatically generated

**Prepared by:**

Mohammed Mohsen Ali

Phoebe Thabit Wadea

Sandy Adel Latef

**Part 1:**

**a)**

* **For weather feature:**

**Gini (Cloudy)** = 1 - [ + ] = .

**Gini (Sunny)** = 1 - [ + ] = 0.

**Gini (Rainy)** = 1 - [ + ] = .

**Gini Children** = ( ) + ( ) + ( ) = 0.266667.

* **For temperature feature:**

**Gini (Hot)** = 1 - [ + ] = .

**Gini (Mild)** = 1 - [ + ] = .

**Gini (Cool)** = 1 - [ + ] = .

**Gini (Cold)** = 1 - [ + ] = .

**Gini Children** = ( ) + ( ) = 0.35.

* **For Humidity feature:**

**Gini (High)** = 1 - [ + ] = .

**Gini (Normal)** = 1 - [ + ] = .

**Gini Children** = ( ) + ( ) = 0.316667.

* **For Wind feature:**

**Gini (Strong)** = 1 - [ + ] = .

**Gini (Weak)** = 1 - [ + ] = .

**Gini Children** = ( ) + ( ) = 0.41905.

===================================================================

* Based on the Gini impurity calculations provided, we can determine the best feature to split on at the root node:

Of all the features, splitting on the Weather feature results in the lowest Gini impurity of 0.266667. Therefore, Weather will be chosen as the root node feature for the decision tree.

weather

Rainy

Cloudy

Sunny

No

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| so the new dataset will be as follows: |  |  |  |  |
| Weather | **Temperature** | **Humidity** | **Wind** | **Hiking** |
| Cloudy | Hot | High | Strong | No |
| Rainy | Cold | Normal | Strong | Yes |
| Cloudy | Mild | Normal | Strong | Yes |
| Rainy | Cool | Normal | Strong | No |
| Cloudy | Mild | High | Weak | Yes |
| Rainy | Hot | Normal | Weak | Yes |

* **For Temperature feature:**

**Gini (Hot)** = 1 - [ + ] = .

**Gini (Mild)** = 1 - [ + ] = 0.

**Gini (Cool)** = 1 - [ + ] = .

**Gini (Cold)** = 1 - [ + ] = .

**Gini Children** = ( ) = 0.166667.

* **For Humidity feature:**

**Gini (High)** = 1 - [ + ] = .

**Gini (Normal)** = 1 - [ + ] = .

**Gini Children** = ( ) + ( ) = 0.416667.

* **For wind feature:**

**Gini (Strong)** = 1 - [ + ] = .

**Gini (Normal)** = 1 - [ + ] = .

**Gini Children** = ( ) = 0.333333.

* The Humidity feature will therefore be the second node in the decision tree, as a child of the Weather root node.

A diagram of weather and weather

Description automatically generated

**b)**

E(S1) = log - log = 0.97095.

G (S1, Weather) = 0.97095 - ( log - log ) - 0 - ( log - log ) .

= 0.97095 – 0.27548 – 2.7548 = 0.41999.

==========================================================

G (S1, Temperature) = 0.97095 - ( log - log ) - ( log - log ) - (- log) - (- log)

= 0.97095 – 0.3245 –0.4= 0.24645.

==========================================================

G (S1, Humidity) = 0.97095 - ( log - log ) - ( log - log )

= 0.97095 – 0.390013 –0.32451= 0.256427.

==========================================================

G (S1, Wind) = 0.97095 - ( log - log ) - ( log - log )

= 0.97095 – 0.60418 –0.27548= 0.09129.

==========================================================

E(S2) = log - log = 0.91829.

G (S2, Temperature) = 0.91829 - ( log - log ) = 0.5849.

G (S2, Humidity) = 0.91829 - ( log - log ) - ( log - log ) = 0.91829 - 0.33333 – 0.54085 = 0.0441.

Rainy

Cloudy

Sunny

Hiking = No

Hot

Hot

Mild

Cold

Cool

Hiking = No

Hiking = Yes

c)

Comparing the pros and cons of the Gini Index and Information Gain

The Gini Index and Information Gain are widely used impurity measures in decision tree algorithms. Here's a comparison of their advantages and disadvantages:

Advantages of the Gini Index:

- Computational Efficiency: The Gini Index is generally faster to compute, especially for large datasets.

- Robust to Outliers: It is less sensitive to outliers, making it more stable in the presence of noise or irrelevant attributes.

- Better for Continuous Variables: The Gini Index can handle continuous variables directly without discretization.

Disadvantages of the Gini Index:

- Biased Towards Balanced Splits: It tends to favor attributes with balanced splits and may not perform as well on imbalanced datasets or attributes with skewed distributions.

- Ignores Class Probabilities: The Gini Index only considers class labels and their proportions, ignoring the actual probabilities associated with each class.

- Limited Information: It does not provide a direct measure of the information gained from a split.

Advantages of Information Gain:

- Captures Information Content: Information Gain explicitly measures the information gained by splitting on an attribute, considering the reduction in entropy or increase in purity.

- Handles Imbalanced Datasets: It is effective in handling imbalanced datasets by considering class probabilities and their distribution.

- Supports Different Attribute Types: Information Gain can handle both categorical and continuous attributes without bias.

Disadvantages of Information Gain:

- Computationally Expensive: Computing Information Gain requires calculating entropy for each feature, which can be computationally expensive for large datasets.

- Biased Towards Attributes with Many Values: It tends to favor attributes with a large number of distinct values, potentially leading to overfitting.

- Prone to Overemphasizing Irrelevant Attributes: Information Gain may assign high importance to attributes that are not directly relevant to the target variable, resulting in complex and less interpretable decision trees.

In summary, the Gini Index offers computational efficiency and robustness to outliers, while Information Gain captures information content and handles imbalanced datasets effectively. The choice between the two depends on factors such as class imbalance, attribute types, and the desired interpretability of the resulting decision tree.

Part 2: Programming Questions

1. Load the dataset which shows 39 columns and 494021 rows.

* First, we load important libraries.

A screen shot of a computer

Description automatically generated

* View the dataset which must show 38 input feature variables and 1 target, variable Obtain input feature variables as X and target variable as Y.

A screenshot of a computer program

Description automatically generated

* Normalize X using MinMaxScaler from sklearn library.

A close-up of a computer code

Description automatically generated

* Compute filter-based feature selection algorithm on dataset by reducing the number of feature variables to 10.

A screen shot of a computer code

Description automatically generated

* show the first five rows again and name this dataset as my data.

A screenshot of a computer

Description automatically generated

1. Use sklearn to split my data using train test split into three subsets, for instance.

* my data 1 with 70% train & 30% test data, my data 2 with 60%train & 40% test data, my data 3 with 50%train & 50% test data.

A screenshot of a computer code

Description automatically generated

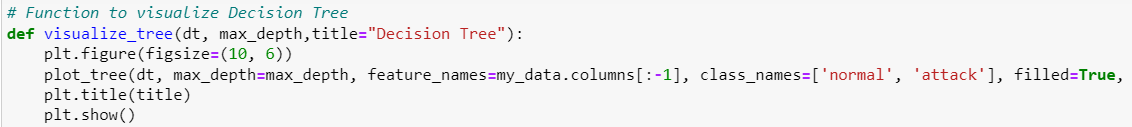
* compute the performance of Decision tree in terms of classification report for each subset.

A screenshot of a document

Description automatically generated

1. Visualize the best split of the Decision tree by considering Entropy as a measure of node impurity and assuming parameters max depth= [4, 6, 8] for each my data 1 with 70% train, my data 2 with 60%train and my data 3 with 50%train.

* First, we make a function to visualize Decision tree.



* Then, we make for loop to visualize (mydata\_1, mydata\_2, mydata\_3) for each depth [4, 6, 8] in Decision tree.

A screenshot of a computer program

Description automatically generated

* mydata\_1 for each depth = [4, 6, 8].

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

* mydata\_2 for each depth = [4, 6, 8].

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

* mydata\_3 for each depth = [4, 6, 8].

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated

1. Compute and compare the classification performance of tuned Decision Tree for each test size my data 1: 30% test data, my data 2: 40% test data, my data 3: 50% test data.

* First, we make a function to plot confusion matrix.

A computer screen shot of text

Description automatically generated

* Then, we make a function to evaluate performance and metrics.

A close-up of text

Description automatically generated

* Then, we make a function to print performance and display metrics.

A close-up of text

Description automatically generated

* we make for loop to compute the accuracy scores, classification report, and confusion matrix respectively (mydata\_1, mydata\_2, mydata\_3) for each depth [4, 6, 8] in Decision tree.

A screenshot of a computer program

Description automatically generated

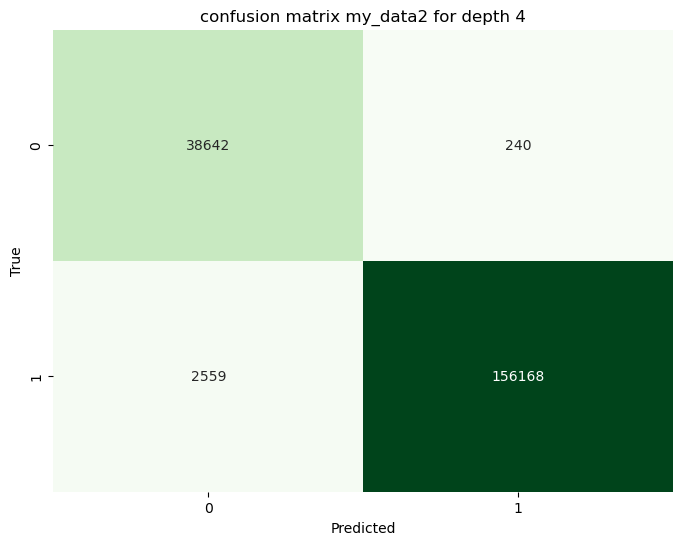
* depth 4 performance for each (my\_data1, my\_data2, my\_data3).

A green squares with white numbers

Description automatically generated

A screenshot of a computer

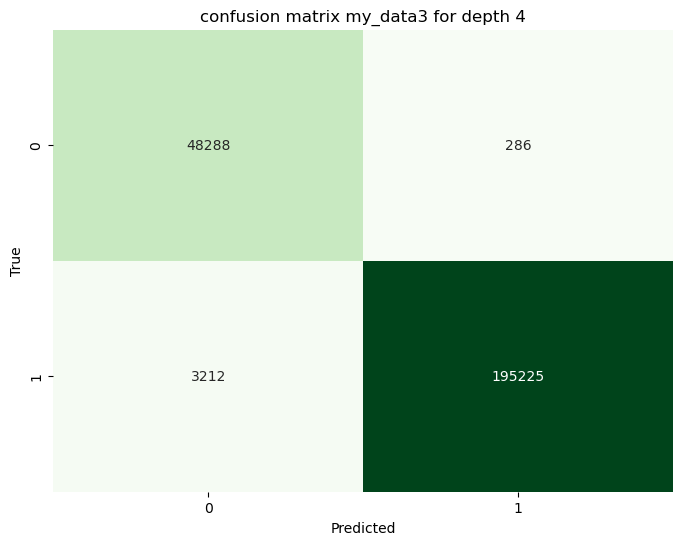
Description automatically generated



A screenshot of a computer

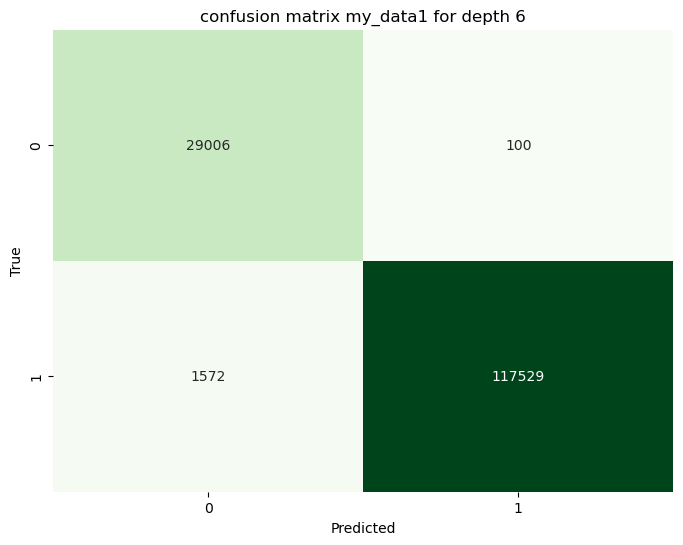
Description automatically generatedA screenshot of a computer

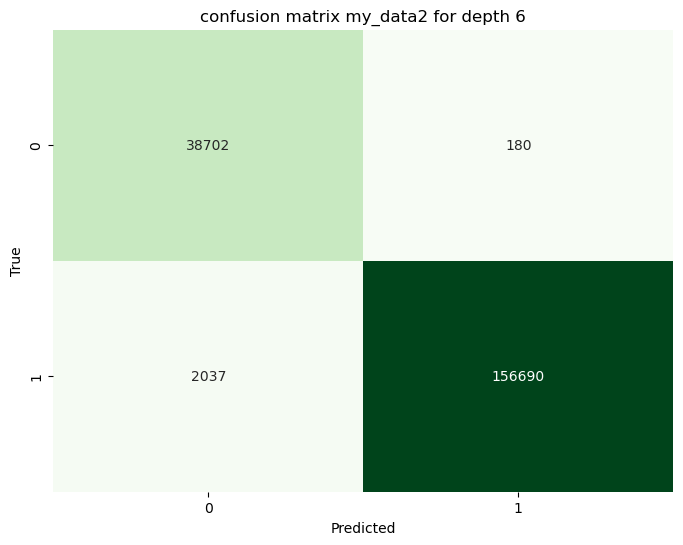
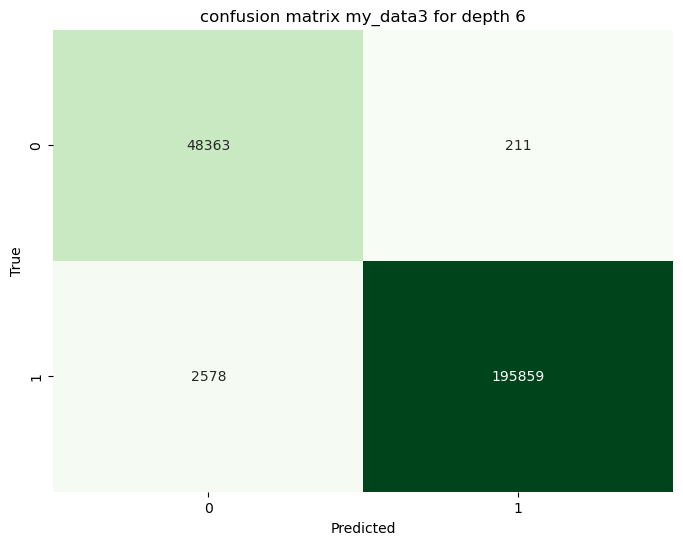
Description automatically generated



* depth 6 performance for each (my\_data1, my\_data2, my\_data3).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* depth 8 performance for each (my\_data1, my\_data2, my\_data3).

A green squares with white numbers

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A green squares with white numbers

Description automatically generatedA green squares with black numbers

Description automatically generated

1. Train Decision Tree with parameters of your choice on my data1.

* display the F1 scores for both train and test data showcasing an issue of overfitting or overlearning.

A screenshot of a computer code

Description automatically generated

* apply three mitigation strategies (pre-prunning, post-prunning and k-fold cross validation) to address the problem of overfitting.

A screenshot of a computer program

Description automatically generated

* display the training and test F1 scores.

A white background with numbers and letters

Description automatically generated

1. additional visualization for Overfitting Performance for Decision Tree.

A graph with blue and orange lines

Description automatically generated