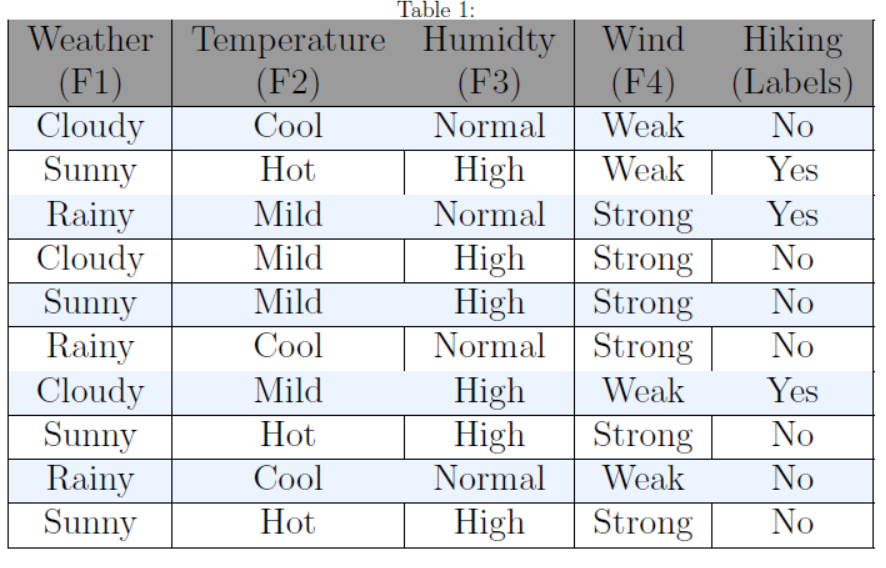
**ELG5255 Applied Machine Learning**

REPORT of: Group Assignment 4 (Group-18)

**Part 1: Calculations**



1. **Build a decision tree by using Gini Index (i.e., Gini = 1 - , where is the number of class).**

Hiking (labels) 🡪 P (Yes) = , P (No) =

**We will calculate probabilities of classes in F1, F2, F3, and F4 in this table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Cloudy) = | P(F2 = Cool) = | P(F3 = Normal) = | P(F4 = Weak) = |
| P(F1 = Sunny) = | P(F2 = Hot) = | P(F3 = High) = | P(F4 = Strong) = |
| P(F1 = Rainy) = | P(F2 = Mild) = |  |  |

**We will calculate the Gini Index for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Cloudy and Hiking = Yes) = | P(F1 = Cloudy and Hiking = No) = |
| P(F1 = Sunny and Hiking = Yes) = | P(F1 = Sunny and Hiking = No) = |
| P(F1 = Rainy and Hiking = Yes) = | P(F1 = Rainy and Hiking = No) = |

Gini Index of Cloudy = 1-(+) = 0.44

Gini Index of Sunny = 1-(+) = 0.375

Gini Index of Rainy = 1-(+) = 0.44

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Weather (F1) = \* 0.44 + \* 0.375 + \* 0.44 = 0.414

**We will calculate the Gini Index for Temperature (F2)**

|  |  |
| --- | --- |
| Temperature (F2) | |
| P(F2 = Cool and Hiking = Yes) = | P(F2 = Cool and Hiking = No) = |
| P(F2 = Hot and Hiking = Yes) = | P(F2 = Hot and Hiking = No) = |
| P(F2 = Mild and Hiking = Yes) = | P(F2 = Mild and Hiking = No) = |

Gini Index of Cool = 1-(+) = 0

Gini Index of Hot = 1-(+) = 0.44

Gini Index of Mild = 1-(+) = 0.5

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Temperature (F2) = \* 0 + \* 0.44 + \* 0.5 = 0.332

**We will calculate the Gini Index for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = Normal and Hiking = Yes) = | P(F3 = Normal and Hiking = No) = |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

Gini Index of Normal = 1-(+) = 0.375

Gini Index of High = 1-(+) = 0.44

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Humidty (F3) = \* 0.375 + \* 0.44 = 0.414

**We will calculate the Gini Index for Wind (F4)**

|  |  |
| --- | --- |
| Wind(F4) | |
| P(F4 = Weak and Hiking = Yes) = | P(F4 = Weak and Hiking = No) = |
| P(F4 = Strong and Hiking = Yes) = | P(F4 = Strong and Hiking = No) = |

Gini Index of Weak = 1-(+) = 0.5

Gini Index of Strong = 1-(+) = 0.278

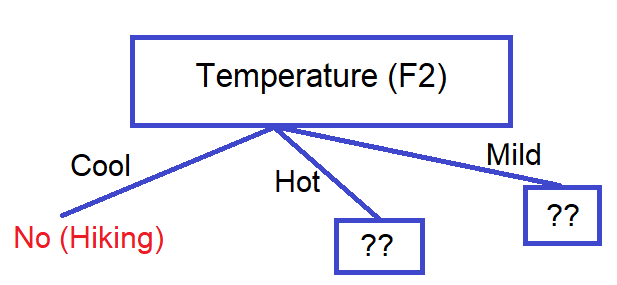
Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Wind (F4) = \* 0.5 + \* 0.278 = 0.367

**Gini Index attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | 0.414 |
| Temperature (F2) | **0.332** |
| Humidty (F3) | 0.414 |
| Wind (F4) | 0.367 |

From the above table, we observe that ‘Temperature (F2)’ has the lowest Gini Index and hence it will be chosen as the root node for how decision tree works.



We will repeat the same procedure to determine the sub-nodes or branches of the decision tree.

We will calculate the Gini Index for the ‘Hot’ branch of Temperature (F2) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Sunny | Hot | High | Weak | Yes |
| Sunny | Hot | High | Strong | No |
| Sunny | Hot | High | Strong | No |

**We will calculate probabilities of classes in F1, F3, and F4 in this table:**

|  |  |  |
| --- | --- | --- |
| **Weather (F1)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Sunny) = | P(F3 = High) = | P(F4 = Weak) = |
|  |  | P(F4 = Strong) = |

**We will calculate the Gini Index for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Sunny and Hiking = Yes) = | P(F1 = Sunny and Hiking = No) = |

Gini Index of Sunny = 1-(+) = 0.44

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Weather (F1) = \* 0.44 = 0.44

**We will calculate the Gini Index for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

Gini Index of High = 1-(+) = 0.44

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Humidty (F3) = \* 0.44 = 0.44

**We will calculate the Gini Index for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
| P(F4 = Weak and Hiking = Yes) = |  |
|  | P(F4 = Strong and Hiking = No) = |

Gini Index of Weak = 1- () = 0

Gini Index of High = 1- () = 0

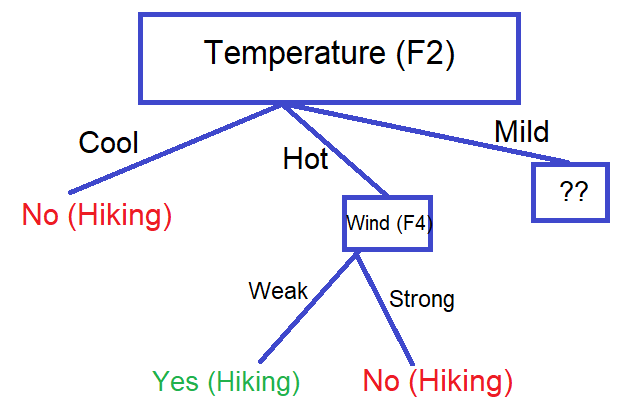
Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Wind (F4) = \* 0 + \* 0 = 0

**Gini Index attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | 0.44 |
| Humidty (F3) | 0.44 |
| Wind (F4) | **0** |

From the above table, we observe that ‘Wind (F4)’ has the lowest Gini Index and hence it will be chosen as the child node for the ‘Hot’ branch of Temperature (F2).



We will calculate the Gini Index for the ‘Mild’ branch of Temperature (F2) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Rainy | Mild | Normal | Strong | Yes |
| Cloudy | Mild | High | Strong | No |
| Sunny | Mild | High | Strong | No |
| Cloudy | Mild | High | Weak | Yes |

**We will calculate probabilities of classes in F1, F3, and F4 in the above table:**

|  |  |  |
| --- | --- | --- |
| **Weather (F1)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Rainy) = | P(F3 = Normal) = | P(F4 = Strong) = |
| P(F1 = Cloudy) = | P(F3 = High) = | P(F4 = Weak) = |
| P(F1 = Sunny) = |  |  |

**We will calculate the Gini Index for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Rainy and Hiking = Yes) = |  |
| P(F1 = Cloudy and Hiking = Yes) = | P(F1 = Cloudy and Hiking = No) = |
| P(F1 = Sunny and Hiking = Yes) = |  |

Gini Index of Rainy = 1-() = 0

Gini Index of Cloudy = 1-(+) = 0.5

Gini Index of Sunny = 1-() = 0

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Weather (F1) = \* 0 + \* 0.5 + \* 0 = 0.25

**We will calculate the Gini Index for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = Normal and Hiking = Yes) = |  |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

Gini Index of Normal = 1-() = 0

Gini Index of High = 1-(+) = 0.44

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Humidty (F3) = \* 0 + \* 0.44 = 0.33

**We will calculate the Gini Index for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
| P(F4 = Weak and Hiking = Yes) = |  |
| P(F4 = Strong and Hiking = Yes) = | P(F4 = Strong and Hiking = No) = |

Gini Index of Weak = 1-() = 0

Gini Index of Strong = 1-(+) = 0.44

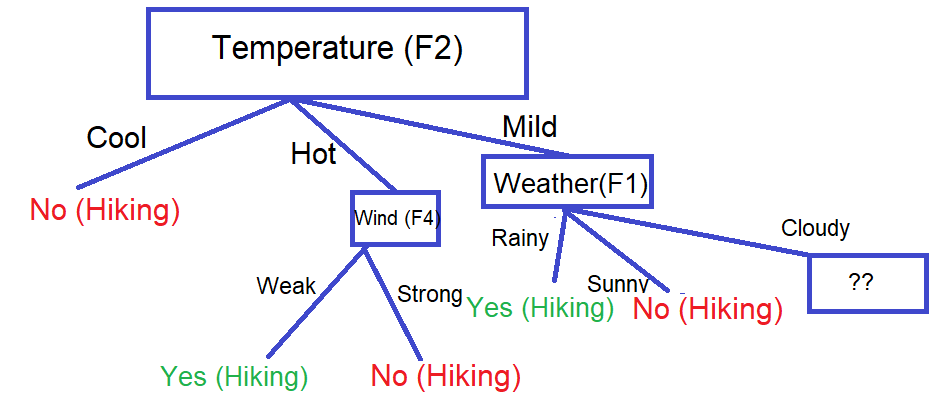
Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Wind (F4) = \* 0 + \* 0.44 = 0.33

**Gini Index attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | **0.25** |
| Humidty (F3) | 0.33 |
| Wind (F4) | 0.33 |

From the above table, we observe that ‘Weather (F1)’ has the lowest Gini Index and hence it will be chosen as the child node for the ‘Mild’ branch of Temperature (F2).



We will calculate the Gini Index for the ‘Cloudy’ branch of Weather (F1) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Cloudy | Mild | High | Strong | No |
| Cloudy | Mild | High | Weak | Yes |

**We will calculate probabilities of classes in F1, F3, and F4 in this table:**

|  |  |
| --- | --- |
| **Humidty (F3)** | **Wind(F4)** |
| P(F3 = High) = | P(F4 = Strong) = |
|  | P(F4 = Weak) = |

**We will calculate the Gini Index for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

Gini Index of High = 1-(+) = 0.5

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Humidty (F3) = \* 0.5 = 0.5

**We will calculate the Gini Index for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
|  | P(F4 = Strong and Hiking = No) = |
| P(F4 = Weak and Hiking = Yes) = |  |

Gini Index of Strong = 1-(= 0.5

Gini Index of Weak = 1-(= 0.5

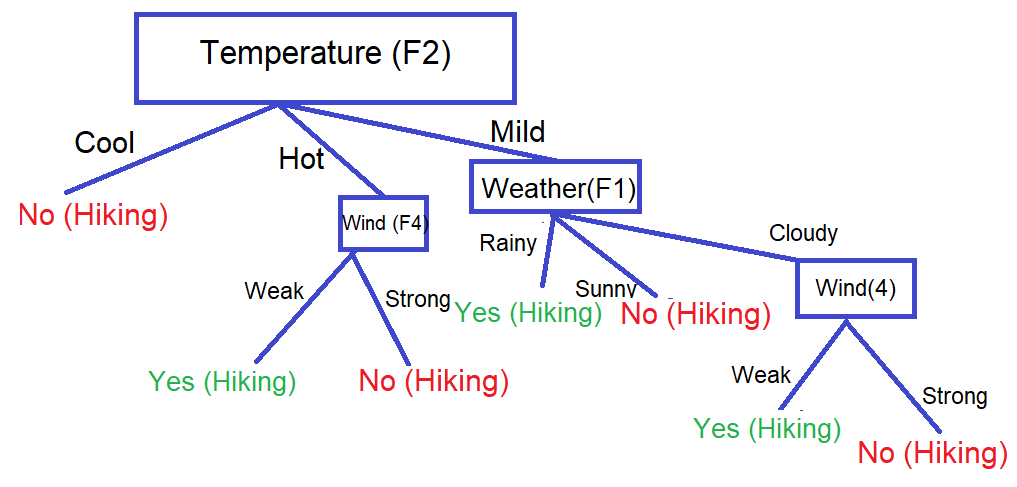
Weighted sum of the Gini Indices can be calculated as follows:

Gini Index of Wind (F4) = \* 0 + \* 0 = 0

**Gini Index attributes or features**

|  |  |
| --- | --- |
| Humidty (F3) | 0.5 |
| Wind (F4) | **0** |

From the above table, we observe that ‘Wind (F4)’ has the lowest Gini Index and hence it will be chosen as the child node for the ‘Cloudy’ branch of Weather (F1).

****

1. **Build a decision tree by using Information Gain (i.e., IG (T, a) = Entropy (T) – Entropy (T |a), More information about IG).**

The first thing that we need to do is work out which feature to use as the root node. We start by computing the entropy of hiking (labels):

🡪 P (Yes) = , P (No) =

Entropy (Hiking) = - - = 0.881

**We will calculate probabilities of classes in F1, F2, F3, and F4 in this table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Cloudy) = | P(F2 = Cool) = | P(F3 = Normal) = | P(F4 = Weak) = |
| P(F1 = Sunny) = | P(F2 = Hot) = | P(F3 = High) = | P(F4 = Strong) = |
| P(F1 = Rainy) = | P(F2 = Mild) = |  |  |

**We will calculate the Information Gain for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Cloudy and Hiking = Yes) = | P(F1 = Cloudy and Hiking = No) = |
| P(F1 = Sunny and Hiking = Yes) = | P(F1 = Sunny and Hiking = No) = |
| P(F1 = Rainy and Hiking = Yes) = | P(F1 = Rainy and Hiking = No) = |

GAIN (Hiking, Weather (F1)) = 0.881 - Entropy ()

* Entropy ()
* Entropy ()

GAIN (Hiking, Weather (F1)) = 0.881 - (- - )

* (- - )
* (- - )

= 0.881 - 0.275 - 0.234 - 0.275 = 0.097

**We will calculate the Information Gain for Temperature (F2)**

|  |  |
| --- | --- |
| Temperature (F2) | |
| P(F2 = Cool and Hiking = Yes) = | P(F2 = Cool and Hiking = No) = |
| P(F2 = Hot and Hiking = Yes) = | P(F2 = Hot and Hiking = No) = |
| P(F2 = Mild and Hiking = Yes) = | P(F2 = Mild and Hiking = No) = |

GAIN (Hiking, Temperature (F2)) = 0.881 - Entropy ()

* Entropy ()
* Entropy ()

GAIN (Hiking, Temperature (F2)) = 0.881 - (- - )

* (- - )
* (- - )

= 0.881 - 0 - 0.275 - 0.4 = 0.206

**We will calculate the Information Gain for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = Normal and Hiking = Yes) = | P(F3 = Normal and Hiking = No) = |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

GAIN (Hiking, Humidty (F3)) = 0.881 - Entropy ()

* Entropy ()

GAIN (Hiking, Humidty (F3)) = 0.881 - (- - )

* (- - )

= 0.881 - 0.324 - 0.551 = 0.006

**We will calculate the Information Gain for Wind (F4)**

|  |  |
| --- | --- |
| Wind(F4) | |
| P(F4 = Weak and Hiking = Yes) = | P(F4 = Weak and Hiking = No) = |
| P(F4 = Strong and Hiking = Yes) = | P(F4 = Strong and Hiking = No) = |

GAIN (Hiking, Wind (F4)) = 0.881 - Entropy ()

* Entropy ()

GAIN (Hiking, Wind (F4)) = 0.881 - (- - )

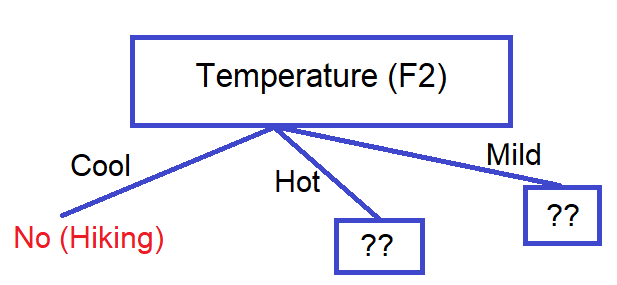
* (- - )

= 0.881 - 0.4 - 0.39 = 0.091

**Information Gain** **attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | 0.097 |
| Temperature (F2) | **0.206** |
| Humidty (F3) | 0.006 |
| Wind (F4) | 0.091 |

From the above table, we observe that ‘Temperature (F2)’ has the highest Information Gain and hence it will be chosen as the root node for how decision tree works.



We will repeat the same procedure to determine the sub-nodes or branches of the decision tree.

We will calculate the Information Gain for the ‘Hot’ branch of Temperature (F2) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Sunny | Hot | High | Weak | Yes |
| Sunny | Hot | High | Strong | No |
| Sunny | Hot | High | Strong | No |

We start by computing the entropy of hiking (labels) in the above table:

🡪 P (Yes) = , P (No) =

Entropy (Hiking) = - - = 0.918

**We will calculate probabilities of classes in F1, F3, and F4 in this table:**

|  |  |  |
| --- | --- | --- |
| **Weather (F1)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Sunny) = | P(F3 = High) = | P(F4 = Weak) = |
|  |  | P(F4 = Strong) = |

**We will calculate the Information Gain for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Sunny and Hiking = Yes) = | P(F1 = Sunny and Hiking = No) = |

GAIN (Hiking, Weather (F1)) = 0.918 - Entropy ()

GAIN (Hiking, Weather (F1)) = 0.918 - (- - )

= 0.918 - 0.918 = 0

**We will calculate the Information Gain for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

GAIN (Hiking, Humidty (F3)) = 0.918 - Entropy ()

GAIN (Hiking, Humidty (F3)) = 0.918 - (- - )

= 0.918 - 0.918 = 0

**We will calculate the Information Gain for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
| P(F4 = Weak and Hiking = Yes) = |  |
|  | P(F4 = Strong and Hiking = No) = |

GAIN (Hiking, Wind (F4)) = 0.918 - Entropy ()

- Entropy ()

GAIN (Hiking, Wind (F4)) = 0.918 - (- )

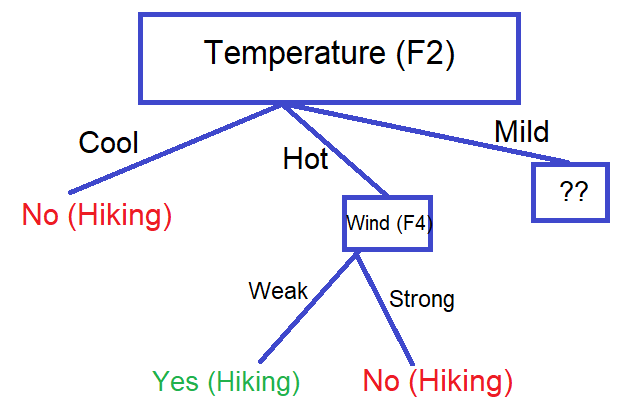
- (- )

= 0.918 - 0 - 0 = 0.918

**Information Gain attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | 0 |
| Humidty (F3) | 0 |
| Wind (F4) | **0.918** |

From the above table, we observe that ‘Wind (F4)’ has the highest Information Gain and hence it will be chosen as the child node for the ‘Hot’ branch of Temperature (F2).



We will calculate the Information Gain for the ‘Mild’ branch of Temperature (F2) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Rainy | Mild | Normal | Strong | Yes |
| Cloudy | Mild | High | Strong | No |
| Sunny | Mild | High | Strong | No |
| Cloudy | Mild | High | Weak | Yes |

We start by computing the entropy of hiking (labels) in the above table:

🡪 P (Yes) = , P (No) =

Entropy (Hiking) = - - = 1

**We will calculate probabilities of classes in F1, F3, and F4 in the above table:**

|  |  |  |
| --- | --- | --- |
| **Weather (F1)** | **Humidty (F3)** | **Wind(F4)** |
| P(F1 = Rainy) = | P(F3 = Normal) = | P(F4 = Strong) = |
| P(F1 = Cloudy) = | P(F3 = High) = | P(F4 = Weak) = |
| P(F1 = Sunny) = |  |  |

**We will calculate the Information Gain for Weather (F1)**

|  |  |
| --- | --- |
| Weather (F1) | |
| P(F1 = Rainy and Hiking = Yes) = |  |
| P(F1 = Cloudy and Hiking = Yes) = | P(F1 = Cloudy and Hiking = No) = |
| P(F1 = Sunny and Hiking = Yes) = |  |

GAIN (Hiking, Weather (F1)) = 1- Entropy ()

- Entropy ()

- Entropy ()

GAIN (Hiking, Weather (F1)) = 1 - (- ) - (- - ) - (- )

= 1 - 0 – 0.5 - 0 = 0.5

**We will calculate Information Gain for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = Normal and Hiking = Yes) = |  |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

GAIN (Hiking, Humidty (F3)) = 1- Entropy ()

- Entropy ()

GAIN (Hiking, Humidty (F3)) = 1 - (- ) - (- - )

= 1 - 0 - 0.612 = 0.388

**We will calculate the Information Gain for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
| P(F4 = Weak and Hiking = Yes) = |  |
| P(F4 = Strong and Hiking = Yes) = | P(F4 = Strong and Hiking = No) = |

GAIN (Hiking, Wind (F4)) = 1- Entropy ()

- Entropy ()

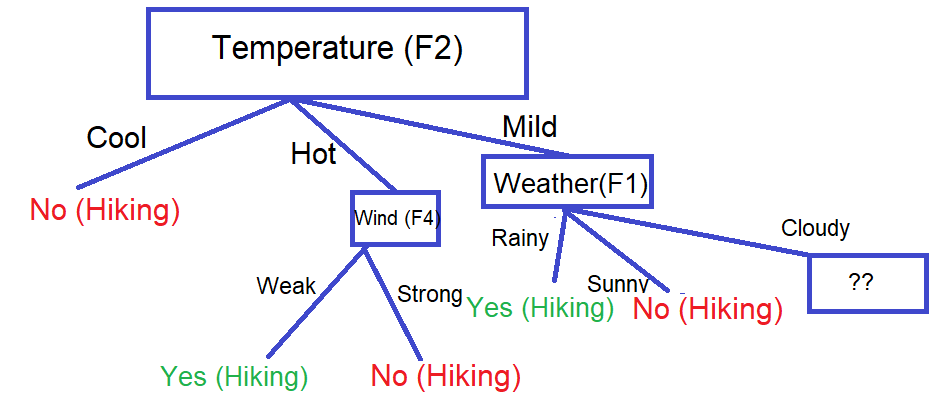
GAIN (Hiking, Wind (F4)) = 1 - (- ) - (- - )

= 1 - 0 - 0.612 = 0.388

**Information Gain attributes or features**

|  |  |
| --- | --- |
| Weather (F1) | **0.5** |
| Humidty (F3) | 0.388 |
| Wind (F4) | 0.388 |

From the above table, we observe that ‘Weather (F1)’ has the highest Information Gain and hence it will be chosen as the child node for the ‘Mild’ branch of Temperature (F2).



We will calculate the Information Gain for the ‘Cloudy’ branch of Weather (F1) as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weather (F1)** | **Temperature (F2)** | **Humidty (F3)** | **Wind(F4)** | **Hiking** |
| Cloudy | Mild | High | Strong | No |
| Cloudy | Mild | High | Weak | Yes |

We start by computing the entropy of hiking (labels) in the above table:

- P (Yes) = , P (No) =

Entropy (Hiking) = - - = 1

**We will calculate probabilities of classes in F1, F3, and F4 in this table:**

|  |  |
| --- | --- |
| **Humidty (F3)** | **Wind(F4)** |
| P(F3 = High) = | P(F4 = Strong) = |
|  | P(F4 = Weak) = |

**We will calculate the Information Gain for Humidty (F3)**

|  |  |
| --- | --- |
| Humidty (F3) | |
| P(F3 = High and Hiking = Yes) = | P(F3 = High and Hiking = No) = |

GAIN (Hiking, Wind (F4)) = 1- Entropy ()

GAIN (Hiking, Wind (F4)) = 1 - (- - ) = 1 – 1 = **0**

**We will calculate the Gini Index for Wind (F4)**

|  |  |
| --- | --- |
| Wind (F4) | |
|  | P(F4 = Strong and Hiking = No) = |
| P(F4 = Weak and Hiking = Yes) = |  |

GAIN (Hiking, Wind (F4)) = 1- Entropy ()

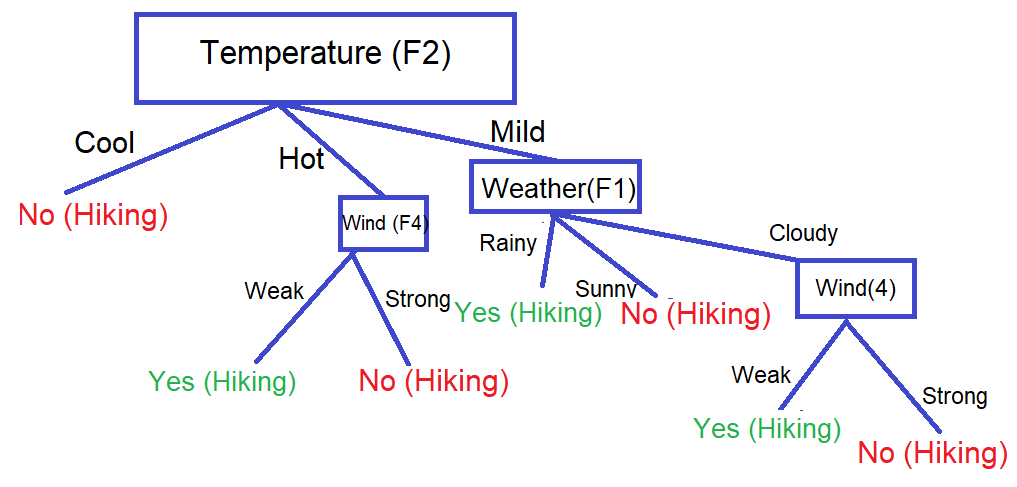
- Entropy ()

GAIN (Hiking, Wind (F4)) = 1 - (- - (- ) = 1 – 0 - 0 = **1**

**Information Gain attributes or features**

|  |  |
| --- | --- |
| Humidty (F3) | 0 |
| Wind (F4) | **1** |

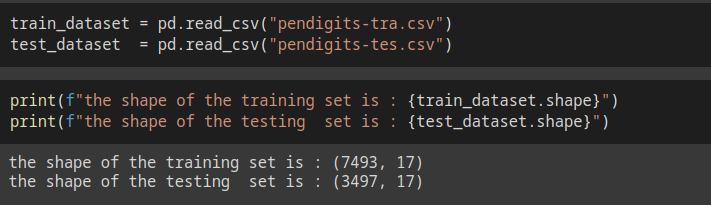
From the above table, we observe that ‘Wind (F4)’ has the highest Information Gain and hence it will be chosen as the child node for the ‘Cloudy’ branch of Weather (F1).

****

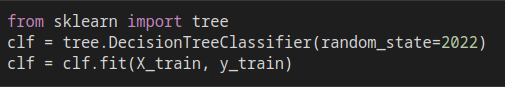
1. **Compare the advantages and disadvantages between Gini Index and Information Gain.**

|  |  |  |
| --- | --- | --- |
|  | **Gini Index** | **Information Gain** |
| **The advantages** | * It favors larger partitions (distributions) and is very easy to implement. * It can handle the values that are non-negative because it is measured by subtracting the sum of squared probabilities of each class from one. * It computes the degree of probability of a specific variable that is wrongly being classified when chosen randomly and a variation of the Gini coefficient. | * It favors partitions that have small counts but many distinct values. * It measures the entropy differences before and after splitting and depicts the impurity in class variables. * It uses Entropy as the base calculation; you have a wider range of results. * It computes the difference between entropy before and after the split and specifies the impurity in-class elements. * The feature with the highest information gain value is accounted for as the best feature to be chosen for the split. |
| **The disadvantages** | * The Gini Index doesn’t have a wider range of results, but it caps at one. * While working on categorical data variables, the Gini index results either in “success” or “failure” and only performs binary splitting. * It is prone to systematic and random data errors. Therefore, inaccurate data can distort the validity of the coefficient. | * It is not preferred as it involves a ‘log’ function that results in computational complexity. * It can’t handle the values that are non-negative. * It supports smaller partitions (distributions) with various distinct values; there is a need to perform an experiment with data and splitting criteria. |

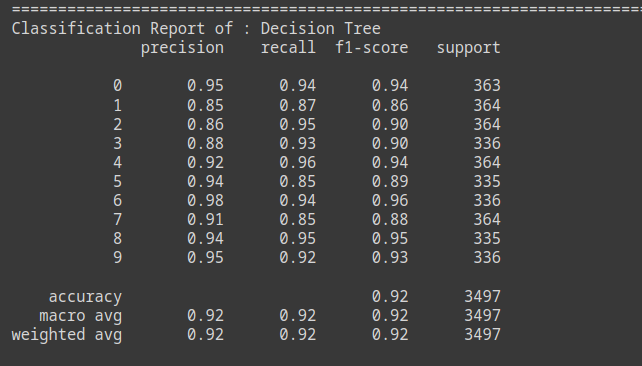
**Part 2: Programming Questions**

****

****

****

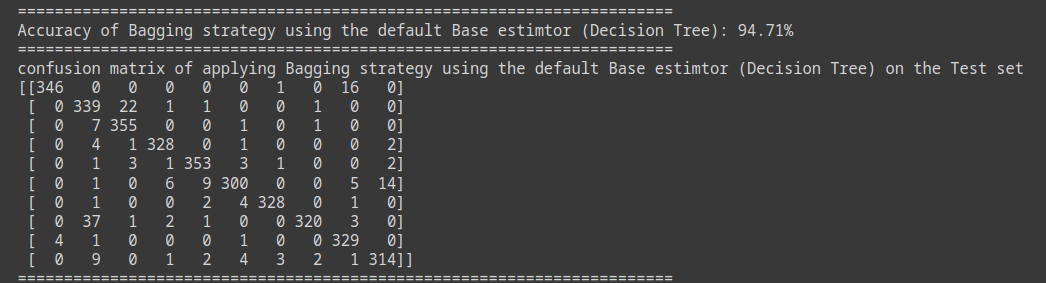
|  |  |
| --- | --- |
|  |  |

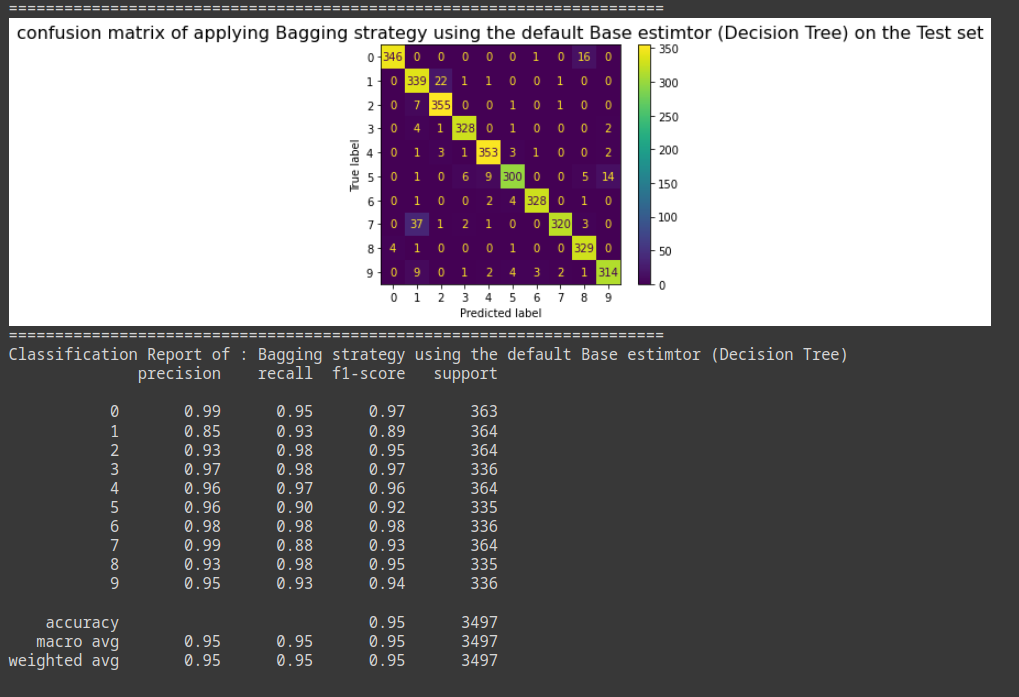
****

****

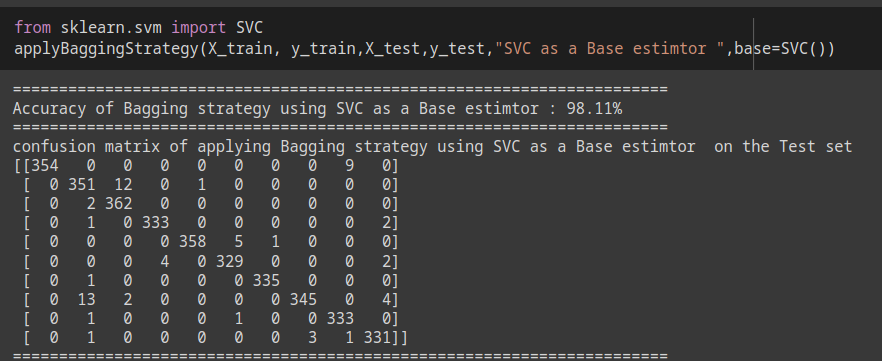
****

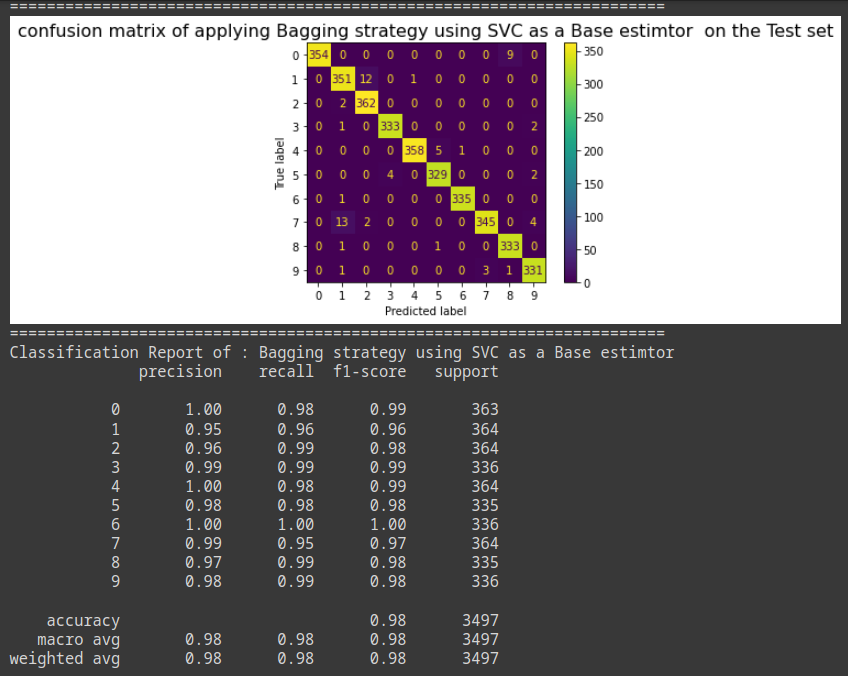
****

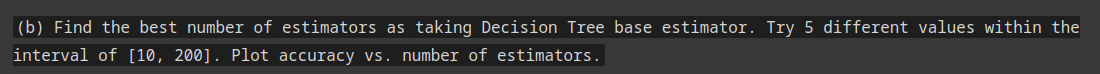
****

****

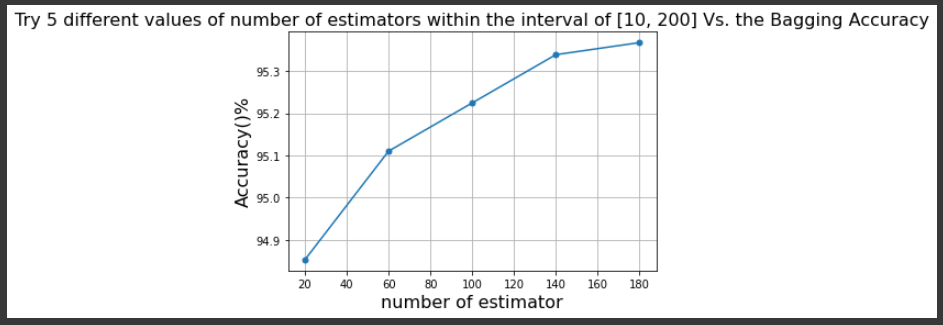
****

****

****

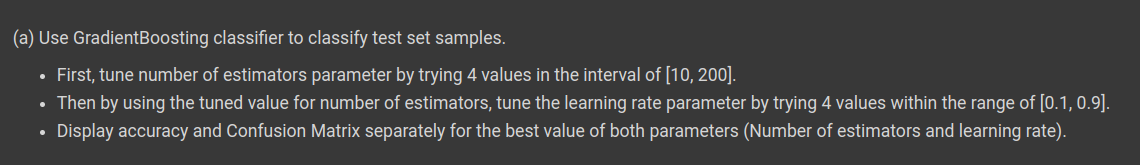
****

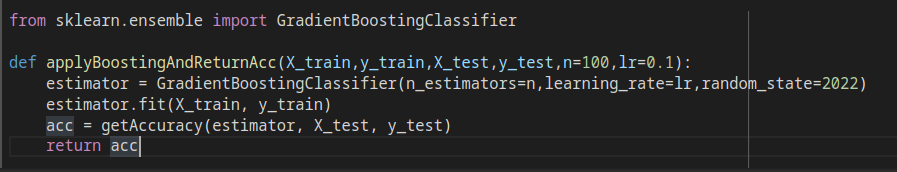
|  |  |
| --- | --- |
|  |  |

****

**Comment: the greater the number of the estimators the higher the accuracy we got.**

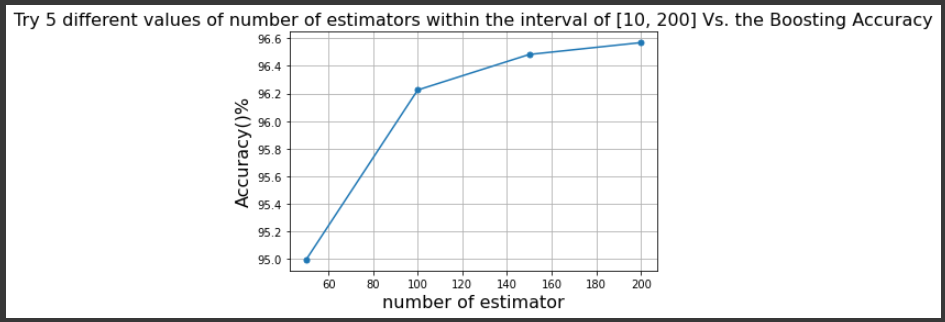
****

****

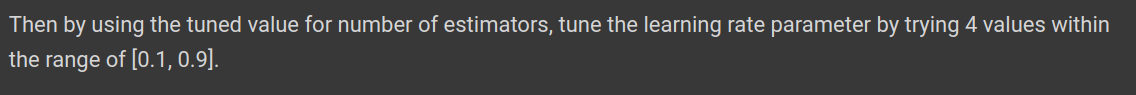
****

****

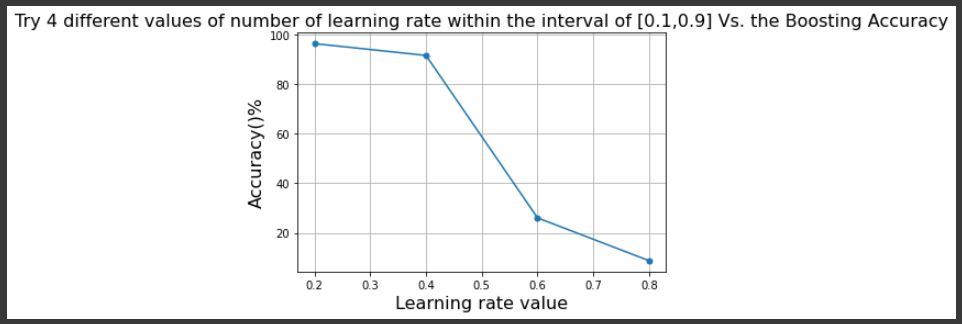
|  |  |
| --- | --- |
|  |  |

****

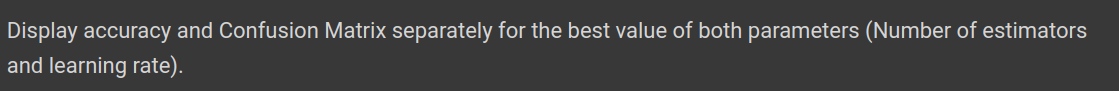
**comment: the greater the number of estimators the higher the accuracy**

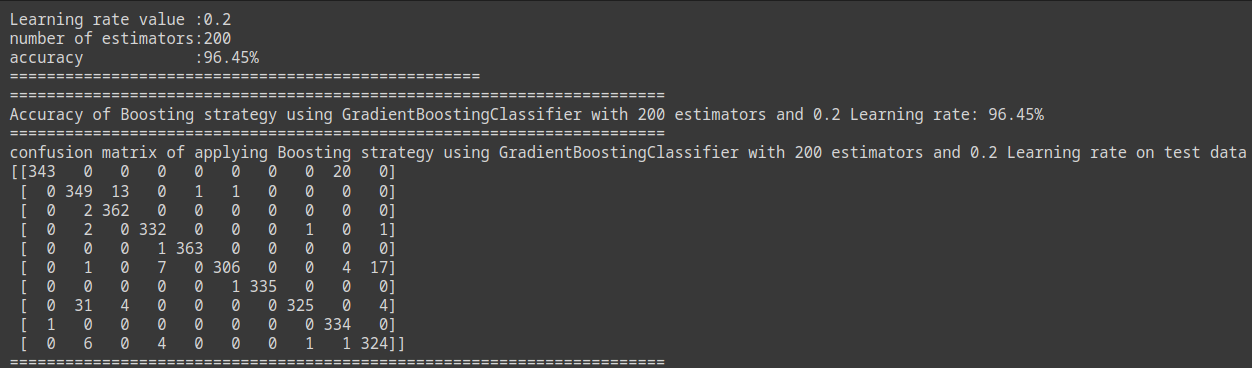
****

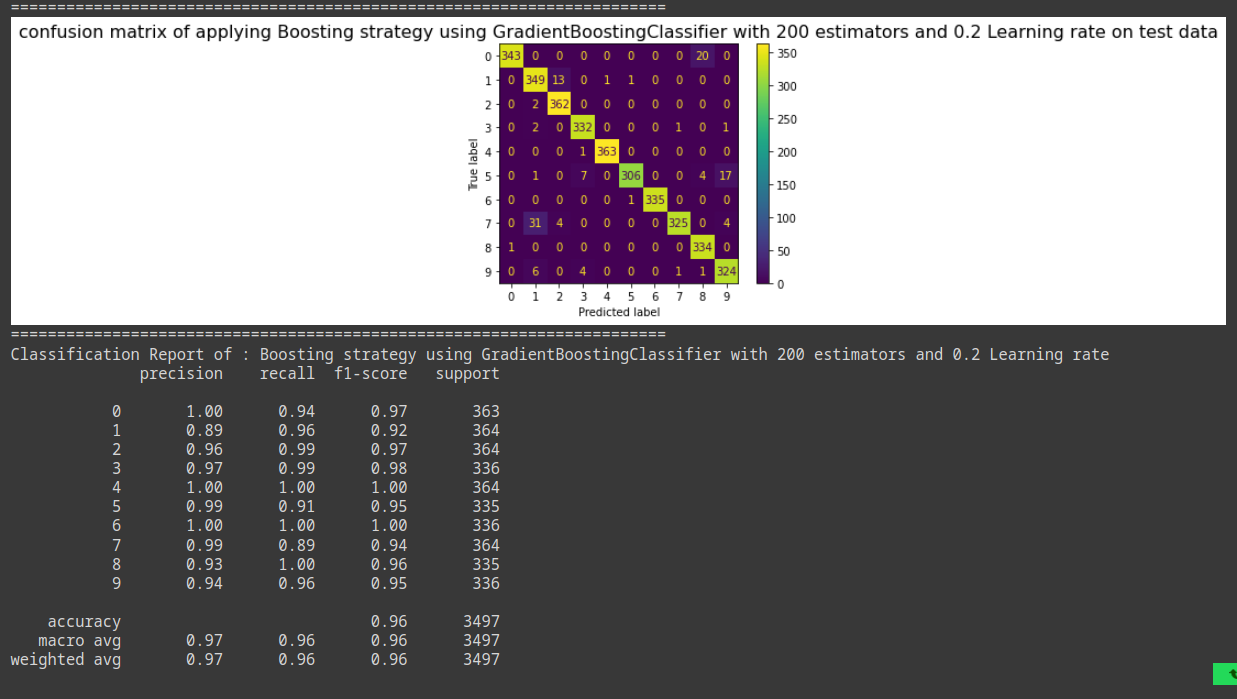
|  |  |
| --- | --- |
|  |  |

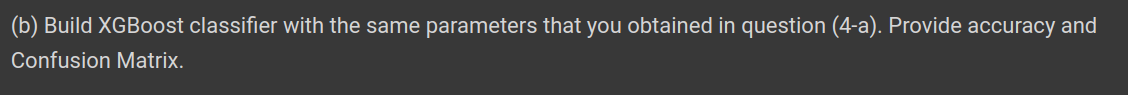
****

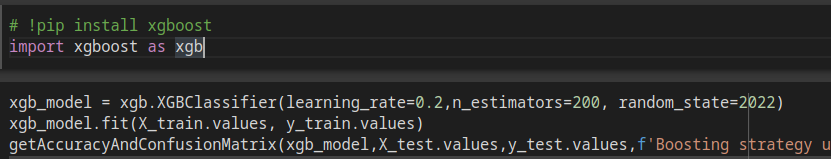
**comment: the greater the learning rate the lower the accuracy of the model.**

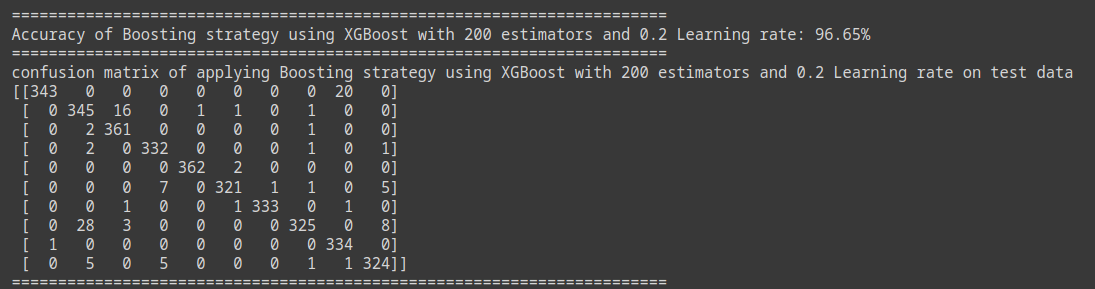
****

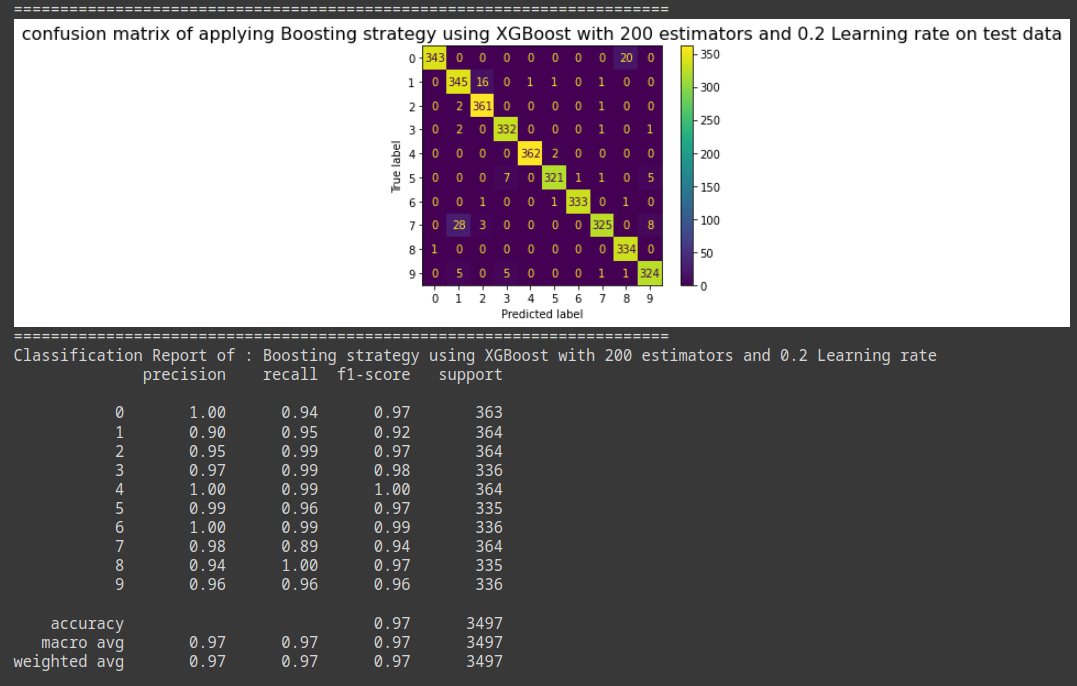
****

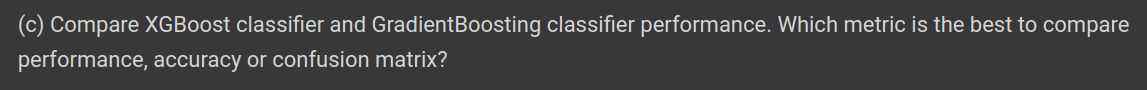
****

****

****

****

****

****

|  |  |
| --- | --- |
|  | **Accuracy** |
| **GradientBoosting**  **Classifier** |  |
| **XGBoost Classifier** |  |

|  |  |
| --- | --- |
|  | **Confusion Matrix** |
| **GradientBoosting**  **Classifier** |  |
| **XGBoost Classifier** |  |

|  |  |
| --- | --- |
|  | **Classification Report** |
| **GradientBoosting**  **Classifier** |  |
| **XGBoost Classifier** |  |

**Comparison Between the Two Models**

**Regarding the accuracy as a performance measure**

* with the same parameter **(n\_estimators =200, LR=0.2)**the accuracy of xgBoost **(96.65%)**is a bit better than the accuracy of the Gradient Boosting Classifier**(96.45%).**
* **So by using the accuracy as a comparison measure we only understood which predicted better but we can't understand the error of the model. so we couldn’t handle which class the model couldn’t predict correctly.**

**Regarding the confusion Matrix as a performance measure :**

**Confusion matrices can help with side-by-side comparisons of different classification methods: Precision, Recall, Accuracy, and F1 Score(**The closer to 1, the better the model**)**.

regarding our dataset, we are dealing with a multiclass classification problem so, we will use the macro average or weighted average to compare the two models in terms of F-score.

The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean (aka **unweighted** mean) of all the per-class F1 scores.

**The weighted-average F1 score** is calculated by taking the mean of all per-class F1 scores while considering each class’s support.

**So regarding the F-score weighted average**

**Gradient Boosting:** 0.96

**XGBoost classifier:** 0.97

so XGBoost predicts a bit better than the Gradient Boosting in our case because it is closer to 1.

**So regarding the recall weighted average**

**Gradient Boosting:** 0.96

**XGBoost classifiers:** 0.97

so XGBoost have a higher recall than Gradient Boosting however both of them have the same parameter.

**So regarding the precision weighted average**

**Gradient Boosting:** 0.97

**XGBoost classifiers:** 0.97

both XGBoost and Gradient Boosting have the same value.

**So regarding the Accuracy**

**Gradient Boosting:** 0.96

**XGBoost classifiers:** 0.97

so in terms of accuracy XGBoost is a bit better.

**So Confusion matrix is better as a performance measure because it provided us with more knowledge about the results than the Accuracy as a performance measure.**

****

1. Bagging is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.
2. Boosting is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

|  |  |  |
| --- | --- | --- |
|  | **Decision tree**  **(n\_estimator = 10,LR=0.1)** | **SVM**  **(n\_estimator = 10,LR=0.1)** |
| **Bagging accuracy** | **94.71 %** | **98.11 %** |

|  |  |  |
| --- | --- | --- |
|  | **Gradiant Boosting**  **(n\_estimator = 200,LR=0.2)** | **xgBoost**  **(n\_estimator = 200,LR=0.2)** |
| **Boosting accuracy** | **96.45%** | **96.65%** |

**So regarding to our experiment in the assignment the best bagging model with SVM of estimator = 10 and learning rate = 0.1 with accuracy 98.11 %**

**and the best boosting model was the xgboost model (n\_estimator = 200,LR=0.2) with accuracy 96.65%**

**so in general the best model in our problem is by applying Bagging strategy with SVM as a base estimator with number of estimator = 10 and learning rate = 0.1 .**

**References:**

**[1]** [**https://www.statology.org/sklearn-classification-report/**](https://www.statology.org/sklearn-classification-report/)

**[2]** [**https://scikit-learn.org/stable/modules/tree.html**](https://scikit-learn.org/stable/modules/tree.html)

**[3]** [**https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html**](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html)

**[4]** [**https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html**](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html)

**[5]** [**https://xgboost.readthedocs.io/en/latest/index.html**](https://xgboost.readthedocs.io/en/latest/index.html)

**[6]** [**https://xgboost.readthedocs.io/en/stable/parameter.html**](https://xgboost.readthedocs.io/en/stable/parameter.html)

**[7]** [**https://www.analyticssteps.com/blogs/what-gini-index-and-information-gain-decision-trees**](https://www.analyticssteps.com/blogs/what-gini-index-and-information-gain-decision-trees)

**[8]** [**https://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/**](https://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/)

**[9]** [**https://www.upgrad.com/blog/bagging-vs-boosting/**](https://www.upgrad.com/blog/bagging-vs-boosting/)