

Assignment 4

DECISION TREE AND ENSEMBLE LEARNING

Hagar Hesham Kamel Ismail Negm
Ekhlās Soliman Khlil Mosa

1 PART 1: NUMERICAL QUESTIONS

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Cloudy	Hot	High	Weak	No
Sunny	Hot	High	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Cloudy	Mild	High	Strong	Yes
Rainy	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Rainy	Mild	High	Weak	Yes
Sunny	Hot	High	Strong	No
Cloudy	Hot	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

1.1 DECISION TREE USING GINI INDEX

Gini Index of Weather (F1):

$$\text{Gini(Cloudy)} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$\text{Gini(Sunny)} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$\text{Gini(Rainy)} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = \frac{3}{8}$$

$$\text{Gini(F1)} = \frac{3}{10} \times \frac{4}{9} + \frac{3}{10} \times \frac{4}{9} + \frac{4}{10} \times \frac{3}{8} = \frac{5}{12} = 0.417$$

Gini Index of Temperature (F2):

$$\text{Gini(Hot)} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Mild)} = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$\text{Gini(Cool)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F2)} = \frac{4}{10} \times \frac{1}{2} + \frac{5}{10} \times \frac{12}{25} + \frac{1}{10} \times 0 = \frac{11}{25} = 0.44$$

Gini Index of Humidity (F3):

$$\text{Gini(High)} = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = \frac{24}{49}$$

$$\text{Gini(Normal)} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$\text{Gini(F3)} = \frac{7}{10} \times \frac{24}{49} + \frac{3}{10} \times \frac{4}{9} = \frac{10}{21} = 0.476$$

Gini Index of Wind (F4):

$$\text{Gini(Weak)} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = \frac{3}{8}$$

$$\text{Gini(Strong)} = 1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2 = \frac{4}{9}$$

$$\text{Gini(F4)} = \frac{4}{10} \times \frac{3}{8} + \frac{6}{10} \times \frac{4}{9} = 0.417$$

The root of the decision tree will be Weather since it is the leftmost feature with the lowest Gini index of 0.417.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Cloudy	Hot	High	Weak	No
Cloudy	Mild	High	Strong	Yes
Cloudy	Hot	Normal	Weak	Yes

Gini Index of Temperature (F2) for Cloudy Weather (F1):

$$\text{Gini(Hot)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Mild)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F2)} = \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 = \frac{1}{3} = 0.333$$

Gini Index of Humidity (F3) for Cloudy Weather (F1):

$$\text{Gini(High)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Normal)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F3)} = \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 = \frac{1}{3} = 0.333$$

Gini Index of Wind (F4) for Cloudy Weather (F1):

$$\text{Gini(Weak)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Strong)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F4)} = \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 = \frac{1}{3} = 0.333$$

The leftmost branch of the root node will be Temperature since it is the leftmost feature with the lowest Gini index of 0.333. It can be observed that for Mild Temperature and Cloudy Weather the Hiking class label is automatically Yes. Hence, this is a leaf node.

Gini Index of Humidity (F3) for Hot Temperature (F2) and Cloudy Weather (F1):

$$\text{Gini(High)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(Normal)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F3)} = \frac{1}{2} \times 0 + \frac{1}{2} \times 0 = 0$$

Gini Index of Wind (F4) for Hot Temperature (F2) and Cloudy Weather (F1):

$$\text{Gini(Weak)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(F4)} = \frac{2}{2} \times \frac{1}{2} = \frac{1}{2} = 0.5$$

The leftmost branch of Hot Temperature and Cloudy Weather will be Humidity since it has the lowest Gini index of 0. Its branches will be leaf nodes since it is a pure node – No for High Humidity and Yes for Normal Humidity.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Sunny	Hot	High	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Hot	High	Strong	No

Gini Index of Temperature (F2) for Sunny Weather (F1):

$$\text{Gini(Hot)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Mild)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F2)} = \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 = \frac{1}{3} = 0.333$$

Gini Index of Humidity (F3) for Sunny Weather (F1):

$$\text{Gini(High)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(Normal)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F3)} = \frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 = \frac{1}{3} = 0.333$$

Gini Index of Wind (F4) for Sunny Weather (F1):

$$\text{Gini(Weak)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(Strong)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(F4)} = \frac{1}{3} \times 0 + \frac{2}{3} \times \frac{1}{2} = \frac{1}{3} = 0.333$$

The middle branch of the root node will be Temperature since it is the leftmost feature with the lowest Gini index of 0.333. It can be observed that for Mild Temperature and Sunny Weather the Hiking class label is automatically Yes. Hence, this is a leaf node.

Gini Index of Humidity (F3) for Hot Temperature (F2) and Sunny Weather (F1):

$$\text{Gini(High)} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$\text{Gini(F3)} = \frac{2}{2} \times \frac{1}{2} = \frac{1}{2} = 0.5$$

Gini Index of Wind (F4) for Hot Temperature (F2) and Sunny Weather (F1):

$$\text{Gini(Weak)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(Strong)} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini(F4)} = \frac{1}{2} \times 0 + \frac{1}{2} \times 0 = 0$$

The leftmost branch of Hot Temperature and Sunny Weather will be Wind since it has the lowest Gini index of 0. Its branches will be leaf nodes since it is a pure node – Yes for Weak Wind and No for Strong Wind.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Rainy	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Rainy	Mild	High	Weak	Yes
Rainy	Mild	High	Strong	No

Gini Index of Temperature (F2) for Sunny Weather (F1):

$$\text{Gini}(\text{Mild}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = \frac{4}{9}$$

$$\text{Gini}(\text{Cool}) = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini}(\text{F2}) = \frac{3}{4} \times \frac{4}{9} + \frac{1}{4} \times 0 = \frac{1}{3} = 0.333$$

Gini Index of Humidity (F3) for Sunny Weather (F1):

$$\text{Gini}(\text{High}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = \frac{4}{9}$$

$$\text{Gini}(\text{Normal}) = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini}(\text{F3}) = \frac{3}{4} \times \frac{4}{9} + \frac{1}{4} \times 0 = \frac{1}{3} = 0.333$$

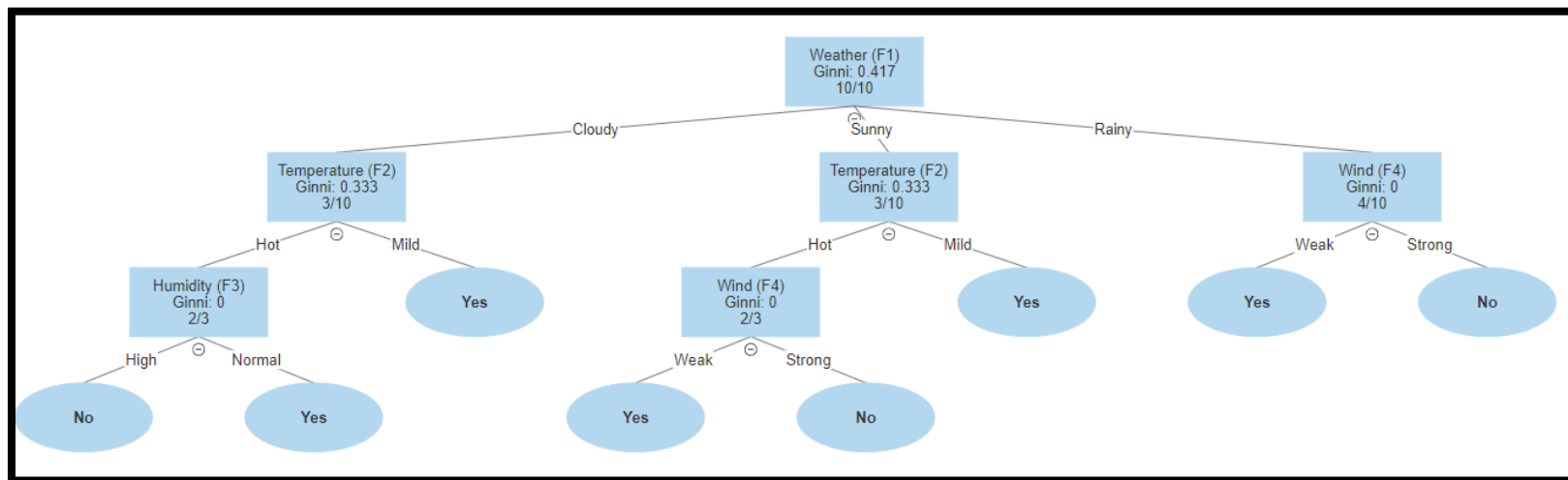
Gini Index of Wind (F4) for Sunny Weather (F1):

$$\text{Gini}(\text{Weak}) = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$\text{Gini}(\text{Strong}) = 1 - \left(\frac{3}{3}\right)^2 = 0$$

$$\text{Gini}(\text{F4}) = \frac{1}{4} \times 0 + \frac{3}{4} \times 0 = 0$$

The rightmost branch of the root node will be Wind since it has the lowest Gini index of 0. Its branches will be leaf nodes since it is a pure node – Yes for Weak Wind and No for Strong Wind.



1.2 DECISION TREE USING INFORMATION GAIN

$$\text{Entropy(S)} = -\frac{5}{10}\log_2 \frac{5}{10} - \frac{5}{10}\log_2 \frac{5}{10} = 1$$

$$\text{Entropy(Cloudy)} = -\frac{2}{3}\log_2 \frac{2}{3} - \frac{1}{3}\log_2 \frac{1}{3} = 0.918$$

$$\text{Entropy(Sunny)} = -\frac{2}{3}\log_2 \frac{2}{3} - \frac{1}{3}\log_2 \frac{1}{3} = 0.918$$

$$\text{Entropy(Rainy)} = -\frac{1}{4}\log_2 \frac{1}{4} - \frac{3}{4}\log_2 \frac{3}{4} = 0.811$$

$$\text{Gain(S, Weather (F1))} = 1 - \frac{3}{10} \times 0.918 - \frac{3}{10} \times 0.918 - \frac{4}{10} \times 0.811 = 0.1248$$

$$\text{Entropy(Hot)} = -\frac{2}{4}\log_2 \frac{2}{4} - \frac{2}{4}\log_2 \frac{2}{4} = 1$$

$$\text{Entropy(Mild)} = -\frac{3}{5}\log_2 \frac{3}{5} - \frac{2}{5}\log_2 \frac{2}{5} = 0.971$$

$$\text{Entropy(Cool)} = -\frac{1}{1}\log_2 \frac{1}{1} = 0$$

$$\text{Gain(S, Temperature (F2))} = 1 - \frac{4}{10} \times 1 - \frac{5}{10} \times 0.971 - \frac{1}{10} \times 0 = 0.1145$$

$$\text{Entropy(High)} = -\frac{3}{7}\log_2 \frac{3}{7} - \frac{4}{7}\log_2 \frac{4}{7} = 0.985$$

$$\text{Entropy(Normal)} = -\frac{2}{3}\log_2 \frac{2}{3} - \frac{1}{3}\log_2 \frac{1}{3} = 0.918$$

$$\text{Gain(S, Humidity (F3))} = 1 - \frac{7}{10} \times 0.985 - \frac{3}{10} \times 0.918 = 0.0351$$

$$\text{Entropy(Weak)} = -\frac{3}{4}\log_2 \frac{3}{4} - \frac{1}{4}\log_2 \frac{1}{4} = 0.811$$

$$\text{Entropy(Strong)} = -\frac{2}{6}\log_2 \frac{2}{6} - \frac{4}{6}\log_2 \frac{4}{6} = 0.918$$

$$\text{Gain(S, Wind (F4))} = 1 - \frac{4}{10} \times 0.811 - \frac{6}{10} \times 0.918 = 0.1248$$

The root of the decision tree will be Weather since it is the leftmost feature with the highest Information Gain of 0.1248.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Cloudy	Hot	High	Weak	No
Cloudy	Mild	High	Strong	Yes
Cloudy	Hot	Normal	Weak	Yes

$$\text{Entropy(S)} = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

$$\text{Entropy(Hot)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(Mild)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Temperature (F2))} = 0.918 - \frac{2}{3} \times 1 - \frac{1}{3} \times 0 = 0.2513$$

$$\text{Entropy(High)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(Normal)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Humidity (F3))} = 0.918 - \frac{2}{3} \times 1 - \frac{1}{3} \times 0 = 0.2513$$

$$\text{Entropy(Weak)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(Strong)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Wind (F4))} = 0.918 - \frac{2}{3} \times 1 - \frac{1}{3} \times 0 = 0.2513$$

The leftmost branch of the root node will be Temperature since it is the leftmost feature with the highest Information Gain of 0.2513. It can be observed that for Mild Temperature and Cloudy Weather the Hiking class label is automatically Yes. Hence, this is a leaf node.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Cloudy	Hot	High	Weak	No
Cloudy	Hot	Normal	Weak	Yes

$$\text{Entropy(S)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(High)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Entropy(Normal)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Humidity (F3))} = 1 - \frac{1}{2} \times 0 - \frac{1}{2} \times 0 = 1$$

$$\text{Entropy(Weak)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Gain(S, Wind (F4))} = 1 - \frac{2}{2} \times 1 = 0$$

The leftmost branch of Hot Temperature and Cloudy Weather will be Humidity since it has the highest Information Gain of 1. Its branches will be leaf nodes since it is a pure node – No for High Humidity and Yes for Normal Humidity.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Sunny	Hot	High	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Hot	High	Strong	No

$$\text{Entropy(S)} = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

$$\text{Entropy(Hot)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(Mild)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Temperature (F2))} = 0.918 - \frac{2}{3} \times 1 - \frac{1}{3} \times 0 = 0.2513$$

$$\text{Entropy(High)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(Normal)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Humidity (F3))} = 0.918 - \frac{2}{3} \times 1 - \frac{1}{3} \times 0 = 0.2513$$

$$\text{Entropy(Weak)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Entropy(Strong)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Gain(S, Wind (F4))} = 0.918 - \frac{1}{3} \times 0 - \frac{2}{3} \times 1 = 0.2513$$

The middle branch of the root node will be Temperature since it is the leftmost feature with the highest Information Gain of 0.2513. It can be observed that for Mild Temperature and Sunny Weather the Hiking class label is automatically Yes. Hence, this is a leaf node.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Sunny	Hot	High	Weak	Yes
Sunny	Hot	High	Strong	No

$$\text{Entropy(S)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Entropy(High)} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$\text{Gain(S, Humidity (F3))} = 1 - \frac{2}{2} \times 1 = 0$$

$$\text{Entropy(Weak)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Entropy(Strong)} = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain(S, Wind (F4))} = 1 - \frac{1}{2} \times 0 - \frac{1}{2} \times 0 = 1$$

The leftmost branch of Hot Temperature and Sunny Weather will be Wind since it has the highest Information Gain of 1. Its branches will be leaf nodes since it is a pure node – Yes for Weak Wind and No for Strong Wind.

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Label)
Rainy	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Rainy	Mild	High	Weak	Yes
Rainy	Mild	High	Strong	No

$$\text{Entropy}(S) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811$$

$$\text{Entropy}(\text{Mild}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$$

$$\text{Entropy}(\text{Cool}) = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Gain}(S, \text{Temperature (F2)}) = 0.811 - \frac{3}{4} \times 0.918 - \frac{1}{4} \times 0 = 0.1225$$

$$\text{Entropy}(\text{High}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$$

$$\text{Entropy}(\text{Normal}) = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

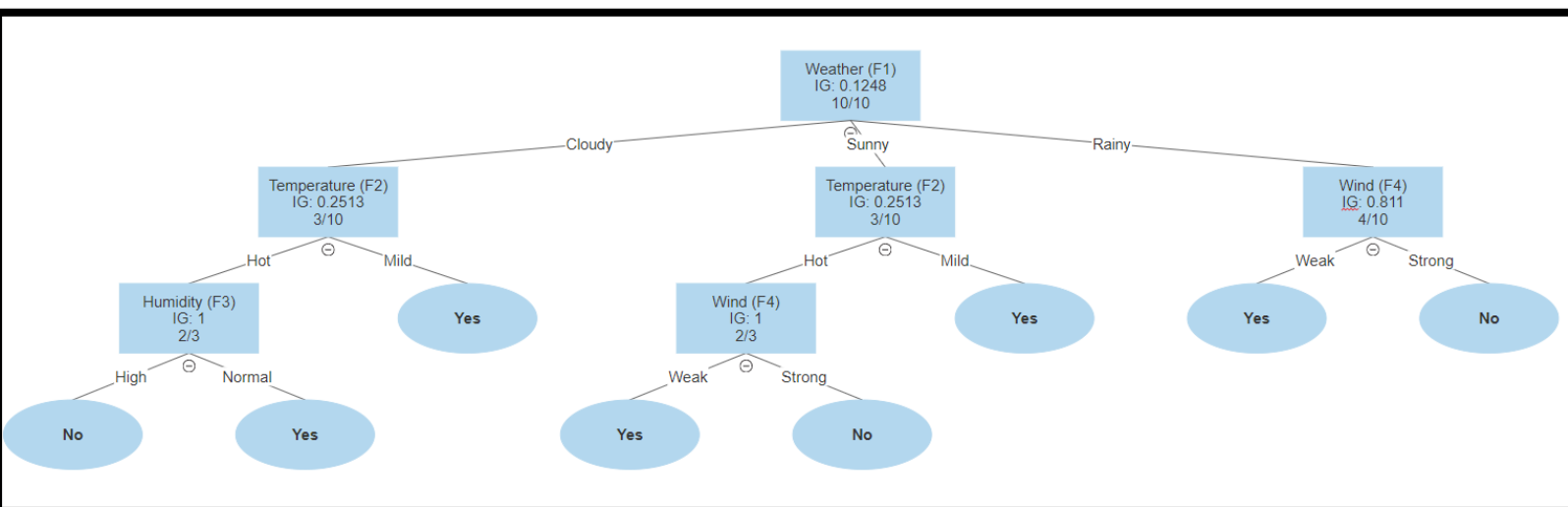
$$\text{Gain}(S, \text{Humidity (F3)}) = 0.811 - \frac{3}{4} \times 0.918 - \frac{1}{4} \times 0 = 0.1225$$

$$\text{Entropy}(\text{Weak}) = -\frac{1}{1}\log_2\frac{1}{1} = 0$$

$$\text{Entropy}(\text{Strong}) = -\frac{3}{3}\log_2\frac{3}{3} = 0$$

$$\text{Gain}(S, \text{Wind (F4)}) = 0.811 - \frac{1}{4} \times 0 - \frac{3}{4} \times 0 = 0.811$$

The rightmost branch of the root node will be Wind since it has the highest Information Gain of 1. Its branches will be leaf nodes since it is a pure node – Yes for Weak Wind and No for Strong Wind.



1.3 COMPARISON BETWEEN GINI INDEX AND INFORMATION GAIN

1.3.1 Gini Index

Advantages:

- Deals with inequality.
- Computationally simple, hence faster calculation.

Disadvantages:

- Dependent on sample size.

1.3.2 Information Gain

Advantages:

- Less frequent classes are assigned less weight

Disadvantages:

- Computationally complex since it makes use of logarithms.
- Prefers splits that result in many small but pure partitions.

2 PART 2: PROGRAMMING QUESTIONS

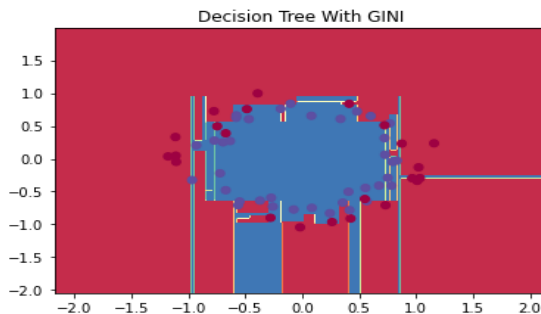
2.1 DECISION TREES

QUESTION 4

Gini

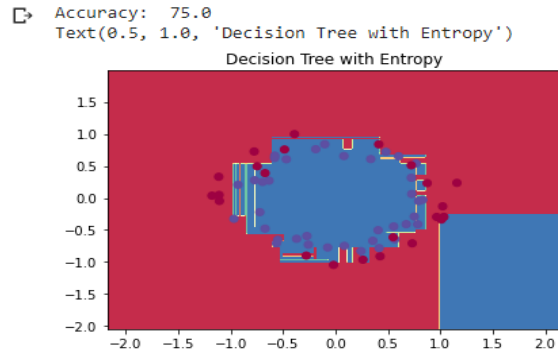
```
▶ clf_gini = DecisionTreeClassifier(criterion='gini', random_state = 0)
  clf_gini.fit(trX_circle, trY_circle)
  yPred_gini = clf_gini.predict(teX_circle)
  print("Accuracy: ", accuracy_score(teY_circle, yPred_gini) * 100)
  plot_decision_boundary(lambda teX_circle: clf_gini.predict(teX_circle))
  plt.title("Decision Tree With GINI")
```

```
☐ Accuracy: 66.66666666666666
  Text(0.5, 1.0, 'Decision Tree With GINI')
```



Entropy

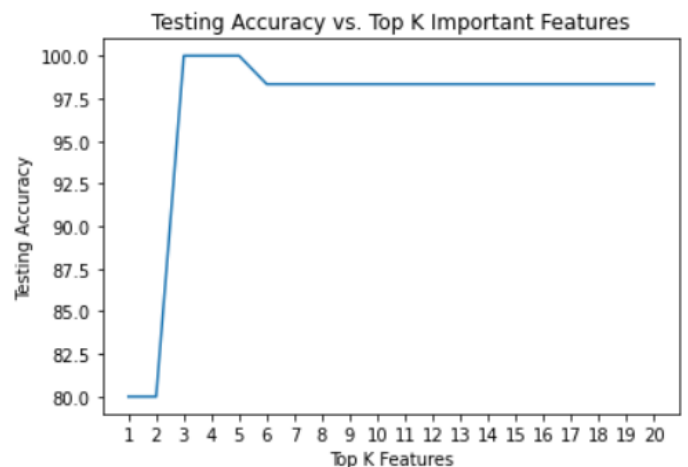
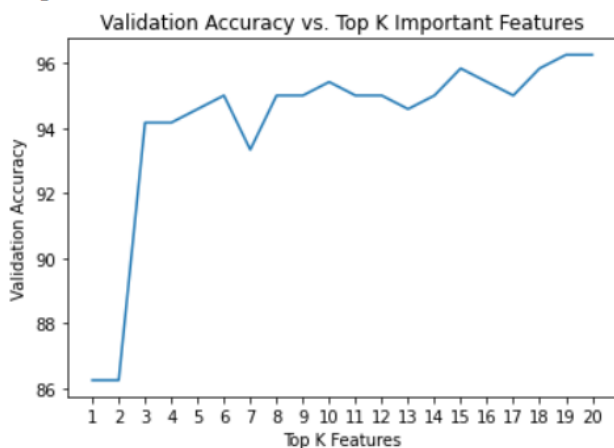
```
1 clf_ent = DecisionTreeClassifier(criterion='entropy', random_state = 0)
2 clf_ent.fit(trX_circle, trY_circle)
3 yPred_ent = clf_ent.predict(teX_circle)
4 print("Accuracy: ", accuracy_score(teY_circle, yPred_ent) * 100)
5 plot_decision_boundary(lambda teX_circle: clf_ent.predict(teX_circle))
6 plt.title("Decision Tree with Entropy")
```



Entropy gives the highest accuracy because its mechanism adds more complexity to the model as it uses log algorithm in splitting decision tree. Entropy is more computationally complex, whereas the Gini index is more computationally simple. Entropy takes values between 0 and 1, while Gini index takes values between 0 and 0.5.

QUESTION 5

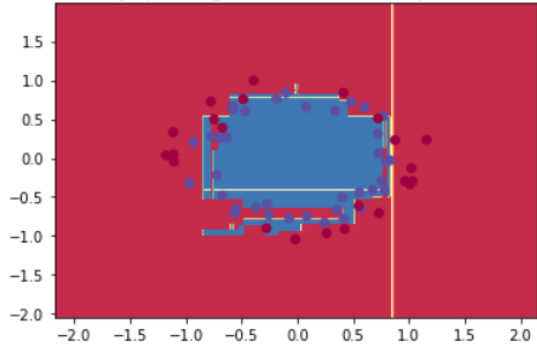
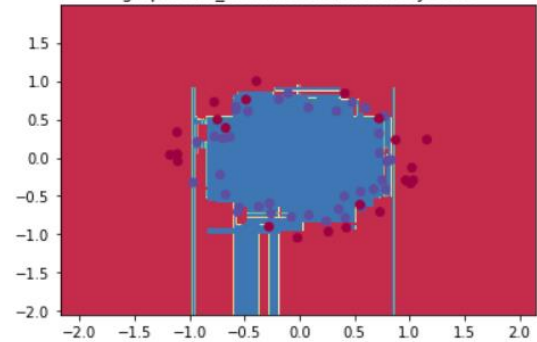
<Figure size 432x288 with 0 Axes>

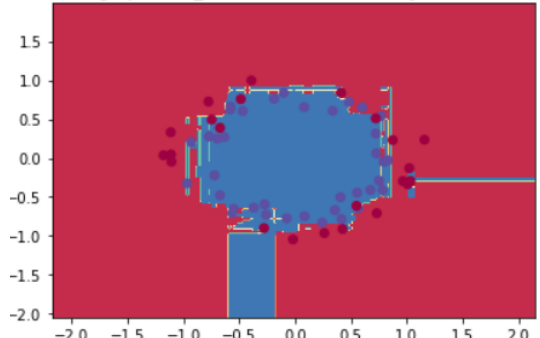
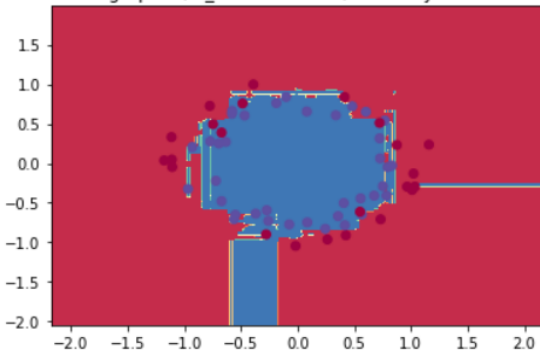


2.2 BAGGING

QUESTION 6

```
1 n_estimators = [2, 5, 15, 20]
2 count = 1
3 for i in n_estimators:
4     randem_model = RandomForestClassifier(n_estimators = i , max_depth=2, random_state=0)
5     randem_model.fit(trX_circle, trY_circle)
6     model_pred = randem_model.predict(teX_circle)
7     randem_acc = accuracy_score(teY_circle, model_pred) * 100
8     print("Accuracy:", accuracy_score(teY_circle, model_pred) * 100)
9     plt.subplots(sharex='col', sharey='row')
10    plot_decision_boundary(lambda trX_circle: randem_model.predict(trX_circle))
11    plt.title("graph: {}, n_estimators: {}, Accuracy: {}".format(count, i , randem_acc))
12    count = count +1
```

N_Estimators	Accuracy and Decision Boundary
2	<p>graph: 1, n_estimators: 2, Accuracy: 60.0</p>  <p>A scatter plot showing data points (blue and red) and a decision boundary (yellow line) for 2 estimators. The plot is titled "graph: 1, n_estimators: 2, Accuracy: 60.0". The x-axis ranges from -2.0 to 2.0, and the y-axis ranges from -2.0 to 1.5. The decision boundary is a simple vertical line at x ≈ 0.8.</p>
5	<p>graph: 2, n_estimators: 5, Accuracy: 75.0</p>  <p>A scatter plot showing data points (blue and red) and a decision boundary (yellow line) for 5 estimators. The plot is titled "graph: 2, n_estimators: 5, Accuracy: 75.0". The x-axis ranges from -2.0 to 2.0, and the y-axis ranges from -2.0 to 1.5. The decision boundary is a more complex, irregular shape that better fits the data points.</p>

15	<p>graph: 3, n_estimators: 15, Accuracy: 71.667</p> 
20	<p>graph: 4, n_estimators: 20, Accuracy: 75.0</p> 

QUESTION 7

Bagging refers to training the same model multiple times on different data sets by using bootstrap (random sampling with replacement from the training set), then combines the predictions after fit the models, so it reduces variance and mitigates the overfitting. The variance is reduced because the training sets in bagging are completely independent, hence sensitivity to individual datapoints is reduced.

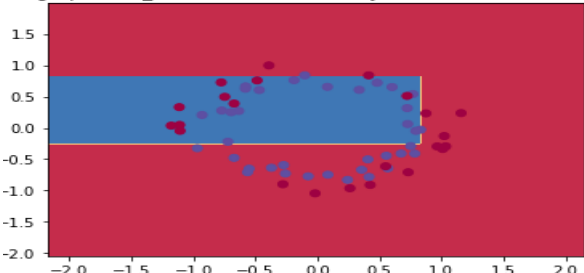
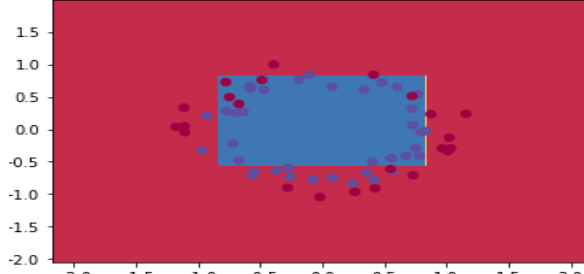
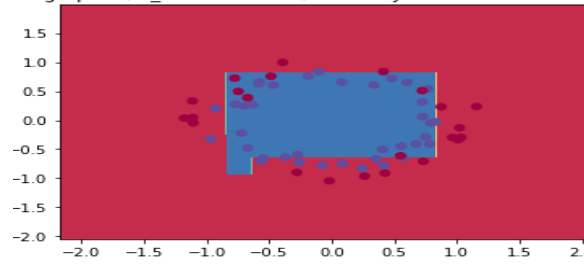
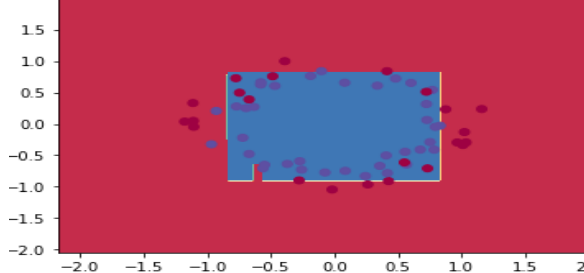
2.3 RANDOM FOREST

QUESTION 8

```

n_estimators = [2, 5, 15, 20]
count = 1
for i in n_estimators:
    randem_model = RandomForestClassifier(n_estimators = i , max_depth=2, random_state=0)
    randem_model.fit(trX_circle, trY_circle)
    model_pred = randem_model.predict(teX_circle)
    randem_acc = accuracy_score(teY_circle, model_pred) * 100
    print("Accuracy:", accuracy_score(teY_circle, model_pred) * 100)
    plt.subplots(sharex='col', sharey='row')
    plot_decision_boundary(lambda trX_circle: randem_model.predict(trX_circle))
    plt.title("graph: {}, n_estimators: {}, Accuracy: {}".format(count, i , randem_acc))
    count = count +1

```

N_Estimator	Accuracy and Decision Boundary
2	<p>graph: 1, n_estimators: 2, Accuracy: 51.6666666666667</p> 
5	<p>graph: 2, n_estimators: 5, Accuracy: 66.6666666666667</p> 
15	<p>graph: 3, n_estimators: 15, Accuracy: 66.6666666666667</p> 
20	<p>graph: 4, n_estimators: 20, Accuracy: 78.3333333333333</p> 

QUESTION 9

When comparing the results between bagging and random forest it's different because of their techniques, as bagging uses all the feature in splitting however random forest uses only a subset of features is selected at random out of the total. We can also see that as the number of estimators increase in the random forest model the accuracy increases consistently, but with the bagging model the accuracy drops at 15 estimators.

2.4 BOOSTING

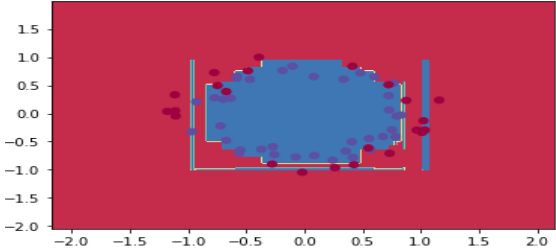
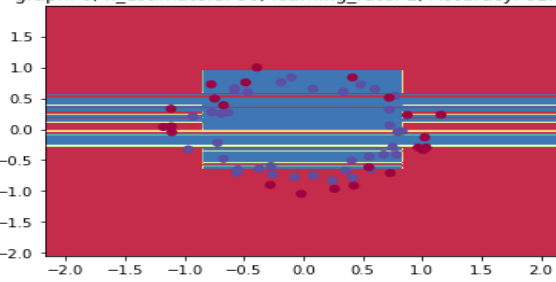
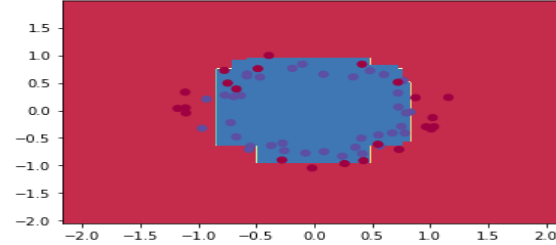
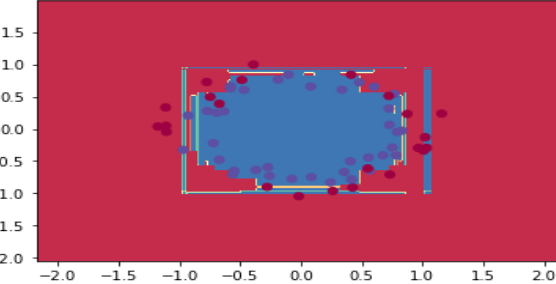

QUESTION 10

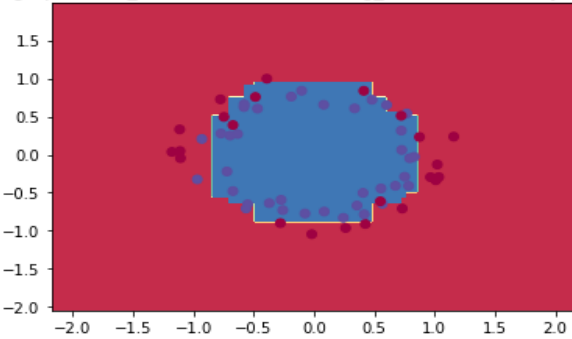
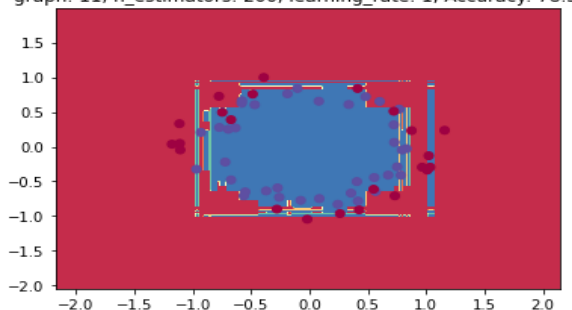
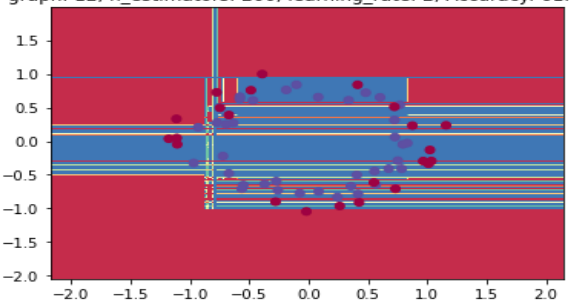
```

n_estimators_boosting = [10, 50, 100, 200]
learning_rate_boosting = [0.1, 1, 2]
count = 1
for i in n_estimators_boosting:
    for j in learning_rate_boosting:
        boost_model = AdaBoostClassifier(n_estimators=i, learning_rate=j, random_state=0)
        boost_model.fit(trX_circle, trY_circle)
        boost_pred = boost_model.predict(teX_circle)
        acc_boost = round(accuracy_score(teY_circle, boost_pred) * 100, 3)
        print("Accuracy:", acc_boost)
        plt.subplots(sharex='col', sharey='row')
        plot_decision_boundary(lambda trX_circle: boost_model.predict(trX_circle))
        plt.title("graph: {}, n_estimators: {}, learning_rate: {}".format(count, i, j, acc_boost))
        count = count + 1

```

N_Estimators, Learning Rate	Accuracy and Decision Boundary
ne=10; lr=0.1	<p>graph: 1, n_estimators: 10, learning_rate: 0.1, Accuracy: 66.667</p>
ne=10; lr=1	<p>graph: 2, n_estimators: 10, learning_rate: 1, Accuracy: 65.0</p>
ne=10; lr=2	<p>graph: 3, n_estimators: 10, learning_rate: 2, Accuracy: 66.667</p>
ne=50; lr=0.1	<p>graph: 4, n_estimators: 50, learning_rate: 0.1, Accuracy: 65.0</p>

ne=50; lr=1	<p>graph: 5, n_estimators: 50, learning_rate: 1, Accuracy: 83.333</p> 
ne=50; lr=2	<p>graph: 6, n_estimators: 50, learning_rate: 2, Accuracy: 61.667</p> 
ne=100; lr=0.1	<p>graph: 7, n_estimators: 100, learning_rate: 0.1, Accuracy: 73.333</p> 
ne=100; lr=1	<p>graph: 8, n_estimators: 100, learning_rate: 1, Accuracy: 80.0</p> 
ne=100; lr=2	<p>graph: 9, n_estimators: 100, learning_rate: 2, Accuracy: 56.667</p> 

<p>ne=200; lr=0.1</p>	<p>graph: 10, n_estimators: 200, learning_rate: 0.1, Accuracy: 80.0</p> 
<p>ne=200; lr=1</p>	<p>graph: 11, n_estimators: 200, learning_rate: 1, Accuracy: 78.333</p> 
<p>ne=200; lr=2</p>	<p>graph: 12, n_estimators: 200, learning_rate: 2, Accuracy: 61.667</p> 

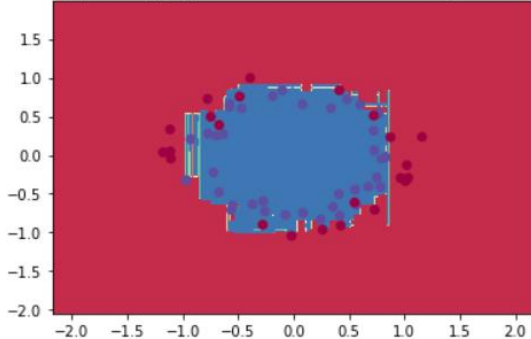
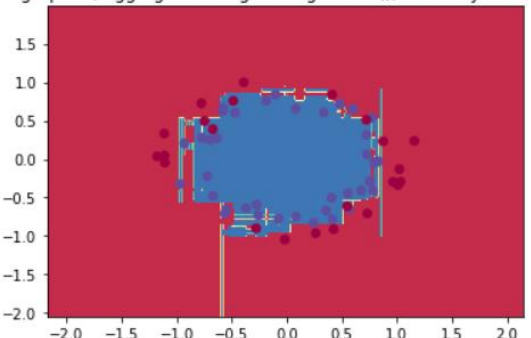
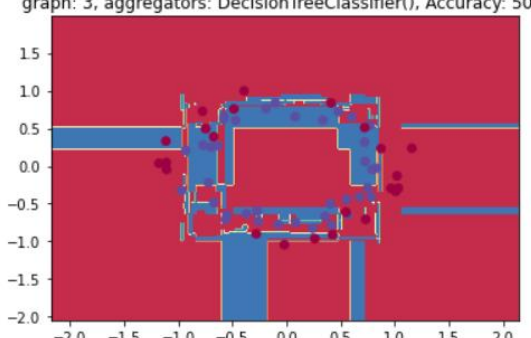
2.5 STACKING

QUESTION 11

```

1 estimators = [('ent_clf' , DecisionTreeClassifier(criterion='entropy', random_state = 0)) ,
2               ('bagging_clf' , BaggingClassifier(clone(dt), n_estimators=5, n_jobs=-1, random_state=0)),
3               ('randem_clf' , RandomForestClassifier(n_estimators = 20 , max_depth=2, random_state=0)),
4               ('boost_clf' , AdaBoostClassifier(n_estimators=50, learning_rate=1, random_state=0))
5               ]
6 aggregators = [GaussianNB() , LogisticRegression() , DecisionTreeClassifier()]
7 count = 1
8 for agg in aggregators:
9     clf_stacking = StackingClassifier( estimators=estimators, final_estimator=agg)
10    clf_stacking.fit(trX_circle, trY_circle)
11    stacking_pred = clf_stacking.predict(teX_circle)
12    acc_stacking = round(accuracy_score(teY_circle, stacking_pred) * 100, 3)
13    print("Accuracy:",acc_stacking)
14    plt.subplots(sharex='col', sharey='row')
15    plot_decision_boundary(lambda teX_circle: clf_stacking.predict(teX_circle))
16    plt.title("graph: {}, aggregators: {}, Accuracy: {}".format(count, agg , acc_stacking))
17    count = count + 1

```

Aggregators	Accuracy and Decision Boundary
Naive Bayes	<p>graph: 1, aggregators: GaussianNB(), Accuracy: 83.333</p>  <p>A scatter plot with x and y axes ranging from -2.0 to 2.0. The plot shows a central cluster of blue data points surrounded by a ring of red data points. The decision boundary, shown as a blue line, is a smooth, roughly elliptical shape that encloses the blue cluster. The background is filled with a solid red color, representing the predicted class for points outside the boundary.</p>
Logistic Regression	<p>graph: 2, aggregators: LogisticRegression(), Accuracy: 73.333</p>  <p>A scatter plot similar to the one above, but with a different decision boundary. The blue line is smooth but has a more complex, slightly irregular shape compared to the Naive Bayes plot. It still encloses the central blue cluster, but with some indentations and protrusions. The background is red.</p>
Decision Tree	<p>graph: 3, aggregators: DecisionTreeClassifier(), Accuracy: 50.0</p>  <p>A scatter plot with a highly complex and non-smooth decision boundary. The blue line consists of many horizontal and vertical segments, creating a jagged, step-like shape that follows the local distribution of the blue data points. This results in a very irregular boundary that captures the noise in the training data. The background is red.</p>

2.6 CONCLUSION

In the numerical questions, we learnt the process of how a decision tree is built and how the most important features are selected incrementally using different splitting criterion such as the Gini Index and the Information Gain.

In the programming questions, we learnt how the different splitting criterion affect the accuracy of the decision tree model. It was observed that the Entropy resulted in a higher accuracy than the Gini Index – 75% and 66.7% respectively. When we plotted the top k features against the validation and testing accuracies, we can see that after the top 5 important features, the testing accuracy slightly drops and then becomes constant.

We also learnt how the different kinds of ensemble models work, such as bagging and boosting models. Bagging models can either be heterogenous or homogenous such as random forests which have one type of base estimators. The base estimators of bagging models work in parallel using different samples of the dataset. However, in boosting models, the models work serially on the whole dataset, each model improving on the previous one. Finally, we implemented a stacking model using our fine-tuned base estimators, and 3 types of models as aggregators – Gaussian Naïve Bayes, Logistic Regression and Decision Tree Classifier.