ELG5255[EG]: Applied Machine Learning

Assignment 4

Decision Tree and Ensemble Methods

Group: 25

The report will go as follows: #) for steps to perform the intended tasks including screenshots of the code we used to tackle this question along with any description needed or relevant figures.

Part #1

Numerical Questions

Q1)

We calculate the Gini for each feature to choose the best split. Hence we calculate first the gini of leaves for each feature then the total gini.

In 1st iteration →

G_total_F1 = **0.416**

G_total_F2 = 0.444

G_total_F3 = 0.475

G_total_F4 = **0.416**

As F1 and F4 have same gini coefficients, we choose F4 for simple split.

Iteration 1:

- F1: Weather, F2: Temperature, F3: Humidity, F4: Wind
- Gini_leaves $= 1-(P(Yes))^2-(P(No))^2$
- Total Gini of node $= \Sigma$ weight of leaf $*$ Gini_leaf
  ↳ $\frac{\#\text{ instance in leaf}}{\#\text{ total instance in Node}}$
- iterat (1): Calculate Gini for each feature & take the lowest

F1

F1 — #10

Cloudy #3

Hiking Yes	Hiking No
2	1

Sunny #3

Hiking Yes	Hiking No
1	2

Rainy #4

Hiking Yes	Hiking No
1	3

$G_1 = 1-(\frac{2}{3})^2-(\frac{1}{3})^2 = 0.444$

$G_2 = 1-(\frac{2}{3})^2-(\frac{1}{3})^2 = 0.44$

$G_3 = 1-(\frac{1}{4})^2-(\frac{3}{4})^2 = 0.375$

$\text{Gini} = \frac{3}{10}*0.444 + \frac{3}{10}*0.444 + \frac{4}{10}*0.375 = \boxed{0.416}$

F2

F2 — #10

Hot #4

Hiking Yes	Hiking No
2	2

Mild #5

Hiking Yes	Hiking No
3	2

Cool #1

Hiking Yes	Hiking No
0	1

$G_1 = 1-(\frac{1}{2})^2-(\frac{1}{2})^2 = 0.5$

$G_2 = 1-(\frac{3}{5})^2-(\frac{2}{5})^2 = 0.48$

$G_3 = 0$ (no impurity) $= 0$

$\text{Gini} = \frac{4}{10}*0.5 + \frac{5}{10}*0.48 + \frac{1}{10}*0 = \boxed{0.444}$

F3

F3 — #10

High #7

Hiking Yes	Hiking No
3	4

Normal #3

Hiking Yes	Hiking No
2	1

$G_1 = 1-(\frac{3}{7})^2-(\frac{4}{7})^2 = 0.489$

$G = 1-(\frac{2}{3})^2-(\frac{1}{3})^2 = 0.444$

$\text{Gini} = \frac{7}{10}*0.489 + \frac{3}{10}*0.444 = \boxed{0.475}$

F4

F4 — #10

Weak #4

Hiking Yes	Hiking No
3	1

Strong #6

Hiking Yes	Hiking No
2	4

$G = 1-(\frac{3}{4})^2-(\frac{1}{4})^2 = 0.375$

$G = 1-(\frac{2}{6})^2-(\frac{4}{6})^2 = 0.444$

$\text{Gini} = \frac{4}{10}*0.375 + \frac{6}{10}*0.444 = \boxed{0.4164}$

We can find there're more impurities in both leaves so both leaves need further split and hence we try gini with the remaining 3 features in both leaves.

Iteration 2:

(2)

The 4 gini values are: f1 = f4 = 0.4164

f2 = 0.444 f3 = 0.475

∴ we choose for f1 or f4 as they've same gini; for simplicity of sketching choose $\boxed{f4}$

. iterate (2) → for feature (4); each leaf in it we shall check if it has impurities. Since both leaves have impurities so we shall calculate the gini for the other 3 features in accordance with feature (1) & choose lowest

(f4) f4

weak → Hiking: Yes 3, No 1 ← impurity

Strong → Hiking: Yes 2, No 4 ← impurity

weak branch (# 4)

(f1) f1

	(2) Cloudy	Sunny (1)	Rainy (1)
Yes	1	1	1
No	1	0	0
	↳ G = 0.5	↳ G = 0	↳ G = 0

total G = 2/4 * 0.5 + 0 = $\boxed{0.25}$

(f2) f2

	(3) hot	mild (1)	Cool (0)
Yes	2	1	0
No	1	0	0
	↳ $G = 1-(\frac{1}{3})^2-(\frac{2}{3})^2$ = 0.444	↳ G = 0	→ G = 0

total G = 3/4 × 0.444 + 0 = $\boxed{0.333}$

(f3) f3

	(3) high	Normal (1)
Yes	2	1
No	1	0
	↳ $G = 1-(\frac{2}{3})^2-(\frac{1}{3})^2 = 0.444$	↳ G = 0

G = 0.444 * 3/4 + 0 = $\boxed{0.333}$

Strong branch (# 6)

(f1) f1

	(1) Cloudy	Sunny (2)	Rainy (3)
Yes	1	1	0
No	0	1	3
	↳ G = 0	↳ G = 0.5	↳ G = 0.

(G) = 2/6 * 0.5 = $\boxed{0.167}$

(f2) f2

	(1) hot	mild (4)	Cool (1)
Yes	0	2	0
No	1	2	1
	↳ G = 0	↳ G = 0.5	↳ G = 0

total G = 4/6 * 0.5 = $\boxed{0.333}$

f3

	(4) high	Normal (2)
Yes	1	1
No	3	1
	↳ $G = 1-(\frac{1}{4})^2-(\frac{3}{4})^2 = \frac{3}{8}$	↳ G = 0.5

total G = 4/6 × 3/8 + 2/6 * 0.5 = $\boxed{0.4167}$

In 2nd iteration →

Wind (F4)	**Weak**	**Strong**
G_total_F1	**0.25**	**0.167**
G_total_F2	0.333	0.333
G_total_F3	0.333	0.4167

Here we can observe that for weak wind the best subsequent split will be by the weather (F1) and same goes for the strong wind as they have the least total gini.

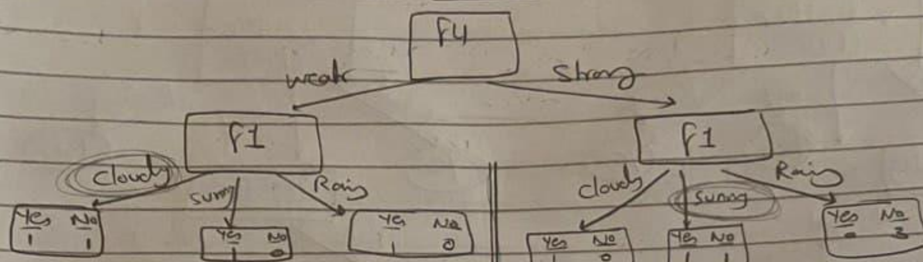
However, there still few impurities after this split so we need one more iteration for splitting.

Iteration 3:

3

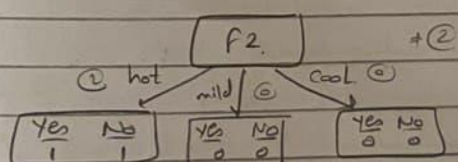
→ for • weak wind (f4) → G_f1 = 0.25 ✓, G_f2 = 0.333, G_f3 = 0.333 ∴ we'll Split by f1

• Strong wind (f4) → G_f1 = 0.167 ✓, G_f2 = 0.333, G_f3 = 0.4167 ∴ we'll Split by f1



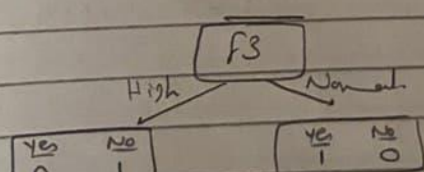
→ As we observe Here; Cloudy weak-wind weather Carry impurity (Try gini)

• f4 = weak && f1 = Cloudy



$G = 1-(\frac{1}{2})^2-(\frac{1}{2})^2 = 0.5$

total $G = \frac{2}{2} * 0.5 + 0 = \boxed{0.5}$



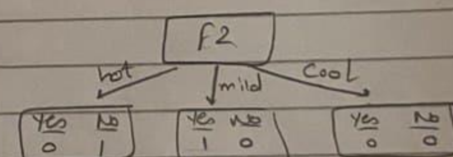
$G=0$ $G=0$

$\boxed{G=0}$

* G_f2 = 0.5
G_f3 = $\boxed{0}$ ✓ Split by f3

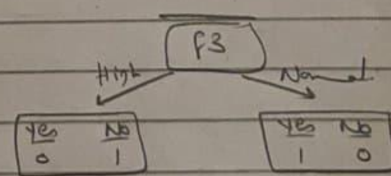
→ Here the Sunny Strong-wind weather has impurity to Split further

• f4 = Strong && f1 = Sunny



$G=0$ $G=0$

total $G = 0+0+0 = \boxed{0}$



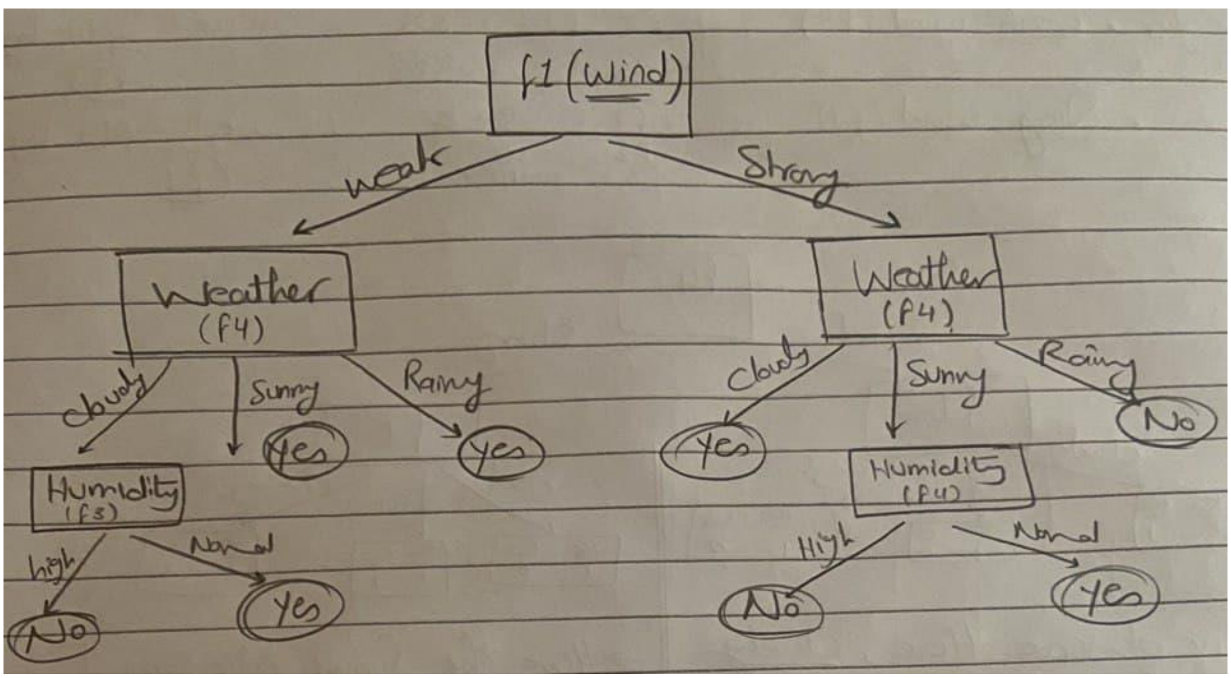
$G=0$ $G=0$

total $G = 0+0 = \boxed{0}$

* G_f2 = G_f3 = 0
→ for Simple Sketch → f3

Wind (F4) & Weather (F1)	Weak & cloudy	Strong & Sunny
G_total_F2	0.5	**0**
G_total_F3	**0**	**0**

For the weak wind and cloudy weather, we can see that F3 (Humidity) yields no impurities.
Also, for the strong wind and sunny weather, we can observe that both features yield no impurities.
We choose F3 for both splits and this is the final tree that we get.



Q2)

$\boxed{1}$

$$H(T) = -\tfrac{5}{10}\log_2\left(\tfrac{5}{10}\right) - \tfrac{5}{10}\log_2\left(\tfrac{5}{10}\right) = 1$$

$$H(T|\text{weather}) = \tfrac{3}{10}\left(-\tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) - \tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right)\right) + \tfrac{3}{10}\left(-\tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) - \tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right)\right) + \tfrac{4}{10}\left(-\tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right) - \tfrac{3}{4}\log_2\left(\tfrac{3}{4}\right)\right) \simeq 0.275 + 0.275 + 0.325 \simeq 0.875$$

$$H(T|\text{Temp}) = \tfrac{4}{10}\left(-\tfrac{2}{4}\log_2\left(\tfrac{2}{4}\right) - \tfrac{2}{4}\log_2\left(\tfrac{2}{4}\right)\right) + \tfrac{5}{10}\left(-\tfrac{3}{5}\log_2\left(\tfrac{3}{5}\right) - \tfrac{2}{5}\log_2\left(\tfrac{2}{5}\right)\right) + \tfrac{1}{10}\left(-\tfrac{1}{1}\log_2(1) - \tfrac{0}{1}\log_2(0)\right) = 0.4 + 0.485 + 0 \simeq 0.885$$

$$H(T|\text{Hum}) = \tfrac{7}{10}\left(-\tfrac{3}{7}\log_2\left(\tfrac{3}{7}\right) - \tfrac{4}{7}\log_2\left(\tfrac{4}{7}\right)\right) + \tfrac{3}{10}\left(-\tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) - \tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right)\right) \simeq 0.690 + 0.275 \simeq 0.965$$

$$H(T|\text{wind}) = \tfrac{4}{10}\left(-\tfrac{3}{4}\log_2\left(\tfrac{3}{4}\right) - \tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right)\right) + \tfrac{6}{10}\left(-\tfrac{2}{6}\log_2\left(\tfrac{2}{6}\right) - \tfrac{4}{6}\log_2\left(\tfrac{4}{6}\right)\right)$$

$$\simeq 0.325 + 0.551 \simeq 0.876$$

$IG(T|\text{weather}) = 1 - 0.875 = \boxed{0.125} \rightarrow$ highest IG

$IG(T|\text{Temp}) = 1 - 0.885 = 0.115$

$IG(T|\text{Hum}) = 1 - 0.965 = 0.035$

$IG(T|\text{wind}) = 1 - 0.876 = 0.124$

→ cloudy

Temperature	Humidity	wind	Hiking
Hot	High	weak	No
Hot	Normal	weak	yes
Mild	High	strong	yes

$H(T) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) \simeq 0.918$

$H(T|Temp) = \frac{2}{3}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) \simeq 0.667$

$H(T|Hum) = \frac{2}{3}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) \simeq 0.667$

$H(T|wind) = \frac{2}{3}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) \simeq 0.667$

$IG(T|Temp) = IG(T|Hum) = IG(T|wind$

$\simeq 0.918 - 0.667 \simeq 0.251$

since all information gain are equal Then I will choose The first predictor (Temperature)

(3)

→ Sunny

Temperature	Humidity	wind	Hiking
HoT	High	weak	Yes
HoT	High	sTrong	No
Mild	Normal	sTrong	Yes

$H(T) = -\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3}) \simeq 0.918$

$H(T|Temp) = \frac{2}{3}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) \simeq 0.667$

$H(T|Hum) = \frac{2}{3}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) \simeq 0.667$

$H(T|wind) = \frac{2}{3}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) \simeq 0.667$

$IG(T|Temp) = H(T|Hum) = H(T|wind) \simeq 0.918 - 0.667 \simeq 0.251$

since all IG are equal Then I'll choose The first predictor (Temperature)

→ Rainy

Temperature	Humidity	wind	Hiking
Mild	High	strong	No
mild	High	weak	Yes
cool	Normal	strong	No

$H(T) = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) \simeq 0.918$

$H(T|Temp) = \frac{2}{3}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) \simeq 0.667$

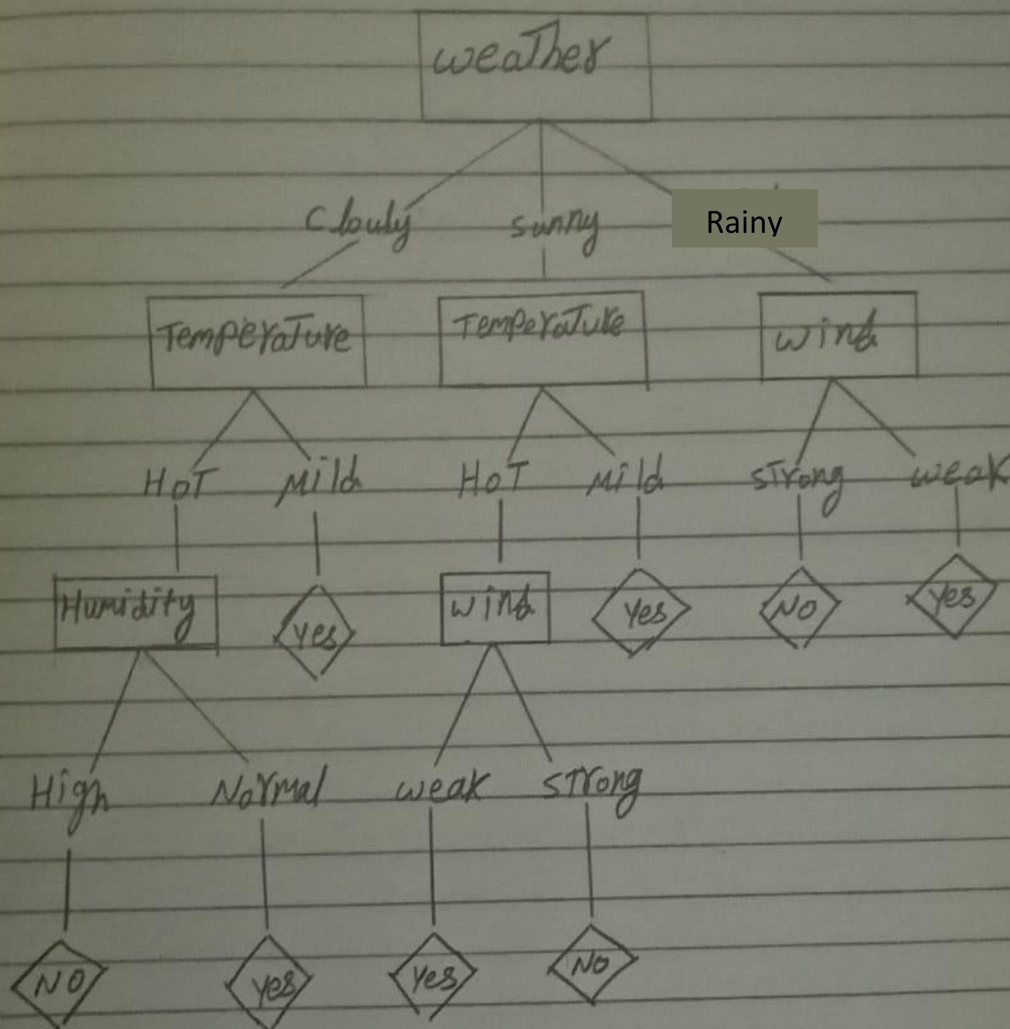
$H(T|Hum) = \frac{2}{3}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) \simeq 0.667$

$H(T|wind) = \frac{2}{3}(0 - \frac{2}{2}\log_2(\frac{2}{2})) + \frac{1}{3}(-\frac{1}{1}\log_2(1) - 0) = 0$

$IG(T|Temp) = IG(T|Hum) \simeq 0.251$

$IG(T|wind) = 0.918 - 0 = \boxed{0.918}$ ⇒ highest IG

5


weather
Clouly
sunny
Rainy
Temperature
Temperature
wind
HoT
Mild
HoT
Mild
STrong
weak
Humidity
Yes
wind
Yes
NO
Yes
High
Normal
weak
STrong
NO
Yes
Yes
NO

Q3) [1]

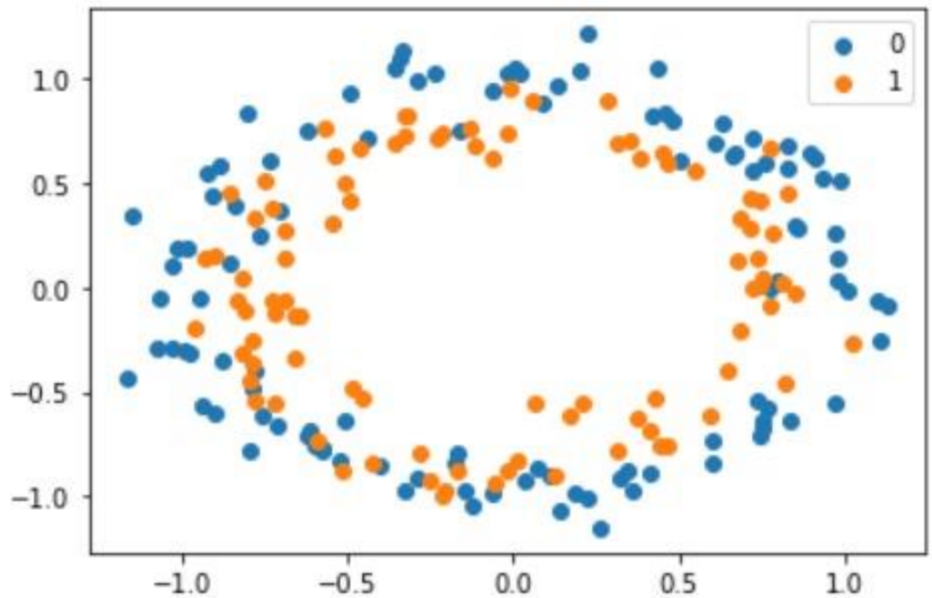
	Gini Index	**Information Gain**
Advantages	Easy to implement	Helpful in exploratory analysis
	Computational non-extensive	Sometimes Outperforms gini in data imbalance
		Better in exponential data distributions
Disasdvantage	Doesn't perform well at some conditions	Computational extensive (Log)
		prefer splits that result in large number of partitions, each being small but pure.

Part #2

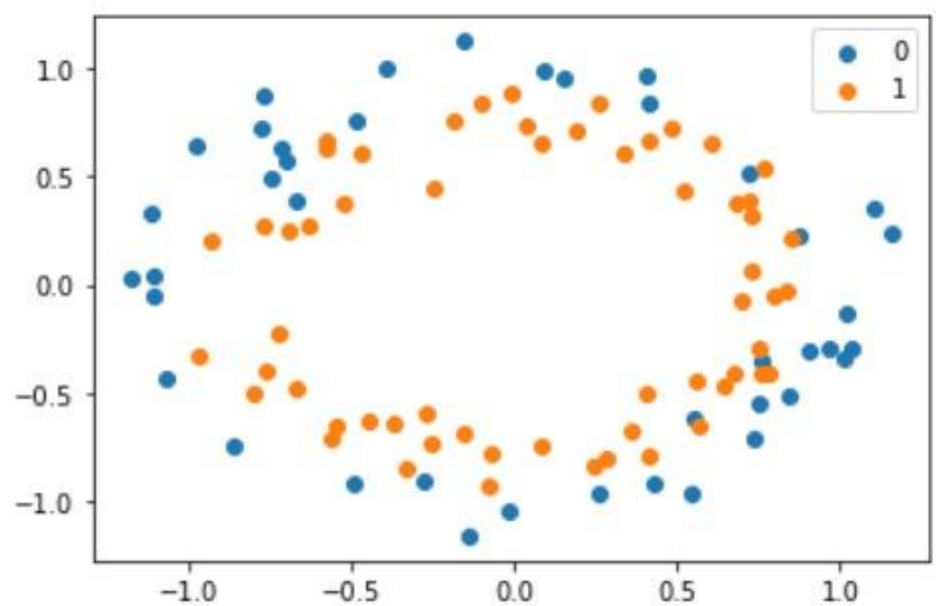
Programming Questions

Q4.1)

Train Data:

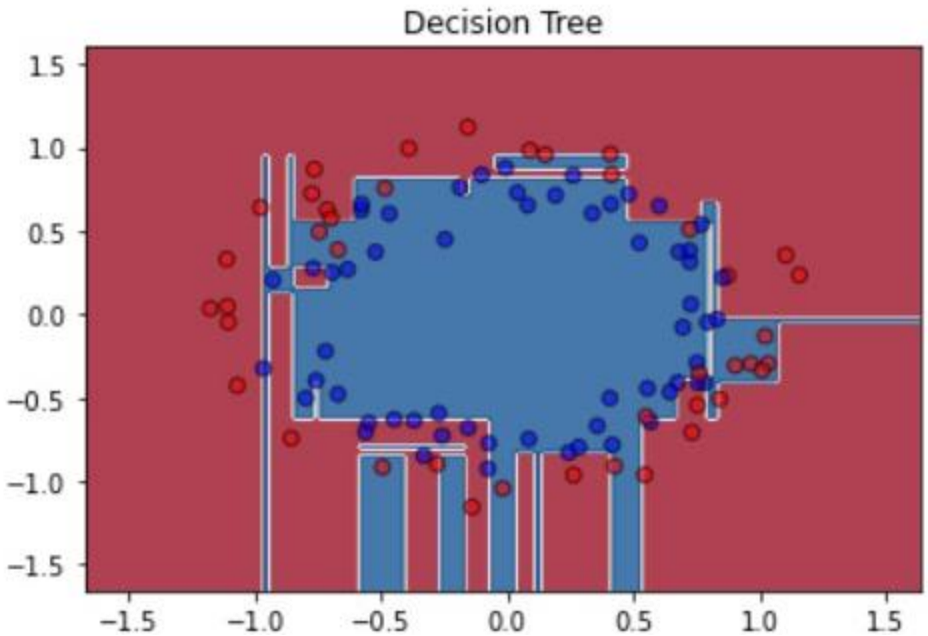


Test Data:



Decision Boundary:

```
The Accuracy =  0.6060606060606061
```



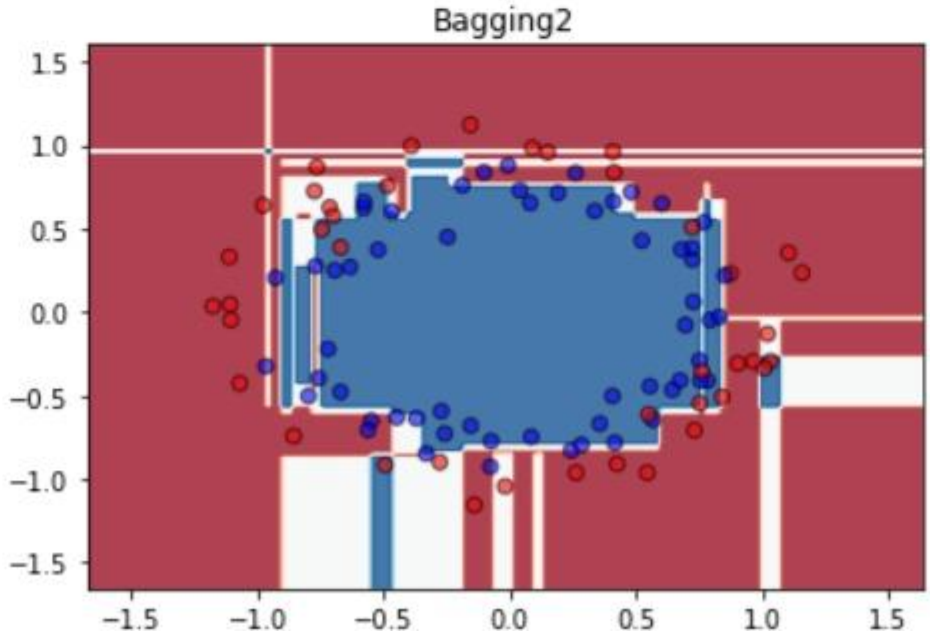
Q4.2) **Code:**

```python
num_estimator = [2,5,15,20]
# bsX = random.choices(trX, k = len(trX))
# bsY = random.choices(trY, k = len(trY))
# bsX, bsY = resample(trX, trY, random_state=rs)
# print(bsX)
h = .02
x_min, x_max = teX[:, 0].min() - .5, teX[:, 0].max() + .5
y_min, y_max = teX[:, 1].min() - .5, teX[:, 1].max() + .5
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
for i in num_estimator:
    df_prediction = pd.DataFrame()
    Z=pd.DataFrame()
    for n in range(i):
        idx = random.choices(range(len(trX)), k = len(trX))
        bsX=trX[idx]
        bsY=trY[idx]
        est =  DecisionTreeClassifier(random_state=rs)
        clf = est.fit(list(bsX), list(bsY))
        predY = clf.predict(teX)
        Z[str(n)] = clf.predict_proba(np.c_[xx.ravel(), yy.ravel()])[:, 1]
        df_prediction['M'+str(n)] = predY
    df_prediction['FinalPredict'] = df_prediction.mode(axis=1)[0]
    Z['vote']=Z.mean(axis=1)

    dtAccuracy = accuracy_score(teY, df_prediction['FinalPredict'])
    print("The Accuracy = ",dtAccuracy)
    plotEstimator(trX, trY, teX, teY, est,'Bagging'+str(i),np.array(Z['vote']))
```

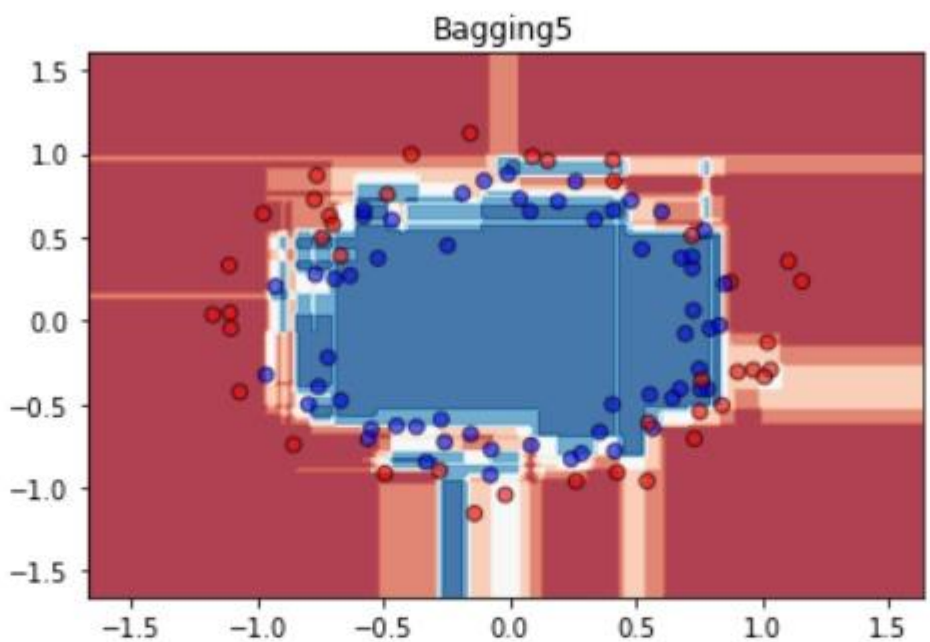
2 Bags

The Accuracy = 0.696969696969697



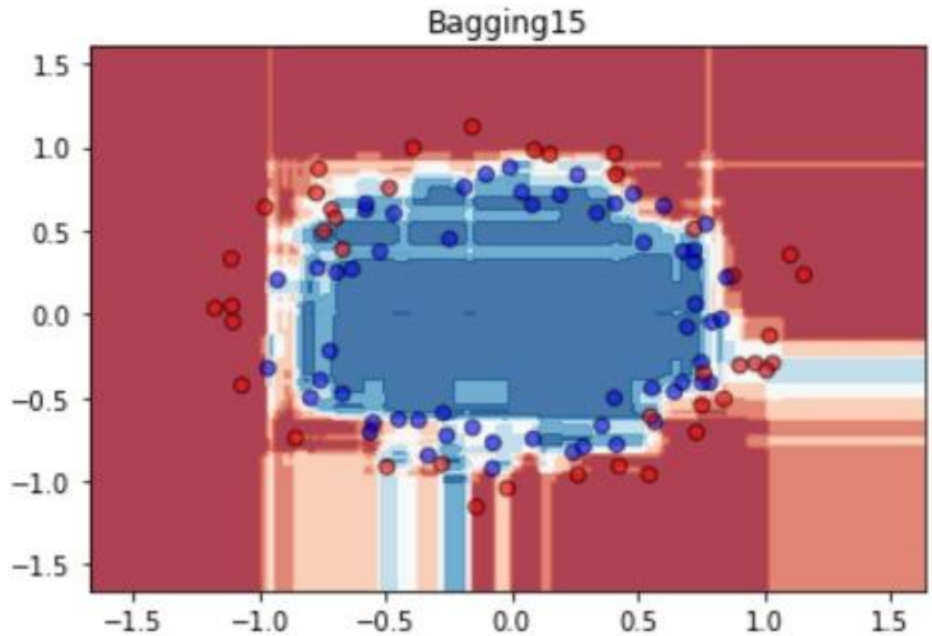
5 Bags

The Accuracy = 0.7171717171717171



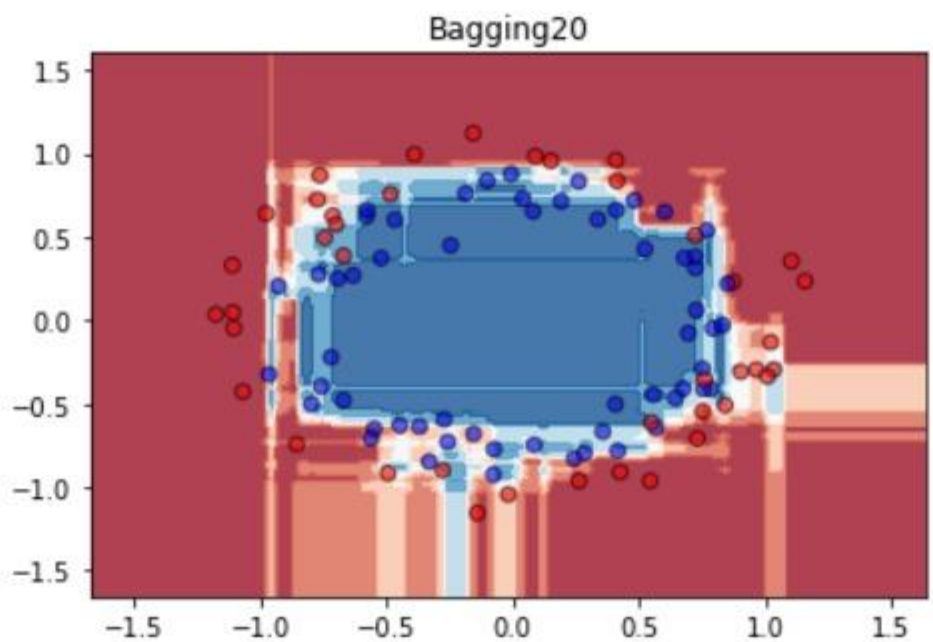
15 Bags

```
The Accuracy =  0.7070707070707071
```



20 Bags

```
The Accuracy =  0.7474747474747475
```

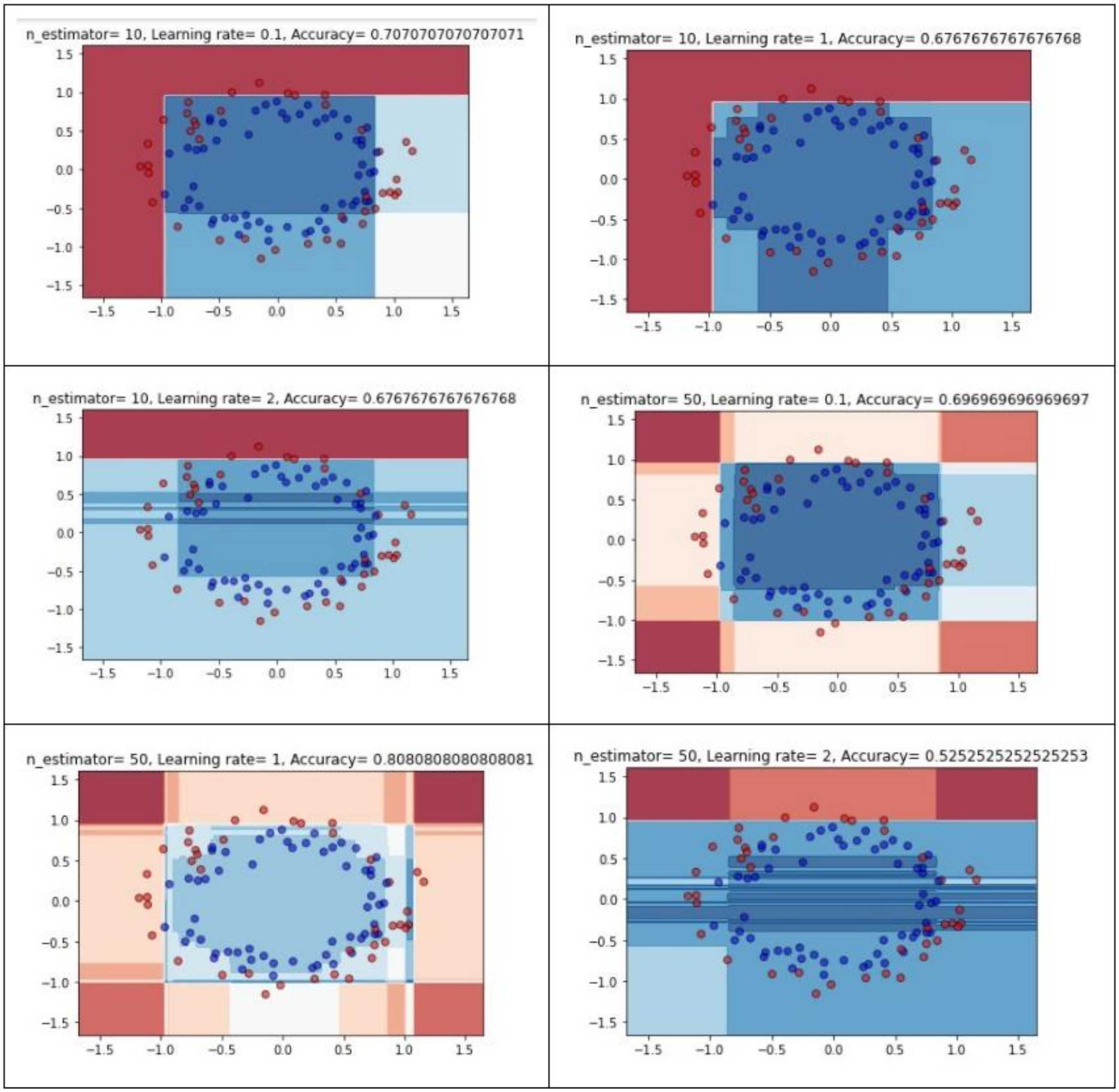


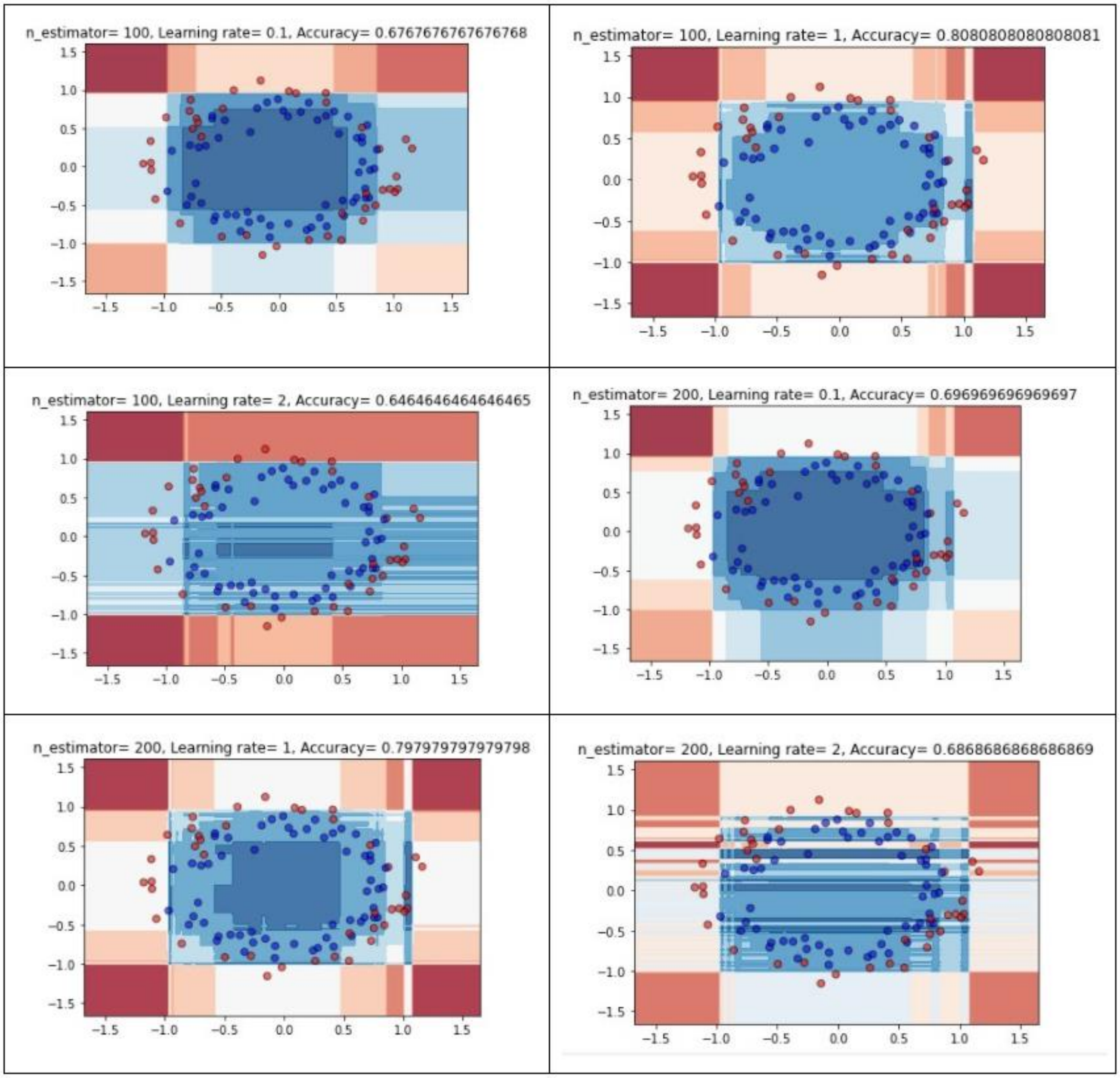
Q5)

A single tree highly overfits the data. Using **‘t’ trees** ensures that each tree uses **different sets of data and variables**, where each tree yields to low bias and high variance. Hence, taking the average over all trees (ensembling: Bagging) or taking the **most frequent prediction** among the trees, each fitted to a subset of the original data set, we arrive to one bagged predictor, where the mean/most frequent prediction will be more stable and **less overfit**. [2]
The strength of bagging lies in the fact that it ensures that all trees are **different**. Since, joining several "weak learners" to provide a "strong learning" results in a smoother, and less wild variance in the model. [3]
Thus, the variance of the Random Forest (Bagging) is smaller compared to the variance of a single Decision Tree. [4]

So, in brief bagging overcomes overfitting by creating random subsets of the features and building smaller trees using those subsets.

Q6)


n_estimator= 10, Learning rate= 0.1, Accuracy= 0.7070707070707071
n_estimator= 10, Learning rate= 1, Accuracy= 0.6767676767676768
n_estimator= 10, Learning rate= 2, Accuracy= 0.6767676767676768
n_estimator= 50, Learning rate= 0.1, Accuracy= 0.696969696969697
n_estimator= 50, Learning rate= 1, Accuracy= 0.8080808080808081
n_estimator= 50, Learning rate= 2, Accuracy= 0.5252525252525253


n_estimator= 100, Learning rate= 0.1, Accuracy= 0.6767676767676768
n_estimator= 100, Learning rate= 1, Accuracy= 0.8080808080808081
n_estimator= 100, Learning rate= 2, Accuracy= 0.6464646464646465
n_estimator= 200, Learning rate= 0.1, Accuracy= 0.696969696969697
n_estimator= 200, Learning rate= 1, Accuracy= 0.797979797979798
n_estimator= 200, Learning rate= 2, Accuracy= 0.6868686868686869

References:

[1] machine learning - When should I use Gini Impurity as opposed to Information Gain (Entropy)? - Data Science Stack Exchange
[2] (8) How does bagging avoid overfitting in Random Forest classification? - Quora
[3] Why does "bagging" in machine learning decrease variance? (techopedia.com)
[4] Understanding the Effect of Bagging on Variance and Bias visually | by Dr. Robert Kübler | Towards Data Science