uOttawa

Professional Master's in Artificial Intelligence
APPLIED MACHINE LEARNING (ELG 5255)

Subject: Assignment 4 (Decision Tree and Ensemble Learning)

By

Mohamed Sayed Abdelwahab Hussein

Abdelmageed Ahmed Abdelmageed Hassan

Under Supervision

Dr. Murat Simsek

1. build a decision tree by using Gini Index.

Page : Date : / /

1- Gini-Split for weather f1

weather → 10 Points

Sunny	Cloudy	Rainy
3 Points	3 Points	4 Points
Yes = 2	Yes = 2	Yes = 1
No = 1	No = 3	No = 3

Gini = $1 - \sum P(i)^2$

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

$= 4/9 = 0.44$ $= 0.44$ $= 0.375$

Gini-Split $= 0.44 * \frac{3}{10} + .44 * \frac{3}{10} + 0.375 * \frac{4}{10}$

$= \boxed{0.417}$

2- Gini-Split for temperature f2

Hot = 4 Points	Mild = 5 Points	Cool = 1 Point
Yes = 2	Yes = 3	Yes = 0
No = 2	No = 2	No = 1

Gini $= 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2$ $= 1 - (\frac{3}{5})^2 - (\frac{2}{5})^2$ $= 1 - 0.1 = 0$

$= 0.5$ $= 0.48$

Gini-Split $= 0.5 * \frac{4}{10} + 0.48 * \frac{5}{10} + 0$

$= \boxed{0.44}$

Page : Date : / /

3- gini-split for humidity (10 Points)

High = 7 Points
yes = 3
No = 4

$gini = 1-(\frac{3}{7})^2-(\frac{4}{7})^2$
$= 0.487$

Normal = 3 Points
yes = 2
No = 1

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

gini-split $= 0.489 \times \frac{7}{10} + 0.44 * \frac{3}{10} =$
$= \boxed{0.476}$

4- gini-split for wind

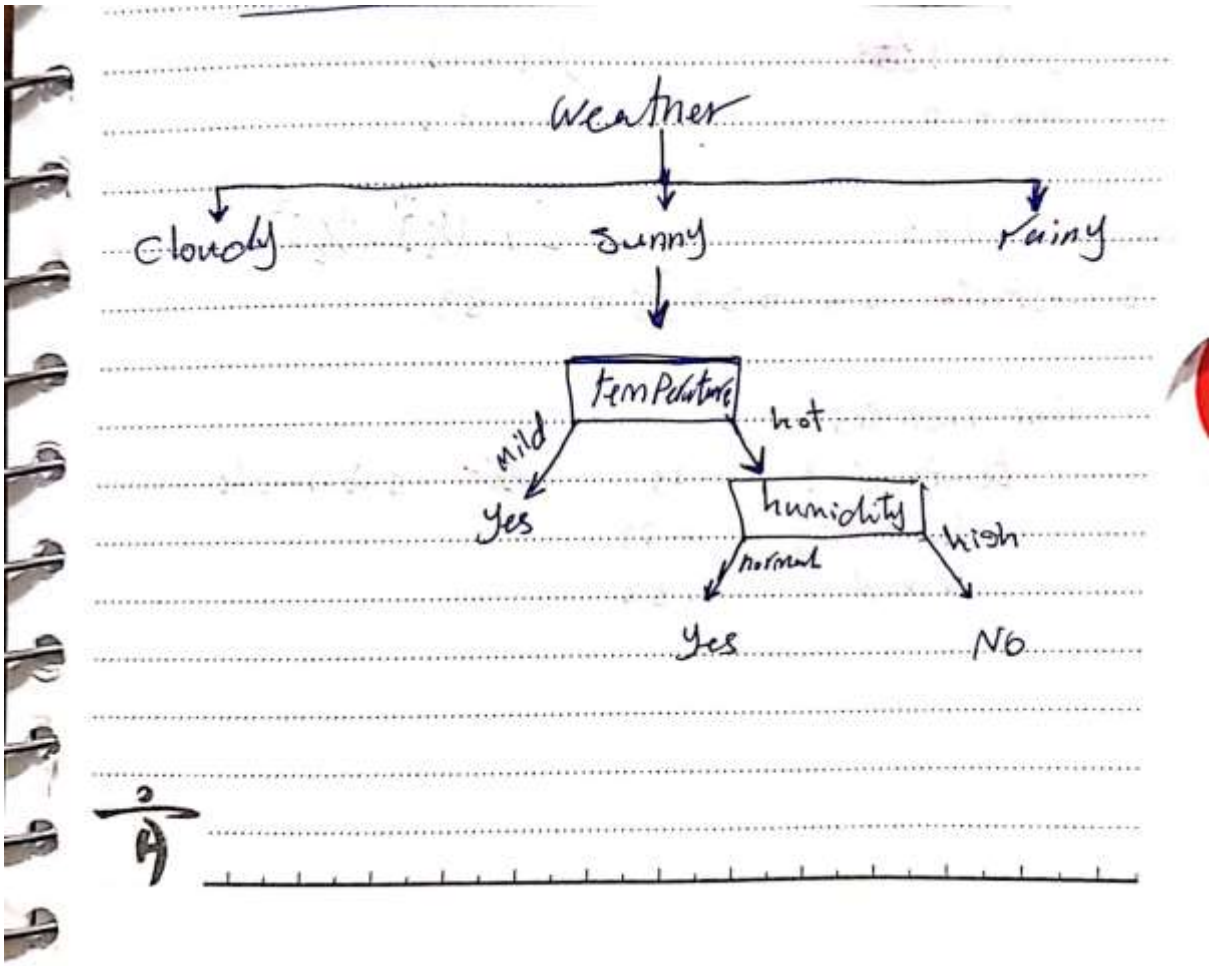
Weak = 4 Point
yes = 3
No = 1

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

strong = 6 Point
yes = 2
No = 4

$= 1-(\frac{2}{6})^2-(\frac{4}{6})^2$
$= 0.44$

gini-split $= 0.375 * \frac{4}{12} + 0.44 * \frac{6}{10} =$
$= \boxed{0.417}$


Weather
Cloudy
Sunny
Rainy
Temperature
Mild
hot
Yes
humidity
normal
High
Yes
No

Page : Date : / /

→ for Cloudy

f_1	f_2	f_3	f_4	Label
Cloudy	hot	high	weak	No
cloudy	mild	high	strong	yes
Cloudy	hot	normal	weak	yes

→ gini-split for temperature f_2 (3 point)

hot = 2 points — mild = 1 point

yes = 1 — yes = 1

No = 1 — No = 0

gini = $1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = 0.5$ — $= 1 - 0 - 1 = 0$

gini-split = $0.5 \times \frac{2}{3} + 0 = \boxed{0.33}$

→ gini-split for humidity f_3 (3 point)

high = 2 point — normal = 0

yes = 1 (No = 1) — yes = 1

No = 0

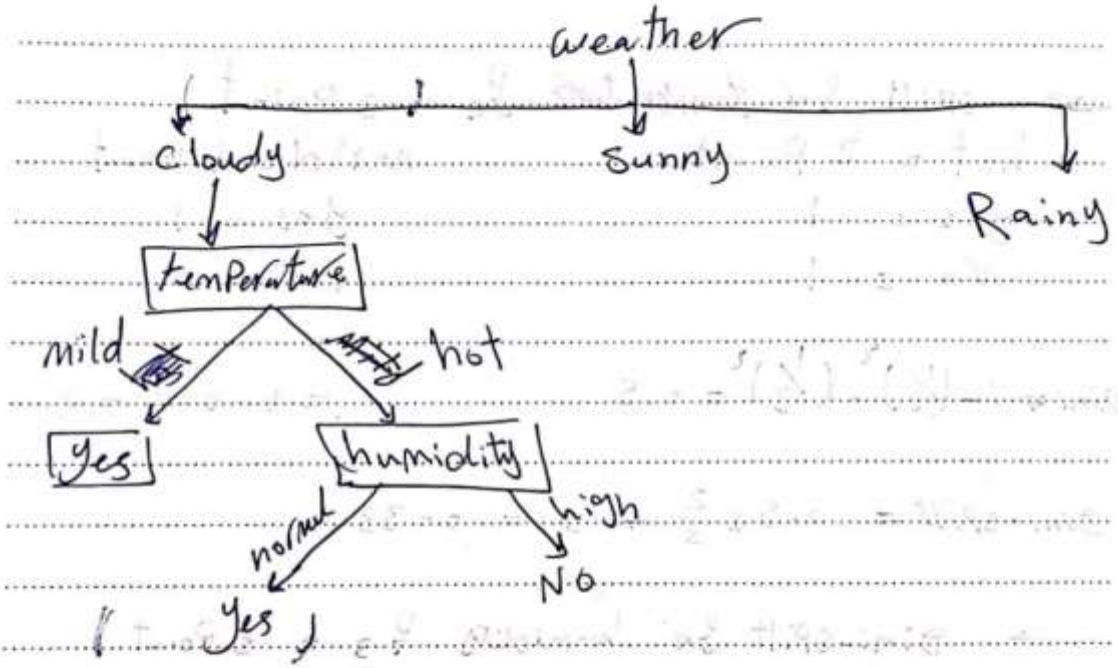
gini-split = $\boxed{0.33}$

→ gini split for wind f_4 = $\boxed{0.33}$

2

Page : Date : / /

gini-index
temperature = 0.333
humidity = 0.333
wind = 0.333



Page : Date : / /

→ for Rainy

f_1	f_2	f_3	f_4	L_i
Rainy	Mild	high	Strong	No
Rainy	Cool	normal	Strong	No
Rainy	Mild	high	Weak	yes
Rainy	Mild	high	Strong	No

→ gini split for temp f_2 (4 Points)

Mild (3 Points) Cool = 1 Point

yes = 1, No = 2 yes = 0, No = 1

gini index $= 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.44$ $= 1 - 0 - 1 = 0$

gini-split $= 0.44 * \frac{3}{4} + 0 = \boxed{0.33}$ (4/9)

→ gini-split for humidity f_3 (4 Points)

high (3) normal (1)

yes = 1, No = 2 yes = 0, No = 1

gini $= 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = \frac{4}{9}$ $= 1 - 0 - 1 = 0$

gini-split = $\boxed{0.33}$

Page : Date : / /

→ Gini-split for wind ↳

Weak (1 Point) | Strong (3 Points)

Yes = 6 / No = 1 | Yes = 0, No = 3

Gini = 1 − 0 − 1 = 0 | = $1-(\frac{0}{3})^2-(\frac{3}{3})^2 = \frac{4}{9}$

= 0

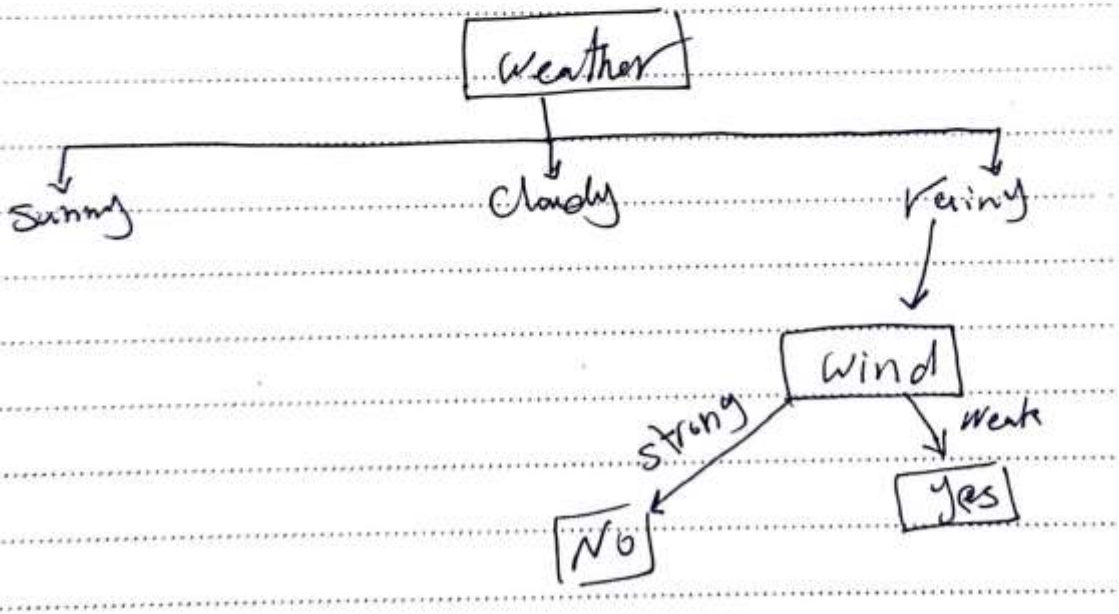
Gini-split = ~~$\frac{4}{9} \times \frac{3}{4}$~~ = 0

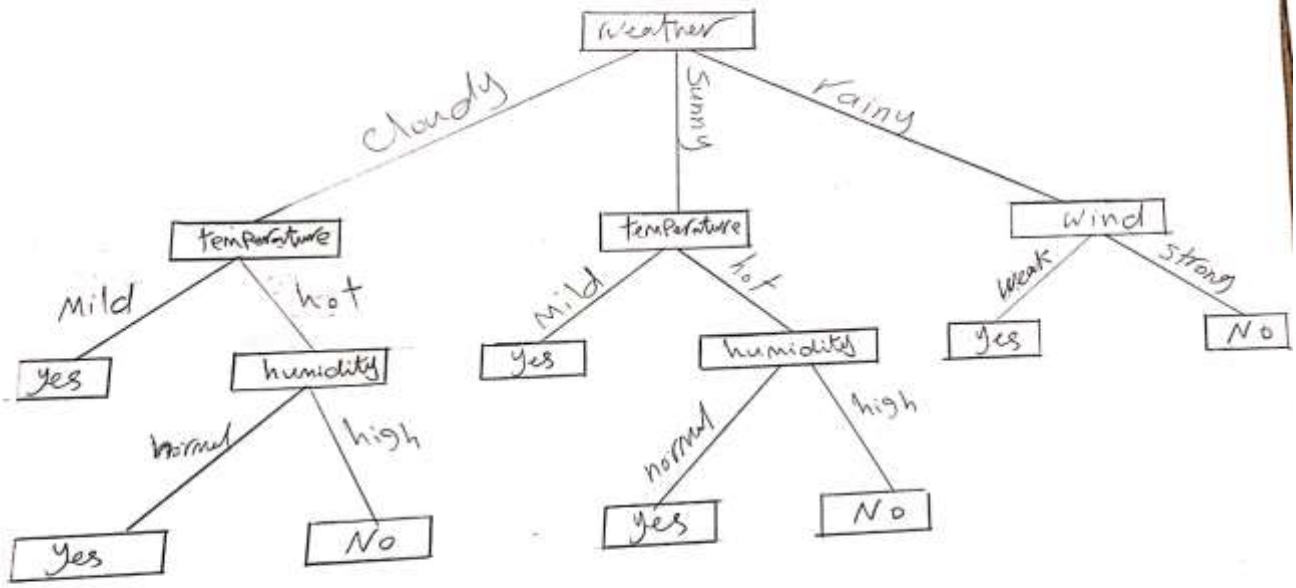
Gini-indices

temperature = 0.33

humidity = 0.33

wind = 0 ✓✓ sub-node




weather
cloudy
sunny
rainy
temperature
temperature
wind
mild
hot
mild
hot
weak
strong
yes
humidity
yes
humidity
yes
No
normal
high
normal
high
Yes
No
Yes
No

2. build a decision tree by using Information Gain.

Page : Date : / /

→ building tree with ID3 entropy Method.

entropy(Label) = $P(yes)\log(P(yes)) - P(no)\log(P(no))$

$= -0.5\log_2(0.5) - 0.5\log(0.5) = 1$

$gain(f_1) = 1 - 0.3\left(\frac{-2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}\right)$
$- 0.3\left(\frac{2}{3}\log\frac{3}{3} - \frac{1}{3}\log_2\frac{1}{3}\right)$
$= 0.124$

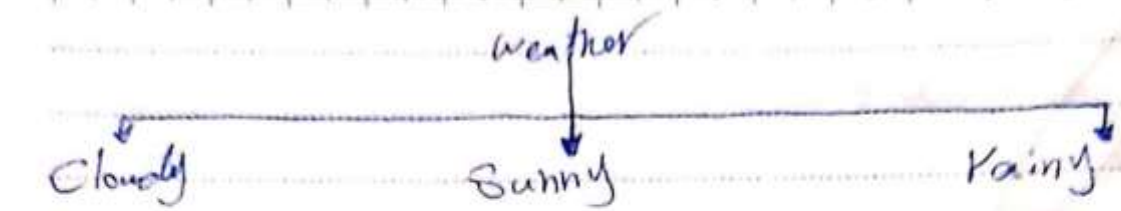
$gain(f_2) = 1 - 0.4\left(-0.5\log_2 0.5 - 0.5\log_2 0.5\right) -$
$0.5\left(\frac{-3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{3}{5}\right) - 0.1 \times 0$
$= 0.115$

$gain(f_3) = 1 - 0.7\left(\frac{-4}{7}\log\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}\right) - 0.3($
$-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) = 0.035$

$gain(f_4) = 1 - 0.6\left(\frac{-2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log\frac{4}{6}\right)$
$- 4/10\left(-3/4\log_2\frac{3}{4} - 1/4\log_2\frac{1}{4}\right)$
$= 0.124$

so f_1 (~~weather~~ temperature) is will be first split as it has maximum split

Page : Date : / /



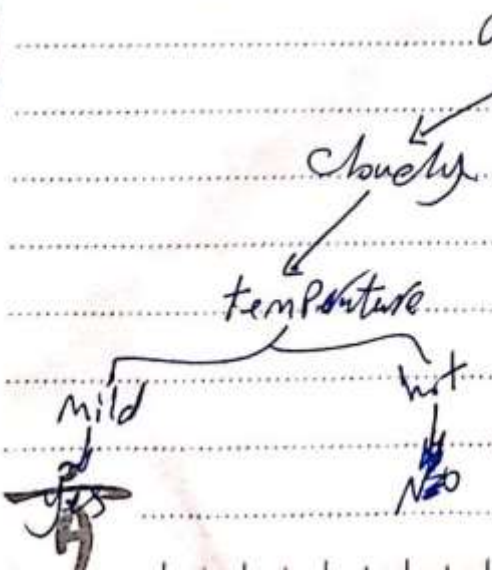
entropy (Cloudy) $= -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$

$IG(f_2) = 0.918 - \frac{3}{5}(0.5\log_2 0.5 - 0.5\log_2 0.5)$
$- \frac{1}{5}(-1*\log_2(1) = 0.251$

$IG(f_3) = 0.918 - \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2})$
$-\frac{1}{3}(-1*\log_2 1) = 0.251$

$IG(f_4) = 0.918 - \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) - 0$
$= 0.251$

So we will split cloud with f_2 (temperature)



Page : Date : / /

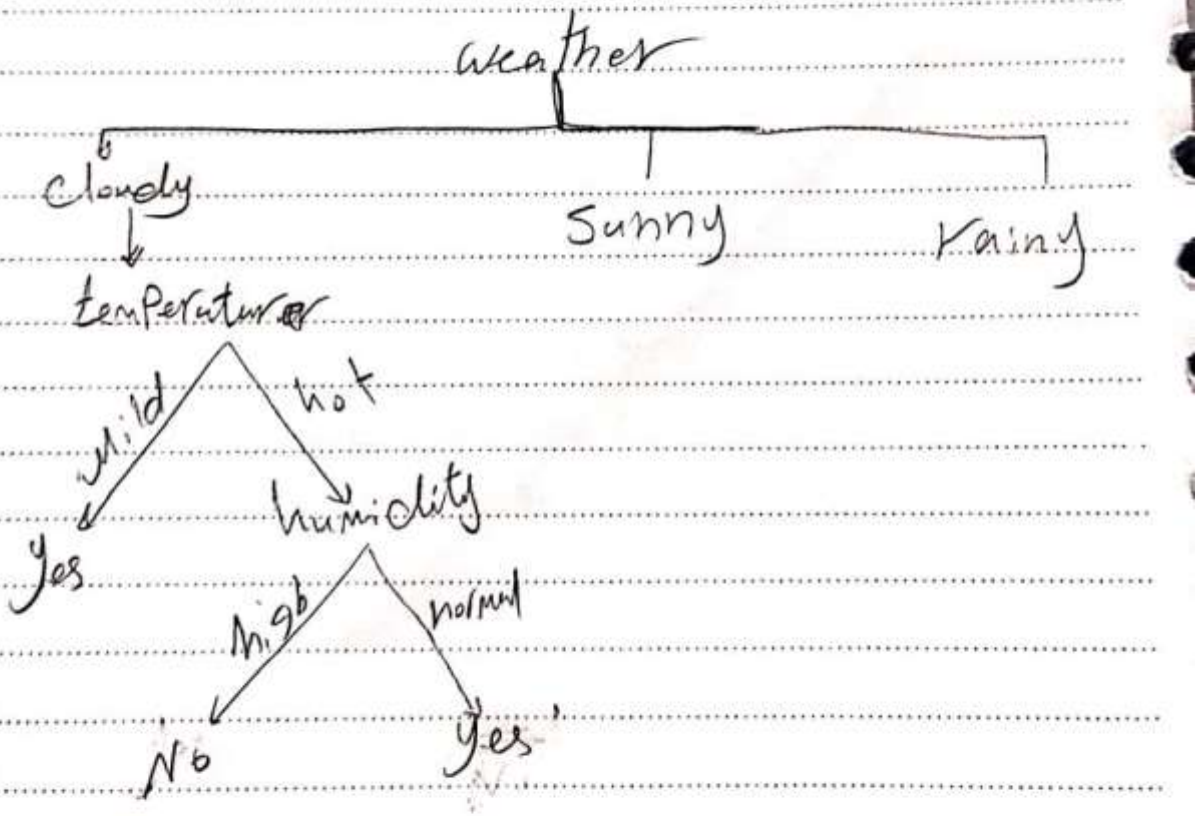
entropy for temperature (hot) of 2 points

$= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

$IG(\text{humidity}) = 1 - \frac{1}{2}(-1\log_2 1) = 1$

$IG(\text{wind}) = 1 - 1(-\log 1) = 1 - 0 = 1$

humidity is the closest to the left



Page : Date : / /

entropy for Sunny

$= -2/3 \log 2/3 - 1/3 \log_2 1/3 = 0.918$

$IG(f_2) = 0.918 - 2/3(-1/2 \log_2 1/2 - 1/2 \log_2 1/2)$
$- 1/3(-\log 1) = 0.25$

$IG(f_3) = 0.918 - 2/3(-0.5 \log 0.5 - 0.5 \log 0.5)$
$= 0$
$= 0.25$

$IG(f_4) = 0.918 - 2/3(-1/2 \log 1/2 - 1/2 \log 1/2) - 0$
$= 0.25$

$IG(f_2) = IG(f_3) = IG(f_4) = 0.25$

f_2 is the closest to the left

if f_2 = mild & label = yes

if "hot"

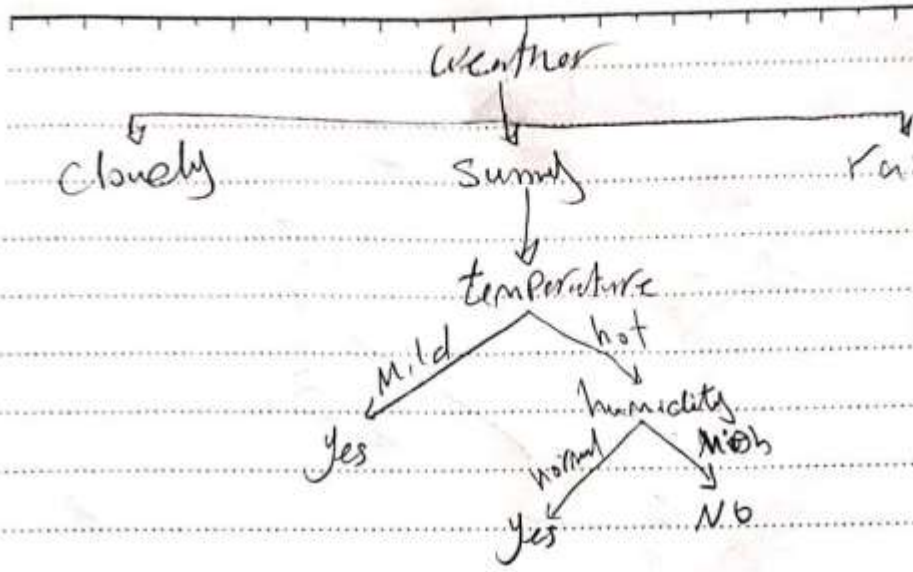
entropy (temperature = hot) $= -1/2 \log_2 1/2 - 1/2 \log_2 1/2$
$= 1$

$IG(\text{humidity}) = 1 - 1(-1/2 \log_2 1/2 - 1/2 \log 1/2) = 0$

$IG(\text{wind}) = 1 - 0.5(-\log 1) - 1/2(-1 \log_2 1) = 1$

2/4

Page : Date : / /



for rainy

entropy $= \frac{-3}{4} \log 3/4 - 1/4 \log_2 1/4 = 0.811$

$GI(f_2) = 0.811 - \frac{3}{4}\left(\frac{-2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right)$
$= -\frac{1}{4}(-1\log_2 1) = 0.122$

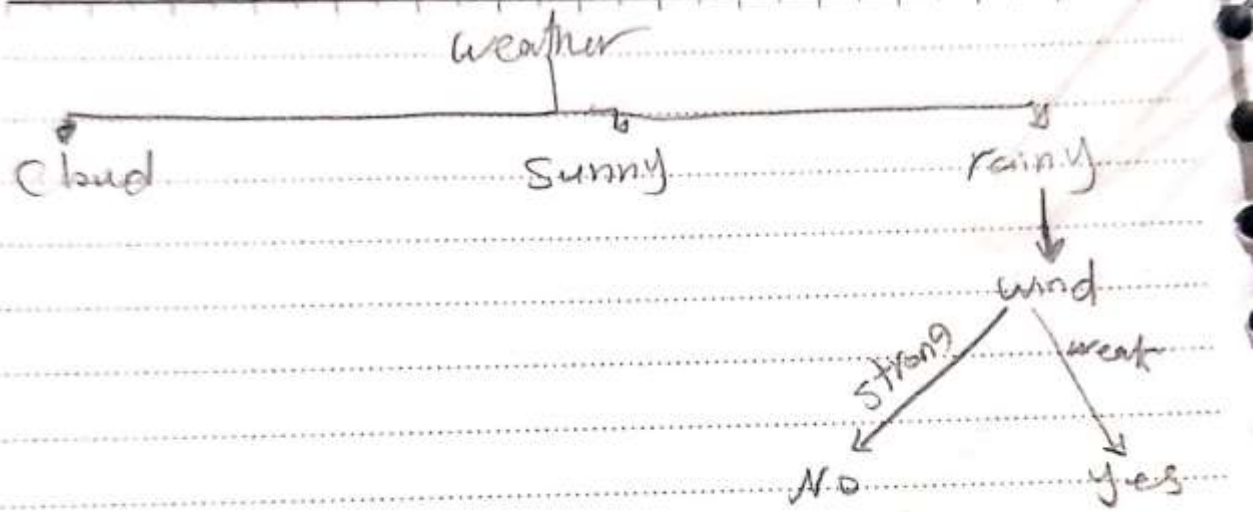
$GI(f_3) = 0.811 - \frac{3}{4}\left(\frac{-2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right)$
$= -\frac{1}{4}(-1\log_2 1) = 0.122$

$GI(f_4) = 0.811 - \frac{3}{4}(-1\log 1) = \frac{1}{4}(-1\log 1)$
$\rightarrow 0.811 - 0 - 0 = 0.811$

$\rightarrow$ So wind has highest IG

$\frac{2}{4}$

Page : Date : / /



ge : ..... Date : ... / ... / ...

Yes
wild
Temperature
Yes
normal
humidity
hot
Cloudy
No
high
Yes
wild
Temperature
Sunny
No
Strong
wind
hot
Yes
weak
Rainy
No
Strong
wind
Yes
weak

3. Please compare the advantages and disadvantages between Gini Index and Information Gain.

- Information Gain Advantages: -
  - Information gain ratio biases the decision tree against considering attributes with a large number of distinct values. So, it solves the drawback of information gain namely, information gain applied to attributes that can take on a large number of distinct values might learn the training set too well.
  - It creates a comprehensive analysis of consequences along each branch and identifies decision nodes that need further analysis.
- Information Gain Disadvantages: -
  - A notable problem occurs when information gain is applied to attributes that can take on a large number of distinct values(biased).
  - Subsets are more likely to be pure if there is a large number of values (overfitting).

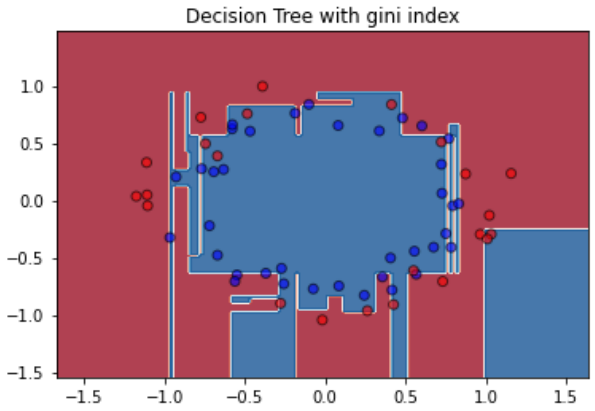
- Gini index Advantages: -
  - favours larger partitions (distributions) and is very easy to implement and interpret.
  - Modification of the information gain that reduces its bias.
  - it deals with inequality, so it can judge the distribution pattern better.
- Gini index disadvantages: -
  - Sample Bias, the validity of the Gini index can be dependent on sample size.
  - Data Inaccuracy, the Gini index is sometimes prone to random and systematic data errors, it can create problems with the index value.
  - Degeneracy, in some exceptional cases, the Gini index value can be the same for different distributions.

4. Use Circle Dataset. Apply decision tree on the Circle Dataset, set criterion as Gini and entropy, get the accuracy of the testing results, plot the decision boundaries and explain the difference between these criterions.

▾ (4.1) DT with gini Index

```
dtEstimator_gini = DecisionTreeClassifier(criterion="gini")
dtEstimator_gini.fit(X_train, y_train)
predY = dtEstimator_gini.predict(X_test)
dtAccuracy = accuracy_score(y_test, predY)
print("test accuracy is: ",round(dtAccuracy,3))
plotEstimator(X_train, y_train, X_test, y_test, dtEstimator_gini, 'Decision Tree with gini index')
```

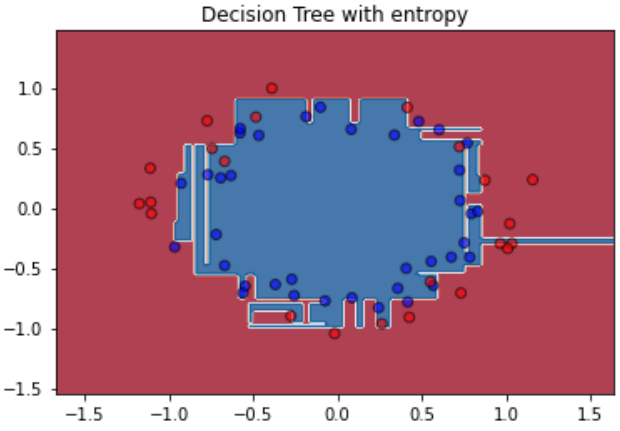
```
test accuracy is:  0.6
```



▾ (4.2) DT with entropy

```
dtEstimator_entropy = DecisionTreeClassifier(criterion="entropy")
dtEstimator_entropy.fit(X_train, y_train)
predY = dtEstimator_entropy.predict(X_test)
dtAccuracy = accuracy_score(y_test, predY)
print("test accuracy is: ",round(dtAccuracy,3))
plotEstimator(X_train, y_train, X_test, y_test, dtEstimator_entropy, 'Decision Tree with entropy')
```

```
test accuracy is:  0.717
```



Gini measurement is the probability of a random sample being classified incorrectly if we randomly pick a label according to the distribution in a branch. Entropy is a measurement of information (or rather lack thereof). You calculate the information gain by making a split. Which is the difference in entropies. This measures how you reduce the uncertainty about the label. Based on circle dataset the **entropy criterion is better than Gini index criterion**.

5. Use Classification Dataset. Use training set to obtain the importance of features. Plot Validation Accuracy (y-axis) vs Top K Important Feature (x-axis) curve; where 4-fold cross validation should be used, and also plot Test Accuracy vs Top K Important Feature curve.

   We will fit the DT on the classification dataset and get the indexes of top features

   ▾ **(5.1)** get top K important features

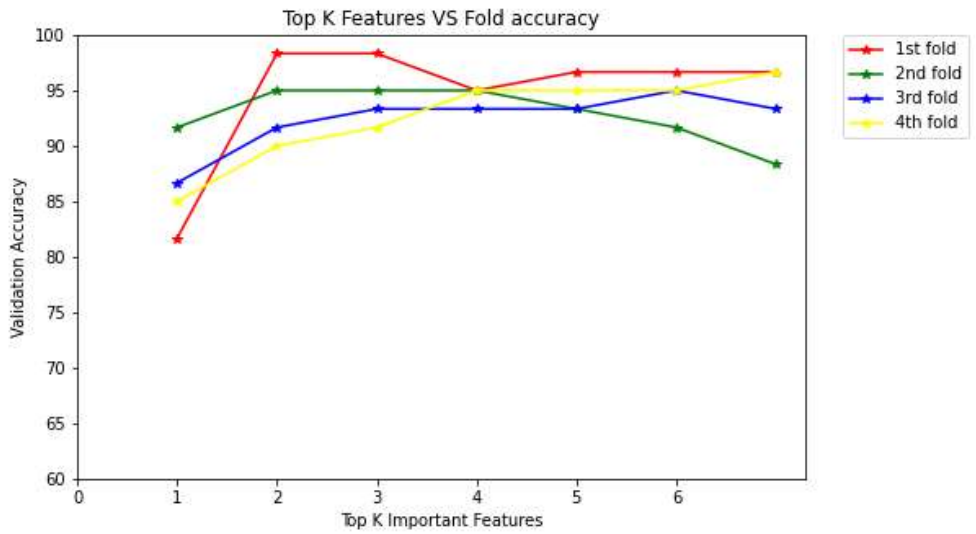
```
[ ]  tree_model = DecisionTreeClassifier(random_state=0)
     tree_model.fit(X_train2, y_train2)
     features_import = tree_model.feature_importances_
     idx_sorted = np.argsort(-features_import)[0:7]
     idx_sorted

     array([ 8,  0, 18, 10, 17, 14, 11])
```

Then we will fit the model with important features incrementally using 4-fold cross validation; first iteration we will fit with most important one, second iteration with two most important features and so on.

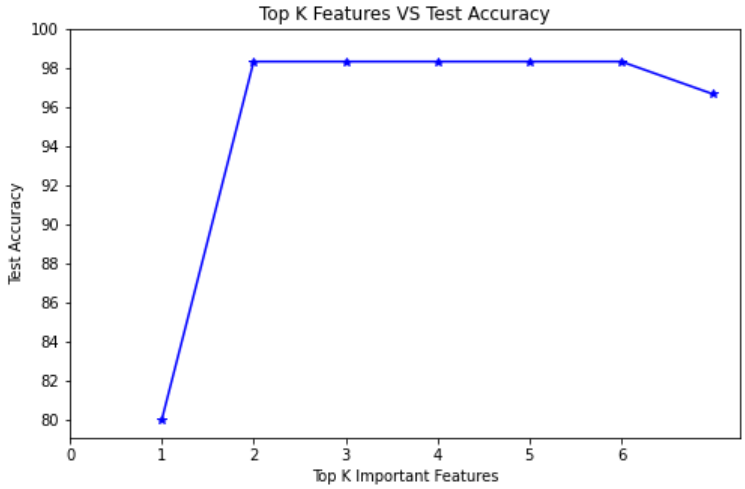
▾ **(5.3)** Plot Validation Accuracy (y-axis) vs Top K Important Feature (x-axis) curve with 4-folds

```
[136] values = [x for x in range(1,8)]
      plot_importance_vs_accuracys(values[:8], valid_acc, "Top K Features VS Fold accuracy")
```

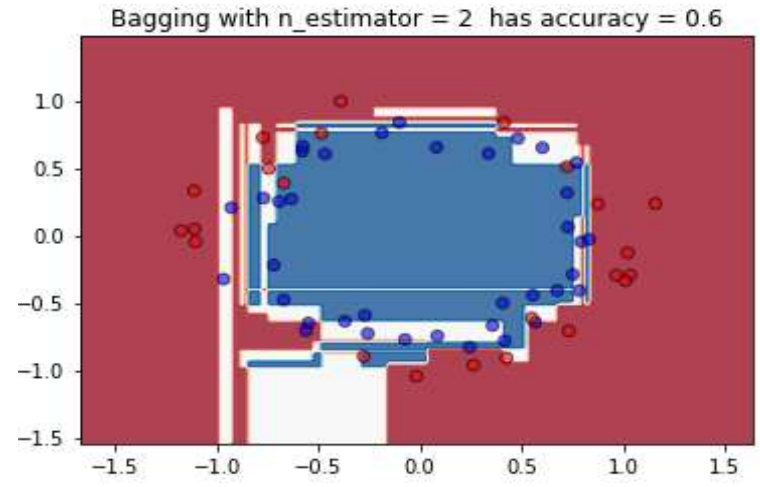


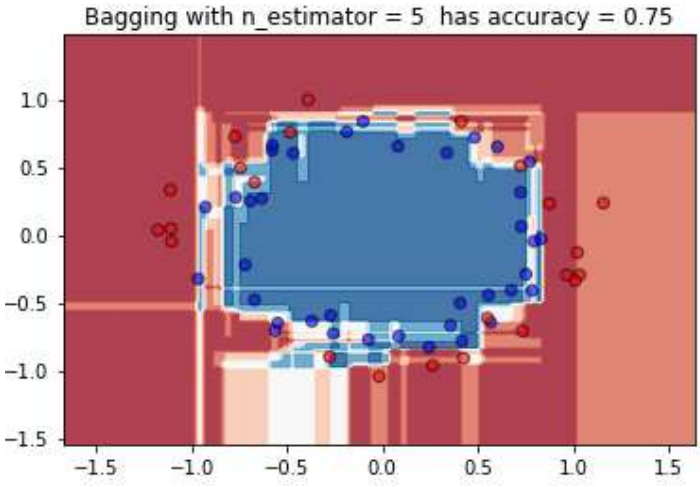
(5.4) plot Test Accuracy vs Top K Important Feature curve

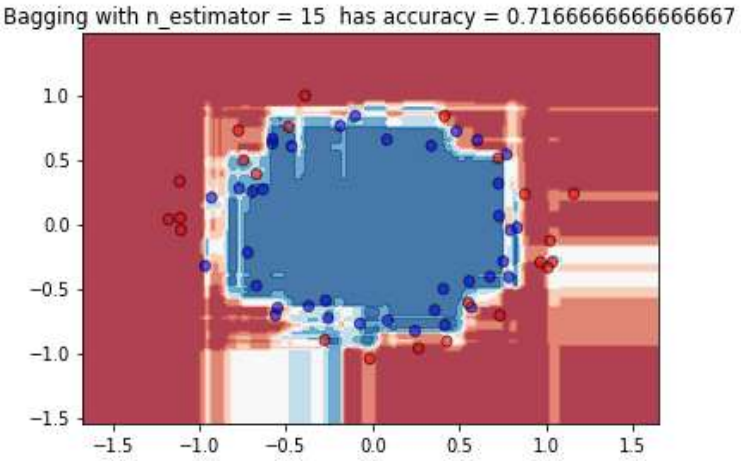
```
[138] values = [x for x in range(1,8)]
      plot_importance_vs_accuracys(values[:8], test_accuracy, "Top K Features VS Test Accuracy");
```

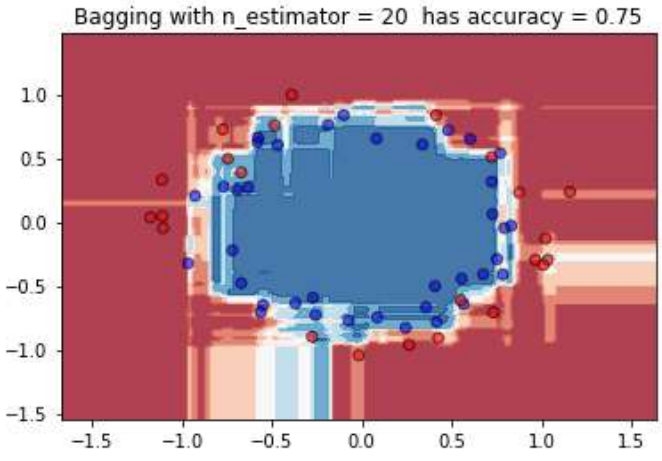


6. Use Circle Dataset. Set the number of estimators as 2, 5, 15, 20 respectively, and generate the results accordingly (i.e., accuracy and decision boundary).





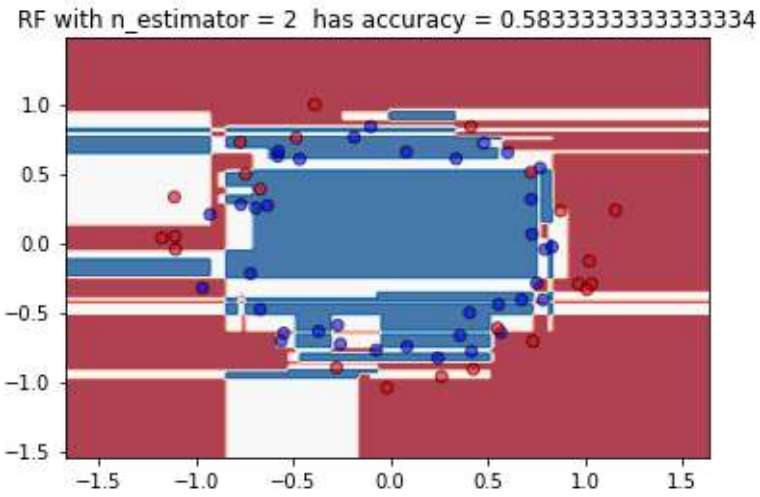


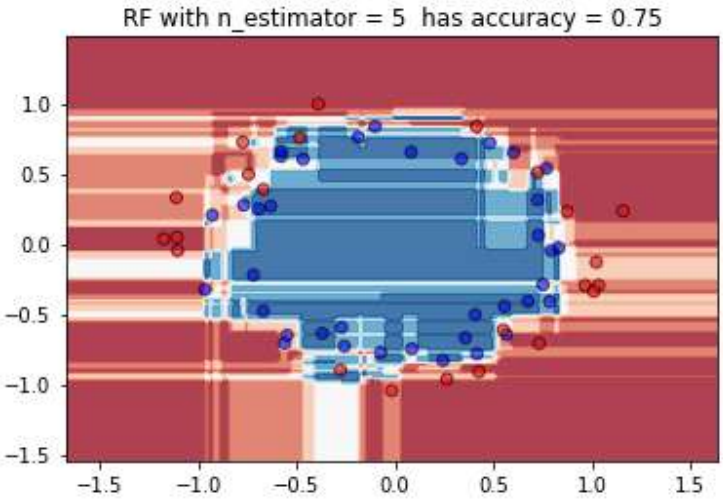


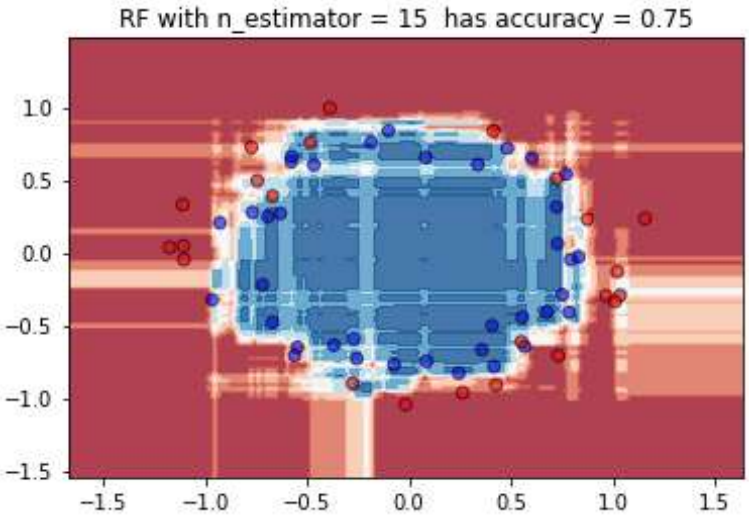
7. Explain why bagging can reduce the variance and mitigate the overfitting problem.

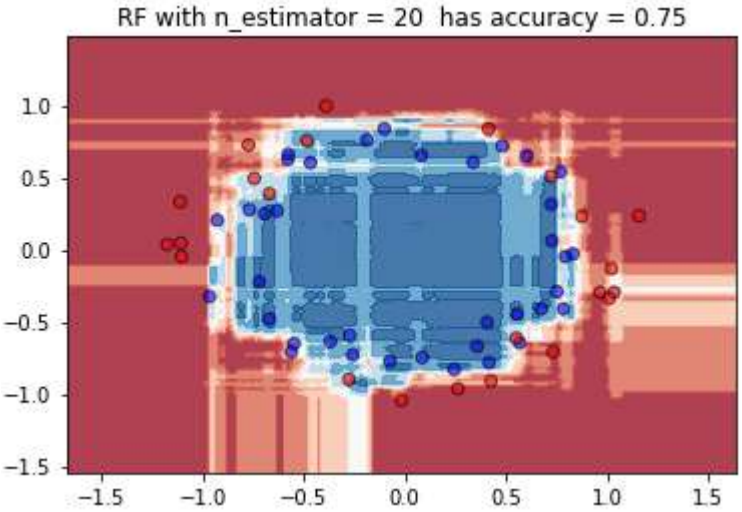
   Bagging can create many predictors by bootstrapping the data randomly subsample the dataset many times, and train a model using each subsample. We can then aggregate our models like averaging out the predictions of each model. and this can reduce variance and overfitting.

8. Use Circle Dataset. Set the number of estimators as 2, 5, 15, 20 respectively, and generate the results accordingly (i.e., accuracy and decision boundary).





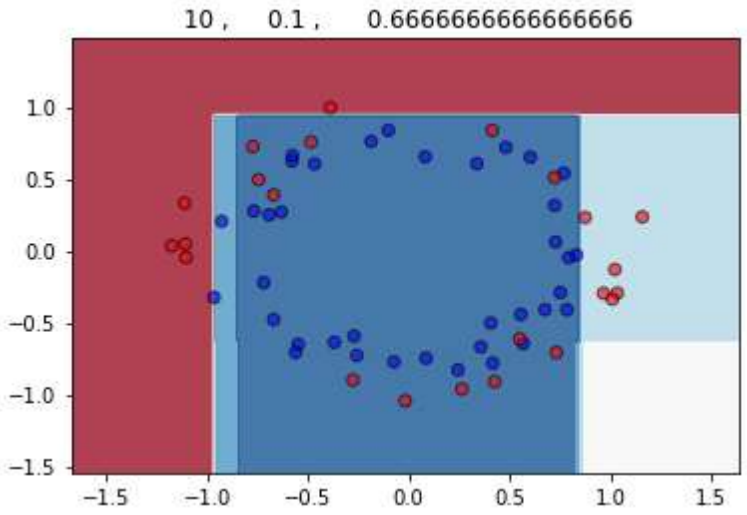


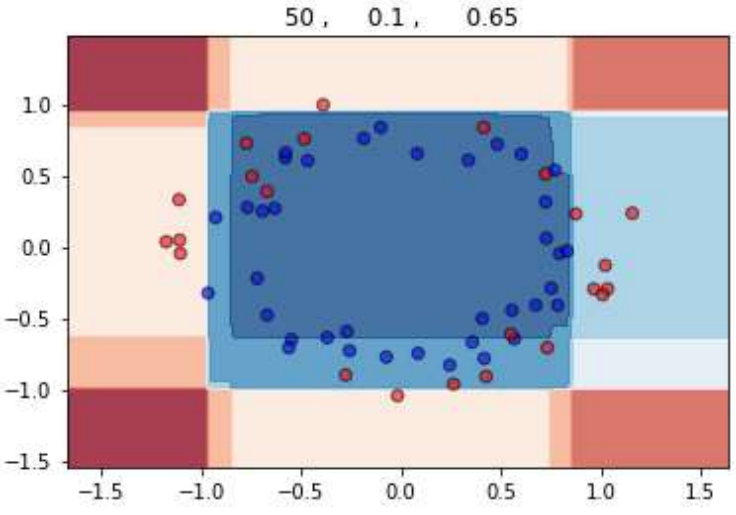


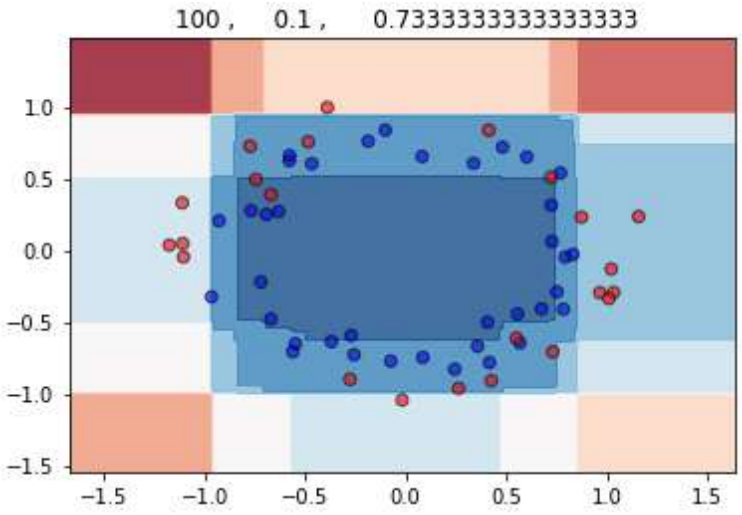
9. Compare with bagging results and explain the difference between Bagging and Random Forest.

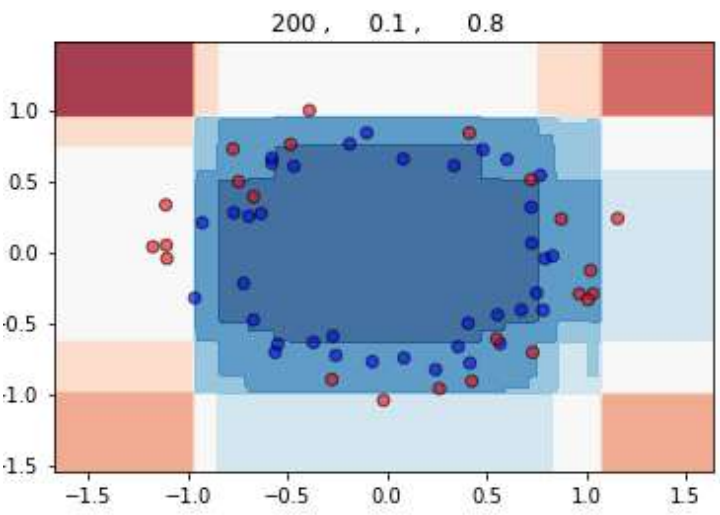
   The fundamental difference is that in Random forests, only a subset of features is selected at random out of the total and the best split feature from the subset is used to split each node in a tree, unlike in bagging where all features are considered for splitting a node. As we can see from the above results there are **slightly difference between bagging accuracy and random forest accuracy**.

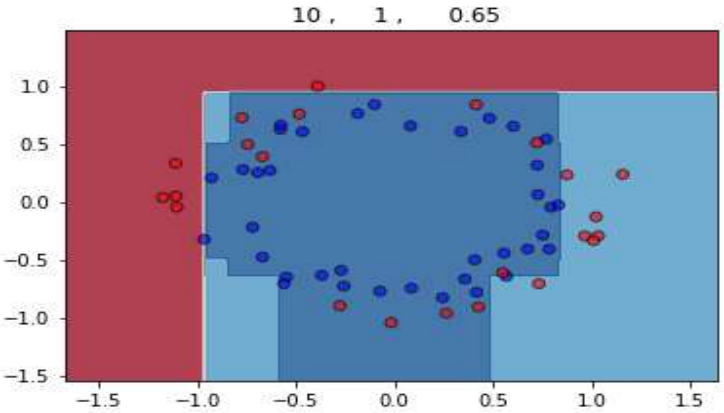
10. Use Circle Dataset. There are 2 important hyperparameters in AdaBoost, i.e., the number of estimators (ne), and learning rate (lr). Please plot 12 subfigures as the following table's setup. Each figure should plot the decision boundary and each of their title should be the same format as {n_estimaotrs}, {learning_rate}, {accuracy}.

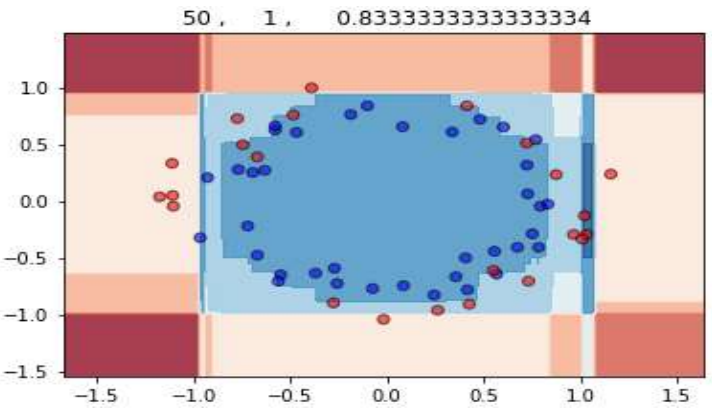


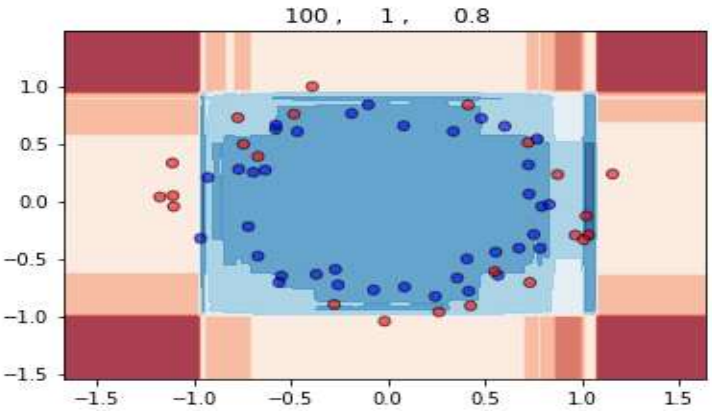


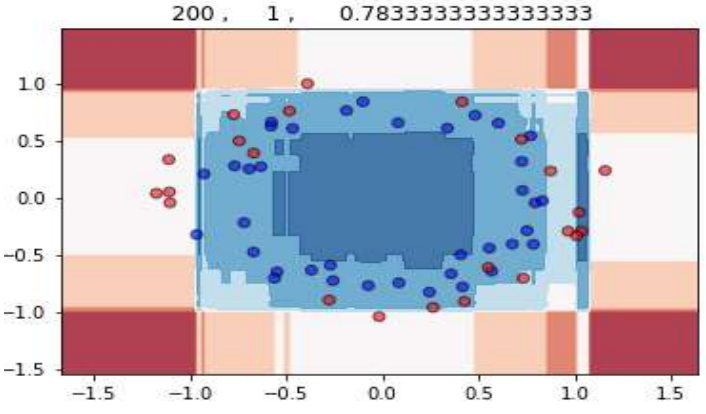


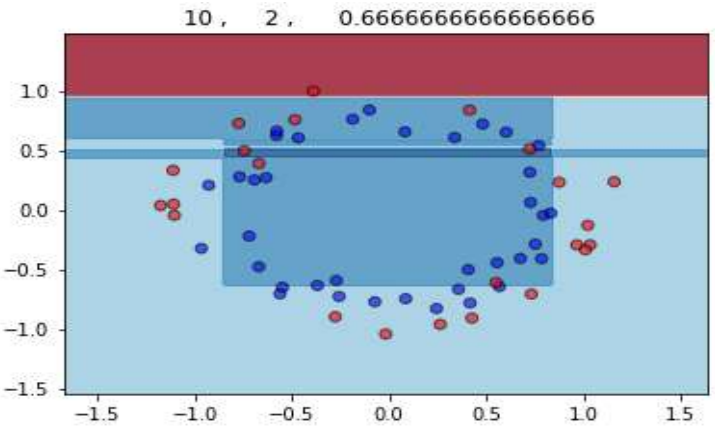


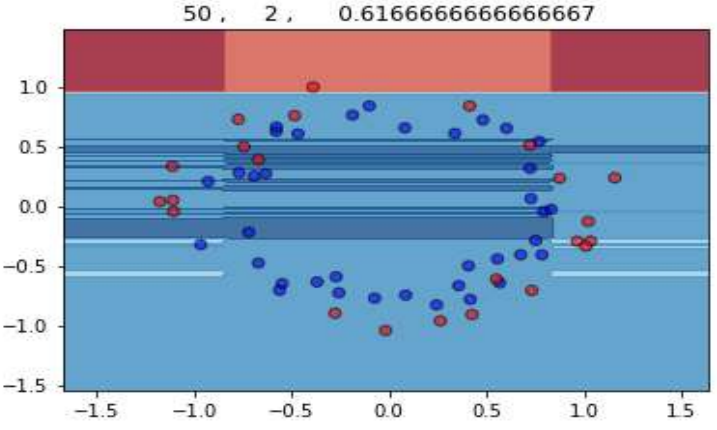

10 , 1 , 0.65

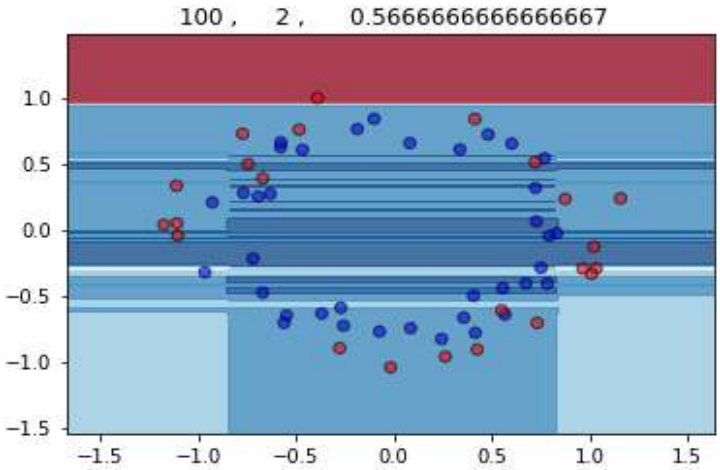

50 , 1 , 0.8333333333333334

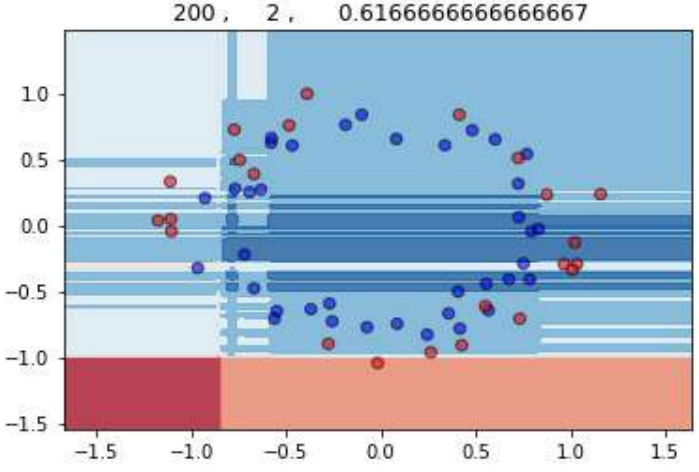

100 , 1 , 0.8


200 , 1 , 0.7833333333333333


10 , 2 , 0.6666666666666666


50 , 2 , 0.6166666666666667


100 , 2 , 0.5666666666666667


200 , 2 , 0.6166666666666667

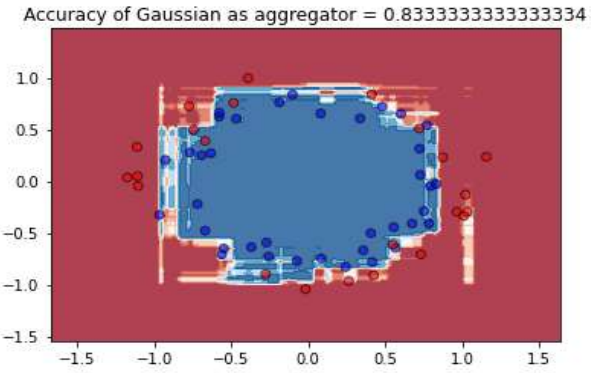
11. We have tuned the Decision Tree, Bagging, Random Forest, and AdaBoost in the previous section. Use these fine-tuned model as base estimators and use Naive Bayes, Logistic Regression, and Decision Tree as aggregators to generate the results accordingly (i.e., accuracy and decision boundary)

▾ Base Estimaters

```
[ ]  base_estimaters = list()
     base_estimaters.append(('DT',DecisionTreeClassifier(criterion="entropy", random_state=0)))
     base_estimaters.append(('Bagging' ,BaggingClassifier(n_estimators=5, random_state=0)))
     base_estimaters.append(('RF', RandomForestClassifier(n_estimators=5, random_state=0)))
     base_estimaters.append(('Adaboost', AdaBoostClassifier(n_estimators=50, learning_rate= 1, random_state=0)))
```

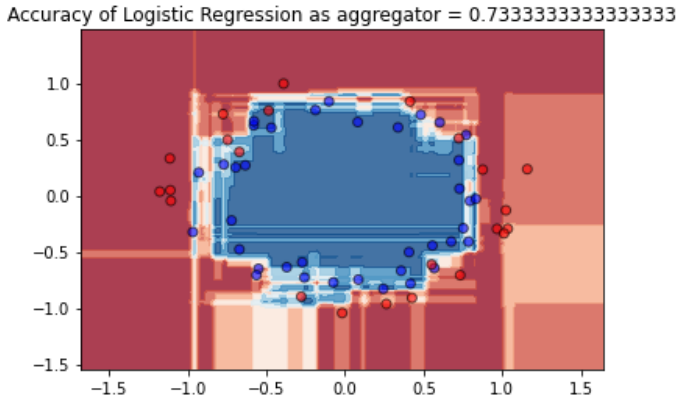
▾ (11.1) Naive Bayes as Aggregator

```
[ ]  aggregator1 =GaussianNB()
     model1 = StackingClassifier(estimators=base_estimaters, final_estimator=aggregator1, cv=5)
     score = model1.fit(X_train, y_train).score(X_test, y_test)
     plotEstimator(X_train, y_train, X_test, y_test, model1, f'Accuracy of Gaussian as aggregator = {score}')
```



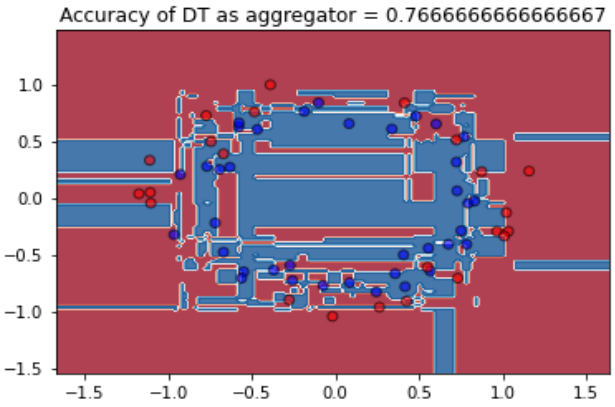
▾ (11.2) Logistic Regression as Aggregator

```
[ ]  aggregator2 =LogisticRegression()
     model2 = StackingClassifier(estimators=base_estimaters, final_estimator=aggregator2, cv=5)
     score = model2.fit(X_train, y_train).score(X_test, y_test)
     plotEstimator(X_train, y_train, X_test, y_test, model2, f'Accuracy of Logistic Regression as aggregator = {score}')
```



▾ (11.3) Decision Tree as Aggregator

```
[ ]  aggregator3 =DecisionTreeClassifier()
     model3 = StackingClassifier(estimators=base_estimaters, final_estimator=aggregator3, cv=5)
     score = model3.fit(X_train, y_train).score(X_test, y_test)
     plotEstimator(X_train, y_train, X_test, y_test, model3, f'Accuracy of DT as aggregator = {score}')
```



12. Provide a conclusion section on your report. Include overview of what you have done and learnt during the assignment.

- Learned the difference between Gini index, Information gain and the math behind them.
- Implemented DT using Gini index and information gain, showing the difference in accuracy between each one.
- Got the top important features and fit DT model on them incrementally using cross validation (4-folds), then plotted curve for top features vs validation accuracy and top features vs test accuracy.
- Implemented Bagging and Random Forest with different number of estimators (2, ,5, 15, 20) and we noticed that there is a slight difference between bagging and random forest in accuracy because they almost use the same technique, and they used for reducing the overfitting.
- Implemented AdaBoosting and tried different values for number of estimators and learning rate. The best accuracy achieved using (n_estimators = 50, learning_rate = 1).
- Used stacking technique to achieve better accuracy, we used the tuned models from DT, RF and Adaboost as a base estimator. And used Naïve Bayes, Logistic Regression and Decision Tree as Aggregators. The best accuracy achieved with Naïve Bayes as Aggregator.