

In [1]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [2]:

```
from sklearn.datasets import load_digits
```

In [3]:

```
digits=load_digits()
```

In [4]:

```
from sklearn.model_selection import train_test_split
```

In [5]:

```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.2)
```

In [6]:

```
dtmodel1=DecisionTreeClassifier()
```

In [7]:

```
dtmodel1.fit(x_train,y_train)
```

Out[7]:

```
DecisionTreeClassifier()
```

In [8]:

```
dtmodel1.score(x_test,y_test)
```

Out[8]:

```
0.8527777777777777
```

In [9]:

```
dtmodel2=DecisionTreeClassifier(max_depth=5)
```

In [10]:

```
dtmodel2.fit(x_train,y_train)
```

Out[10]:

```
DecisionTreeClassifier(max_depth=5)
```

In [11]:

```
dtmodel2.score(x_test,y_test)
```

Out[11]:

```
0.6694444444444444
```

In [12]:

```
dtmodel3=DecisionTreeClassifier(max_depth=20)
```

In [13]:

```
dtmodel3.fit(x_train,y_train)
```

Out[13]:

```
DecisionTreeClassifier(max_depth=20)
```

In [14]:

```
dtmodel3.score(x_test,y_test)
```

Out[14]:

```
0.8388888888888889
```

In [15]:

```
dtmodel4=DecisionTreeClassifier(max_depth=50)
```

In [16]:

```
dtmodel4.fit(x_train,y_train)
```

Out[16]:

```
DecisionTreeClassifier(max_depth=50)
```

In [17]:

```
dtmodel4.score(x_test,y_test)
```

Out[17]:

```
0.8416666666666667
```

In [18]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [19]:

```
rfmodel1=RandomForestClassifier()
```

In [20]:

```
rfmodel1.fit(x_train,y_train)
```

Out[20]:

```
RandomForestClassifier()
```

In [21]:

```
rfmodel1.score(x_test,y_test)
```

Out[21]:

```
0.9861111111111112
```

In [22]:

```
from sklearn.tree import DecisionTreeRegressor
```

In [23]:

```
from sklearn.datasets import load_boston
```

In [24]:

```
boston=load_boston()
```

In [25]:

```
import pandas as pd
```

In [26]:

```
data=pd.DataFrame(boston.data,columns=boston.feature_names)
```

In [27]:

```
data['MEDV']=pd.DataFrame(boston.target)
```

In [28]:

```
x=data[['RM','ZN','LSTAT']]
```

In [29]:

```
y=data[['MEDV']]
```

In [30]:

```
x_train_1,x_test_1,y_train_1,y_test_1=train_test_split(x,y,test_size=0.2)
```

In [31]:

```
dtmodel1=DecisionTreeRegressor()
```

In [32]:

```
dtmodel1.fit(x_train_1,y_train_1)
```

Out[32]:

```
DecisionTreeRegressor()
```

In [33]:

```
yTestPredict=dtmodel1.predict(x_test_1)
```

In [34]:

```
import numpy as np
```

In [35]:

```
from sklearn.metrics import mean_squared_error
```

In [36]:

```
np.sqrt(mean_squared_error(y_test_1,yTestPredict))
```

Out[36]:

6.314634418803456

In [37]:

```
from sklearn.ensemble import RandomForestRegressor
```

In [38]:

```
rfmodel1=RandomForestRegressor()
```

In [39]:

```
rfmodel1.fit(x_train_1,y_train_1.values.ravel())
```

Out[39]:

RandomForestRegressor()

In [40]:

```
yTestPredict=rfmodel1.predict(x_test_1)
```

In [41]:

```
np.sqrt(mean_squared_error(y_test_1,yTestPredict))
```

Out[41]:

5.001909118859846

In [42]:

```
from sklearn.ensemble import BaggingClassifier
```

In [43]:

```
bagmodel1=BaggingClassifier(DecisionTreeClassifier())
```

In [44]:

```
bagmodel1.fit(x_train,y_train)
```

Out[44]:

BaggingClassifier(base_estimator=DecisionTreeClassifier())

In [45]:

```
bagmodel1.score(x_test,y_test)
```

Out[45]:

0.9305555555555556

In [46]:

```
from sklearn.ensemble import BaggingRegressor
```

In [47]:

```
bagrmodel1=BaggingRegressor(DecisionTreeRegressor())
```

In [48]:

```
bagrmodel1.fit(x_train_1,y_train_1.values.ravel())
```

Out[48]:

```
BaggingRegressor(base_estimator=DecisionTreeRegressor())
```

In [49]:

```
yTestPredict=bagrmodel1.predict(x_test_1)
```

In [50]:

```
np.sqrt(mean_squared_error(y_test_1,yTestPredict))
```

Out[50]:

```
5.04968199348178
```

In [51]:

```
from sklearn.ensemble import AdaBoostClassifier
```

In [52]:

```
abrmodel1=AdaBoostClassifier(DecisionTreeClassifier())
```

In [53]:

```
abrmodel1.fit(x_train,y_train)
```

Out[53]:

```
AdaBoostClassifier(base_estimator=DecisionTreeClassifier())
```

In [54]:

```
abrmodel1.score(x_test,y_test)
```

Out[54]:

```
0.8333333333333334
```

In [55]:

```
from sklearn.linear_model import LogisticRegression
```

In [56]:

```
logReg1=LogisticRegression(max_iter=5000)
```

In [57]:

```
logReg1.fit(x_train,y_train)
```

Out[57]:

```
LogisticRegression(max_iter=5000)
```

In [58]:

```
logReg1.score(x_test,y_test)
```

Out[58]:

```
0.9583333333333334
```

In [59]:

```
abrmodel2=AdaBoostClassifier(LogisticRegression(max_iter=5000))
```

In [60]:

```
abrmodel2.fit(x_train,y_train)
```

Out[60]:

```
AdaBoostClassifier(base_estimator=LogisticRegression(max_iter=5000))
```

In [61]:

```
abrmodel2.score(x_test,y_test)
```

Out[61]:

```
0.9111111111111111
```

In [62]:

```
from sklearn.ensemble import AdaBoostRegressor
```

In [63]:

```
abrmodel1=AdaBoostRegressor(DecisionTreeRegressor())
```

In [64]:

```
abrmodel1.fit(x_train_1,y_train_1.values.ravel())
```

Out[64]:

```
AdaBoostRegressor(base_estimator=DecisionTreeRegressor())
```

In [65]:

```
yTestPredict=abrmodel1.predict(x_test_1)
```

In [66]:

```
np.sqrt(mean_squared_error(y_test_1,yTestPredict))
```

Out[66]:

5.937270119388897