Step 1: Load the breast_cancer dataset from sklearn.datasets using load_breast_cancer class and split it into traing and testing subsets.

In [1]:

```
from sklearn.datasets import load_breast_cancer

dataset = load_breast_cancer()
x = dataset.data
y = dataset.target
print(x.shape)
print(y.shape)
print(x[0:2])
print(y[0:2])
```

```
(569, 30)
(569,)
[[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
  1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
  6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
  1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
  4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
  7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
  5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
  2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
  2.750e-01 8.902e-02]]
[0 0]
```

In [2]:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

Step 2: Build the Decision Tree, Random Forest and AdaBoost models

--> Import DecisionTreeClassifier from sklearn.tree module

--> Import RandomForestClassifier and AdaBoostClassifier classes from sklearn.ensemble module.

--> We can use n_estimators parameter while creating models to change the default number of 10 weak models.

In [3]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier

tree = DecisionTreeClassifier()
tree.fit(x_train, y_train)
```

Out[3]:

```
DecisionTreeClassifier()
```

In [4]:

```
RFmodel = RandomForestClassifier(n_estimators = 5)
RFmodel.fit(x_train, y_train)
```

Out[4]:

```
RandomForestClassifier(n_estimators=5)
```

In [5]:

```
ABmodel = AdaBoostClassifier(n_estimators = 5)
ABmodel.fit(x_train, y_train)
```

Out[5]:

```
AdaBoostClassifier(n_estimators=5)
```

Step 3: Compare the performance of all three models

--> Import accuracy_score from sklearn.metrics module.

--> Compute both Traing and Test accuracies to observe any overfitting.

In [6]:

```
from sklearn.metrics import accuracy_score
```

In [7]:

```
print("Training Accuracies:")
print("Decision Tree Accuracy:", accuracy_score(y_train, tree.predict(x_train)))
print("Random Forest Accuracy:", accuracy_score(y_train, RFmodel.predict(x_train)))
print("AdaBoost Accuracy:", accuracy_score(y_train, ABmodel.predict(x_train)))
```

```
Training Accuracies:
Decision Tree Accuracy: 1.0
Random Forest Accuracy: 0.9953051643192489
AdaBoost Accuracy: 0.9507042253521126
```

In [8]:

```
print("Test Accuracies:")

print("Decision Tree Accuracy:", accuracy_score(y_test, tree.predict(x_test)))

print("Random Forest Accuracy:", accuracy_score(y_test, RFmodel.predict(x_test)))

print("AdaBoost Accuracy:", accuracy_score(y_test, ABmodel.predict(x_test)))
```

```
Test Accuracies:
Decision Tree Accuracy: 0.9020979020979021
Random Forest Accuracy: 0.965034965034965
AdaBoost Accuracy: 0.9370629370629371
```

Step 4: Perform the above steps with digits data set and compare the performance of Decision Tree, Random Forest and AdaBoost classifiers

In [9]:

```
from sklearn.datasets import load_digits

dataset = load_digits()
x = dataset.data
y = dataset.target
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)

tree = DecisionTreeClassifier()
tree.fit(x_train, y_train)

RFmodel = RandomForestClassifier(n_estimators = 10)
RFmodel.fit(x_train, y_train)

ABmodel = AdaBoostClassifier(n_estimators = 10)
ABmodel.fit(x_train, y_train)

print("Decision Tree Accuracy:", accuracy_score(y_test, tree.predict(x_test)))
print("Random Forest Accuracy:", accuracy_score(y_test, RFmodel.predict(x_test)))
print("AdaBoost Accuracy:", accuracy_score(y_test, ABmodel.predict(x_test)))
```

```
Decision Tree Accuracy: 0.8355555555555556
Random Forest Accuracy: 0.9422222222222222
AdaBoost Accuracy: 0.29333333333333333
```

```
In [10]:
print("Decision Tree Accuracy:", accuracy_score(y_train, tree.predict(x_train)))
print("Random Forest Accuracy:", accuracy_score(y_train, RFmodel.predict(x_train)))
print("AdaBoost Accuracy:", accuracy_score(y_train, ABmodel.predict(x_train)))
```

```
Decision Tree Accuracy: 1.0
Random Forest Accuracy: 1.0
AdaBoost Accuracy: 0.28210838901262064
```