

In [1]:

```
from sklearn.datasets import load_breast_cancer
dataset=load_breast_cancer()
```

In [2]:

```
data=dataset['data']
targetdata=dataset['target']
```

In [3]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data,targetdata,test_size=0.20,random_state=
print('Shape of x_train is: ',x_train.shape)
print('Shape of x_test is: ',x_test.shape)
print('Shape of y_train is: ',y_train.shape)
print('Shape of y_test is: ',y_test.shape)
```

```
Shape of x_train is: (455, 30)
Shape of x_test is: (114, 30)
Shape of y_train is: (455,)
Shape of y_test is: (114,)
```

In [4]:

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

In [6]:

```
dtc = DecisionTreeClassifier()
model = BaggingClassifier(base_estimator=dtc,n_estimators= 100 ,random_state= 42)
results = cross_val_score(model, data, targetdata, cv= 10)
print(results)
print(results.mean())
```

```
[0.94736842 0.89473684 0.9122807  0.92982456 0.98245614 0.98245614
 0.94736842 0.98245614 0.98245614 1.          ]
0.956140350877193
```

## AdaBoost Classification

In [7]:

```
# AdaBoost Classification

from sklearn.ensemble import AdaBoostClassifier

model = AdaBoostClassifier(n_estimators=100, random_state= 42)
results = cross_val_score(model, data, targetdata, cv= 10)
print(results.mean())
```

0.9683270676691729

## Stacking

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.naive_bayes import GaussianNB

# create the sub models
estimators = []
model1 = GaussianNB()
estimators.append(('Naive_Bais', model1))
model2 = DecisionTreeClassifier()
estimators.append(('cart', model2))
model3 = SVC()
estimators.append(('svm', model3))
# create the ensemble model
ensemble = VotingClassifier(estimators)
results = cross_val_score(ensemble, data, targetdata, cv= 10)
print(results.mean())
```

0.9526002506265664

In [32]:

```
ensemble
```

Out[32]:

```
VotingClassifier(estimators=[('Naive_Bais', GaussianNB()),
                             ('cart', DecisionTreeClassifier()),
                             ('svm', SVC())])
```

In [33]:

```
results
```

Out[33]:

```
array([0.96491228, 0.87719298, 0.9122807 , 0.92982456, 0.98245614,
        0.96491228, 0.92982456, 0.96491228, 0.98245614, 0.98214286])
```

# GradientBoostingClassifier

In [9]:

```
# importing machine learning models for prediction
from sklearn.ensemble import GradientBoostingClassifier
```

In [10]:

```
# initializing the boosting module with default parameters
model = GradientBoostingClassifier()
#model = AdaBoostClassifier(n_estimators=100, random_state= 42)
results = cross_val_score(model, data, targetdata, cv= 10)
print(results.mean())
```

0.9614035087719298

In [ ]:

In [ ]:

In [ ]:

In [ ]: