In [4]:
```
from sklearn.datasets import load_breast_cancer
dataset = load_breast_cancer()
```

In [7]:
```
data = dataset['data']
targetdata = dataset['target']
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, targetdata, random_state=7)
```

In [9]:
```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

In [10]:
```
dtc = DecisionTreeClassifier()
model = BaggingClassifier(base_estimator=dtc,n_estimators=100)
results = cross_val_score(model, data, targetdata, cv = 10)
print(results.mean())
```

```
0.956140350877193
```

In [ ]:
```

```

In [11]:
```
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
```

In [17]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

```
from sklearn.ensemble import VotingClassifier
from sklearn.naive_bayes import GaussianNB
```

In [18]:
```
# 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 [19]:
```
ensemble
```

Out[19]:
```
VotingClassifier(estimators=[('Naive_Bais', GaussianNB()),
                             ('cart', DecisionTreeClassifier()),
                             ('svm', SVC())])
```

In [20]:
```
from sklearn.ensemble import GradientBoostingClassifier
```

In [21]:
```
model = GradientBoostingClassifier()
results = cross_val_score(model, data, targetdata, cv = 10)
print(results.mean())
```

```
0.9614035087719298
```

In [ ]:

In [ ]: