In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
path = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
```

In [3]:
```
headernames = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Cla
```

In [4]:
```
D= pd.read_csv(path,names = headernames)
```

In [5]:
```
D.head()
```

Out[5]:

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [8]:
```
x=D.iloc[:,0:4].values
y=D.iloc[:,4].values
```

In [11]:
```
from sklearn.model_selection import train_test_split
#Train and Test split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [12]:
```
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: (105, 4)
shape of x_test is: (45, 4)
shape of y_train is: (105,)
shape of y_test is: (45,)
```

In [13]:
```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

In [15]:
```
#BaggingClassifier
dtc=DecisionTreeClassifier() #object of DTC
model=BaggingClassifier(base_estimator=dtc,n_estimators=100,random_state=42)
result=cross_val_score(model,x,y,cv=10)
print(result.mean())
```

```
0.96
```

In [16]:
```
#AdaBoost classification
from sklearn.ensemble import AdaBoostClassifier
model=AdaBoostClassifier(n_estimators=100,random_state=42)
result=cross_val_score(model,x,y,cv=10)
print(result.mean())
```

```
0.9466666666666667
```

In [18]:
```
#STACKING:
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.naive_bayes import GaussianNB

#create sub model for classifiers
estimators=[] #it take list of algo nd it apply in voting classifier
model1 = GaussianNB()
estimators.append(('naive_Bais',model1))
model2 = DecisionTreeClassifier()
estimators.append(('cart',model2))
model3 = SVC()
estimators.append(('svm',model2))
```

In [21]:
```
#create the ensemble model
ensemble = VotingClassifier(estimators)
results= cross_val_score(ensemble,x,y,cv=10)
print(result.mean())
```

```
0.9466666666666667
```

In [22]:
```
#GradientBoostingClassifier
from sklearn.ensemble import GradientBoostingClassifier
model=GradientBoostingClassifier(n_estimators=100,random_state=42)
result=cross_val_score(model,x,y,cv=10)
print(result.mean())
```

```
0.96
```

In [ ]:
```
```