In [31]:
```
Aim: To implement Bagging
```

In [ ]:
```
import numpy as nm
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from matplotlib.colors import ListedColormap
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn import model_selection
from sklearn.ensemble import BaggingClassifier
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings('ignore')
```

In [21]:
```
dataset = pd.read_csv('Iris.csv')
```

In [22]:
```
x = dataset.iloc[:, [0,1,2,3]].values
y = dataset.iloc[:, 4].values
```

In [23]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.25, random_state = 0)
```

In [24]:
```
Single = GaussianNB()
Single.fit(xtrain, ytrain)
```

Out[24]:
```
▾ GaussianNB

GaussianNB()
```

In [25]:
```
y_pred = Single.predict(xtest)
print("Predicted values for single Naïve Bayes Classifier:")
y_pred
```

```
Predicted values for single Naïve Bayes Classifier:
```

Out[25]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor'], dtype='<U15')
```

In [26]:
```
Acc_Single= accuracy_score(ytest,y_pred)*100
print ("\n\nAccuracy using single Naïve Bayes Classifier: ",Acc_Single)
```

```
Accuracy using single Naïve Bayes Classifier:  100.0
```

In [27]:
```
cm = confusion_matrix(ytest, y_pred)
print ("\n\n Confusion Matrix -using single Naïve Bayes Classifier: \n", cm)
```

```
 Confusion Matrix -using single Naïve Bayes Classifier:
 [[13  0  0]
 [ 0 16  0]
 [ 0  0  9]]
```

In [28]:
```
base_cls = GaussianNB()
num_class = 100
```

In [29]:
```
Bag = BaggingClassifier(base_estimator = base_cls, n_estimators = num_class, random_state = 0)
Bag.fit(xtrain, ytrain)
```

Out[29]:

BaggingClassifier

BaggingClassifier(base_estimator=GaussianNB(), n_estimators=100, random_state=0)

base_estimator: GaussianNB

GaussianNB()

GaussianNB

GaussianNB()

In [30]:
```
results = model_selection.cross_val_score(Bag, xtest, ytest, cv = 10)
print("\n\nAccuracy using Bagged Set of Naïve Bayes Classifiers :", results.mean()*100)
```

```
Accuracy using Bagged Set of Naïve Bayes Classifiers : 94.16666666666667
```

In [ ]: