In [1]:
```
#Vedanshu Patel
#20BCE0865
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.ensemble import BaggingClassifier
```

In [2]:
```
dataset = pd.read_csv("processed.cleveland.data.csv", names=['age', 'sex', 'cp',
```

In [3]:
```
#Data Preprocessing

#Filling missing values Statistics measures
print("*****Before Filling Missing values Row 166, 192, 287, 302*****")
print(dataset.loc[287])
dataset1 = dataset
df1 = pd.DataFrame(dataset1)

print("----- Mean of Column 11 'ca' -----")
print(df1['ca'].mean())
df1.fillna(df1.mean(), inplace=True)
print("*****After Filling Missing values Row 166, 192, 287, 302*****")
print(df1.loc[[166, 192, 287, 302]])

print("----- Mean of Column 12 'thal' -----")
print(df1['thal'].mean())
df1.fillna(df1.mean(), inplace=True)
print("*****After Filling Missing values Row 87, 266*****")
print(df1.loc[[87, 266]])
```

```
*****Before Filling Missing values Row 166, 192, 287, 302*****
age          58.0
sex           1.0
cp            2.0
trestbps    125.0
chol        220.0
fbs           0.0
restecg       0.0
thalach     144.0
exang         0.0
oldpeak       0.4
slope         2.0
ca            NaN
thal          7.0
output        0.0
Name: 287, dtype: float64
----- Mean of Column 11 'ca' -----
0.6722408026755853
*****After Filling Missing values Row 166, 192, 287, 302*****
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
166   52    1   3       138   223    0        0      169      0      0.0
192   43    1   4       132   247    1        2      143      1      0.1
287   58    1   2       125   220    0        0      144      0      0.4
302   38    1   3       138   175    0        0      173      0      0.0

     slope        ca  thal  output
166      1  0.672241   3.0       0
192      2  0.672241   7.0       1
287      2  0.672241   7.0       0
302      1  0.672241   3.0       0
----- Mean of Column 12 'thal' -----
4.73421926910299
*****After Filling Missing values Row 87, 266*****
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
87    53    0   3       128   216    0        2      115      0      0.0
266   52    1   4       128   204    1        0      156      1      1.0
```

```
     slope   ca      thal  output
87       1  0.0  4.734219       0
266      2  0.0  4.734219       2
```

In [4]:
```
feature_cols = list(dataset.columns[0:13])

print("Feature columns: \n{}".format(feature_cols))
```

```
Feature columns:
['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope', 'ca', 'thal']
```

In [5]:
```
X = dataset[feature_cols]
y = dataset['output'].values
```

In [6]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_
print(X_train)
```

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
3     37    1   3       130   250    0        0      187      0      3.5
55    54    1   4       124   266    0        2      109      1      2.2
225   34    0   2       118   210    0        0      192      0      0.7
224   63    0   4       108   269    0        0      169      1      1.8
75    65    0   3       160   360    0        2      151      0      0.8
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
8     63    1   4       130   254    0        2      147      0      1.4
73    65    1   4       110   248    0        2      158      0      0.6
118   63    1   4       130   330    1        2      132      1      1.8
189   69    1   3       140   254    0        2      146      0      2.0
206   58    1   4       128   259    0        2      130      1      3.0

     slope   ca  thal
3        3  0.0   3.0
55       2  1.0   7.0
225      1  0.0   3.0
224      2  2.0   3.0
75       1  0.0   3.0
..     ...  ...   ...
8        2  1.0   7.0
73       1  2.0   6.0
118      1  3.0   7.0
189      2  3.0   7.0
206      2  2.0   7.0

[212 rows x 13 columns]
```

In [7]:
```
cl1 = RandomForestClassifier(n_estimators = 50, random_state = 1)
cl1.fit(X_train, y_train)
ypred = cl1.predict(X_test)

cm1 = confusion_matrix(y_test, ypred)
print("Accuracy :", accuracy_score(ypred, y_test))
```

```
Accuracy : 0.6153846153846154
```

In [8]:
```
cl1 = AdaBoostClassifier(n_estimators = 50)
cl1.fit(X_train, y_train)
ypred = cl1.predict(X_test)

cm1 = confusion_matrix(y_test, ypred)
print("Accuracy :", accuracy_score(ypred, y_test))
```

```
Accuracy : 0.5274725274725275
```

In [ ]:
```

```