Heart Disease Prediction By Sabyasachi Bandyopadhyay

In [ ]:

Dataset

The dataset has 14 attributes:

- **age:** age in years.
- **sex:** sex (1 = male; 0 = female).
- **cp:** chest pain type (Value 0: typical angina; Value 1: atypical angina; Value 2: non-anginal pain; Value 3: asymptomatic).
- **trestbps:** resting blood pressure in mm Hg on admission to the hospital.
- **chol:** serum cholesterol in mg/dl.
- **fbs:** fasting blood sugar > 120 mg/dl (1 = true; 0 = false).
- **restecg:** resting electrocardiographic results (Value 0: normal; Value 1: having ST-T wave abnormality; Value 2: probable or definite left ventricular hypertrophy).
- **thalach:** maximum heart rate achieved.
- **exang:** exercise induced angina (1 = yes; 0 = no)
- **oldpeak:** ST depression induced by exercise relative to rest.
- **slope:** the slope of the peak exercise ST segment (Value 0: upsloping; Value 1: flat; Value 2: downsloping).
- **ca:** number of major vessels (0-3) colored by flourosopy.
- **thal:** thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect).
- **target:** heart disease (1 = no, 2 = yes)

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from IPython.display import display
from sklearn.preprocessing import LabelEncoder, MinMaxScaler, StandardScaler
import pickle
```

In [3]:
```
df = pd.read_csv('heart.csv')
print('NO OF ROWS AND COLUMN IN DATASET ',df.shape,'\n')
display(df.head(),'\n',df.dtypes)
```

```
NO OF ROWS AND COLUMN IN DATASET  (303, 14)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
'\n'

age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [4]:
```
print('COUNT NO OF NULL IN EACH COLUMN','\n',df.isnull().sum())
```

```
COUNT NO OF NULL IN EACH COLUMN
 age         0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
thalach      0
exang        0
oldpeak      0
slope        0
ca           0
thal         0
target       0
dtype: int64
```

In [5]:
```
df.describe().T
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
sex	303.0	0.683168	0.466011	0.0	0.0	1.0	1.0	1.0
cp	303.0	0.966997	1.032052	0.0	0.0	1.0	2.0	3.0
trestbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
fbs	303.0	0.148515	0.356198	0.0	0.0	0.0	0.0	1.0
restecg	303.0	0.528053	0.525860	0.0	0.0	1.0	1.0	2.0
thalach	303.0	149.646865	22.905161	71.0	133.5	153.0	166.0	202.0
exang	303.0	0.326733	0.469794	0.0	0.0	0.0	1.0	1.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2
slope	303.0	1.399340	0.616226	0.0	1.0	1.0	2.0	2.0
ca	303.0	0.729373	1.022606	0.0	0.0	0.0	1.0	4.0
thal	303.0	2.313531	0.612277	0.0	2.0	2.0	3.0	3.0
target	303.0	0.544554	0.498835	0.0	0.0	1.0	1.0	1.0

In [6]:
```
data = df.copy(deep = True)
## showing the count of Nans
print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [7]:
```
#corr=data.corr()
#corr.nlargest(15,'target')['target']
print(data.corr()["target"].abs().sort_values(ascending=False))
```

```
target      1.000000
exang       0.436757
cp          0.433798
oldpeak     0.430696
thalach     0.421741
ca          0.391724
slope       0.345877
thal        0.344029
sex         0.280937
age         0.225439
trestbps    0.144931
restecg     0.137230
chol        0.085239
fbs         0.028046
Name: target, dtype: float64
```

age,sex,cp=Chest pain ,trestbps=resting blood pressure ,chol=cholesterol,fbs=fasting blood suger,

restecg= resting electrocardiographic, ca=number of major vessels colored by fluoroscopy.

thalch= maximum heart rate achived,exang=exercise include angina,slope.

In [8]:
```
x = data[['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','slope','ca','thal']]
y = data['target']
x.head()
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	0	0	1
1	37	1	2	130	250	0	1	187	0	0	0	2
2	41	0	1	130	204	0	0	172	0	2	0	2
3	56	1	1	120	236	0	1	178	0	2	0	2
4	57	0	0	120	354	0	1	163	1	2	0	2

In [9]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

print('X TRAIN DATA ', x_train.shape)
print('X TEST DATA ', x_test.shape)
print('Y TRAIN DATA ', y_train.shape)
print('Y TEST DATA ', y_test.shape)
```

```
X TRAIN DATA  (242, 12)
X TEST DATA  (61, 12)
Y TRAIN DATA  (242,)
Y TEST DATA  (61,)
```

In [10]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=20)
clf.fit(x_train, y_train)
```

Out[10]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=20,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [11]:
```
y_pred = clf.predict(x_test)
```

In [12]:
```
from sklearn.metrics import confusion_matrix

print(confusion_matrix(y_test,y_pred))
```

```
[[23  4]
 [ 6 28]]
```

In [13]:
```
from sklearn.metrics import classification_report

print('CLASSIFICATION REPORT','\n',classification_report(y_test, y_pred))
```

```
CLASSIFICATION REPORT
               precision    recall  f1-score   support

           0       0.79      0.85      0.82        27
           1       0.88      0.82      0.85        34

    accuracy                           0.84        61
   macro avg       0.83      0.84      0.83        61
weighted avg       0.84      0.84      0.84        61
```

In [14]:
```
filename = open('heartdiseasespredictmodel.pkl', 'wb')
pickle.dump(clf,filename )
```

In [15]:
```
filename.close()
```