In [1]:
```
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 [2]:
```
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 [3]:
```
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 [4]:
```
df.describe().T
```

Out[4]:

	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 [5]:
```
data = df.copy(deep = True)
##data[['age','sex','cp','trestbps','chol','fbs','restecg','thalch','exang','oldpeak','slope','ca','thal']] = data[['age','sex',

## 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 [6]:
```
corr=data.corr()
corr.nlargest(15,'target')['target']
```

Out[6]:
```
target      1.000000
cp          0.433798
thalach     0.421741
slope       0.345877
restecg     0.137230
fbs        -0.028046
chol       -0.085239
trestbps   -0.144931
age        -0.225439
sex        -0.280937
thal       -0.344029
ca         -0.391724
oldpeak    -0.430696
exang      -0.436757
Name: target, dtype: float64
```

age,cp=cerebral palsy,chol=cholesterol,fbs=fasting bloodpressure,

restecg= resting electrocardiographic,

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

In [7]:
```
x = data[['age','cp','chol','fbs','restecg','thalach','exang','slope']]
y = data['target']
x.head()
```

Out[7]:

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

In [8]:
```
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, 8)
X TEST DATA  (61, 8)
Y TRAIN DATA  (242,)
Y TEST DATA  (61,)
```

In [9]:
```
#from sklearn.preprocessing import StandardScaler

#sc = StandardScaler()
#xtrain = sc.fit_transform(x_train)
#xtest = sc.transform(x_test)
```

In [10]:
```
from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()

clf.fit(x_train, y_train)
```

Out[10]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [11]:
```
filename = open('heartdiseasespredictmodel.pkl', 'wb')
pickle.dump(clf,filename )
```

In [12]:
```
filename.close()
```

In [13]:
```
y_pred = clf.predict(x_test)
```

In [14]:
```
from sklearn.metrics import confusion_matrix

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

```
[[19  8]
 [ 5 29]]
```

In [15]:
```
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      0.70      0.75        27
           1       0.78      0.85      0.82        34

    accuracy                           0.79        61
   macro avg       0.79      0.78      0.78        61
weighted avg       0.79      0.79      0.79        61
```

In [16]:
```
#from sklearn.ensemble import RandomForestClassifier
#clf = RandomForestClassifier(n_estimators=20)
#clf.fit(xtrain, y_train)
```

In [17]:
```
#y_pred = clf.predict(xtest)
```

In [18]:
```
#from sklearn.metrics import confusion_matrix

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

In [19]:
```
#from sklearn.metrics import classification_report

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