# Untitled6

June 5, 2020

## 0.1 HEART DISEASE CLASSIFICATION USING SUPPORT VECTOR MACHINE (SVM)

### 0.1.1 IMPORTING THE LIABRARIES

[1]:
```
#importing the liabraries.....
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

### 0.1.2 IMPORTING THE DATASET

[2]:
```
#Reading the dataset
ds=pd.read_csv('heart.csv')
print(ds)
```

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   …   …  ..       …   …  …      …      …    …      …
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     …  ..   …     …
298      1   0     3       0
```

```
299        1    0      3        0
300        1    2      3        0
301        1    1      3        0
302        1    1      2        0

[303 rows x 14 columns]
```

[3]:
```python
ds.head()
```

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

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
```

[4]:
```python
ds.tail()
```

[4]:
```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0
```

[5]:
```python
#splitting the dataset into independent and dependent variables
X = ds.iloc[:,:-1].values
y = ds.iloc[:,-1].values
print(X)
print(y)
```

```
[[63.  1.  3. …  0.  0.  1.]
 [37.  1.  2. …  0.  0.  2.]
 [41.  0.  1. …  2.  0.  2.]
```

```
 …
 [68.  1.  0. …  1.  2.  3.]
 [57.  1.  0. …  1.  1.  3.]
 [57.  0.  1. …  1.  1.  2.]]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0]
```

### 0.2 FEATURE SCALING

[6]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
print(X)
```

```
[[ 0.9521966   0.68100522  1.97312292 … -2.27457861 -0.71442887
  -2.14887271]
 [-1.91531289  0.68100522  1.00257707 … -2.27457861 -0.71442887
  -0.51292188]
 [-1.47415758 -1.46841752  0.03203122 …  0.97635214 -0.71442887
  -0.51292188]
 …
 [ 1.50364073  0.68100522 -0.93851463 … -0.64911323  1.24459328
   1.12302895]
 [ 0.29046364  0.68100522 -0.93851463 … -0.64911323  0.26508221
   1.12302895]
 [ 0.29046364 -1.46841752  0.03203122 … -0.64911323  0.26508221
  -0.51292188]]
```

#### 0.2.1 SPLITTING THE DATASET INTO TRAINING SET AND TEST SET

[7]:
```
from sklearn.model_selection import train_test_split
X_train,x_test ,y_train, y_test = train_test_split(X, y, test_size=0.
 ↪25,random_state=5)
```

[8]:
```
print(X_train)
```

```
[[ 0.40075247 -1.46841752 -0.93851463 … -0.64911323  1.24459328
  -2.14887271]
 [ 0.18017482 -1.46841752  0.03203122 … -0.64911323 -0.71442887
```

```
 -0.51292188]
 [ 0.9521966  -1.46841752  0.03203122 …  0.97635214  1.24459328
 -0.51292188]
 …
 [-0.92271345 -1.46841752  0.03203122 …  0.97635214 -0.71442887
 -0.51292188]
 [-1.47415758  0.68100522 -0.93851463 …  0.97635214 -0.71442887
  1.12302895]
 [ 0.5110413   0.68100522 -0.93851463 … -0.64911323  0.26508221
  1.12302895]]
```

[9]:
```python
print(y_train)
```

```
[0 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 1 0 0 1 1 0 1
 0 1 1 1 0 1 1 1 1 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1
 0 1 1 1 0 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0
 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0
 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 1 0 0
 0 1 1 0 1 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0
 1 1 1 0 0]
```

[10]:
```python
print(x_test)
```

```
[[ 0.29046364 -1.46841752  0.03203122 -0.09273778 -0.19835726 -0.41763453
  -1.00583187  1.0649749  -0.69663055 -0.89686172 -0.64911323  0.26508221
  -0.51292188]
 [ 0.29046364  0.68100522 -0.93851463  1.1637461   0.53601107 -0.41763453
   0.89896224 -2.69584852  1.43548113  0.13837295 -0.64911323  0.26508221
   1.12302895]
 [-0.04040284  0.68100522 -0.93851463 -0.54964101  0.76791685 -0.41763453
  -1.00583187 -1.47139438  1.43548113  1.86376406 -0.64911323  1.24459328
  -0.51292188]
 [ 0.73161895 -1.46841752 -0.93851463  0.76395577  1.17375198 -0.41763453
  -1.00583187 -0.15947923  1.43548113 -0.03416616 -0.64911323 -0.71442887
   1.12302895]
 [-1.58444641  0.68100522  1.97312292  0.47839125 -0.91340011 -0.41763453
   0.89896224  1.23989692  1.43548113  0.31091206  0.97635214 -0.71442887
   1.12302895]
 [ 0.73161895  0.68100522 -0.93851463  0.47839125 -0.75879625 -0.41763453
  -1.00583187 -0.50932327  1.43548113  0.74225984  0.97635214  0.26508221
   1.12302895]
 [ 0.18017482  0.68100522  0.03203122 -0.09273778 -0.4882395  -0.41763453
  -1.00583187  0.58393935 -0.69663055 -0.89686172  0.97635214 -0.71442887
   1.12302895]
 [ 0.9521966   0.68100522 -0.93851463  0.47839125 -1.14530589 -0.41763453
  -1.00583187 -0.24694024  1.43548113  2.55392051  0.97635214  1.24459328
   1.12302895]
 [-0.37126932 -1.46841752  1.00257707 -0.09273778  0.18815239 -0.41763453
```

```
 -1.00583187 -0.02828772 -0.69663055 -0.46551394  0.97635214 -0.71442887
 -0.51292188]
[-1.80502406  0.68100522  1.00257707  0.36416545 -1.37721168 -0.41763453
  0.89896224  1.0212444  -0.69663055 -0.89686172  0.97635214  3.20361543
 -0.51292188]
[-0.15069166 -1.46841752 -0.93851463 -0.09273778  0.34275624 -0.41763453
 -1.00583187 -0.29067075 -0.69663055 -0.55178349 -0.64911323 -0.71442887
 -0.51292188]
[ 0.5110413   0.68100522 -0.93851463  2.19177836  1.54093615 -0.41763453
 -1.00583187 -0.42186226  1.43548113  2.03630317 -2.27457861 -0.71442887
  1.12302895]
[-0.7021358   0.68100522  1.00257707 -0.43541521  0.1688269   2.394438
  0.89896224  1.10870541 -0.69663055 -0.89686172  0.97635214  1.24459328
 -0.51292188]
[-1.69473524 -1.46841752  1.00257707  0.36416545 -0.50756498 -0.41763453
  0.89896224  0.1029038  -0.69663055 -0.89686172 -0.64911323 -0.71442887
 -0.51292188]
[ 1.39335191  0.68100522 -0.93851463 -0.66386682 -0.17903178 -0.41763453
  0.89896224 -3.4392671  -0.69663055 -0.03416616 -0.64911323 -0.71442887
 -0.51292188]
[ 0.84190778 -1.46841752 -0.93851463  0.47839125  0.42005817 -0.41763453
 -1.00583187  0.45274783 -0.69663055  2.20884228 -2.27457861  1.24459328
 -0.51292188]
[-0.92271345  0.68100522 -0.93851463 -0.66386682  0.05287401 -0.41763453
 -1.00583187 -0.24694024 -0.69663055 -0.20670527  0.97635214 -0.71442887
  1.12302895]
[-1.1432911   0.68100522 -0.93851463 -1.23499586 -0.95205107 -0.41763453
 -1.00583187  1.19616642 -0.69663055 -0.89686172  0.97635214  0.26508221
 -0.51292188]
[ 1.83450721 -1.46841752  1.00257707 -1.23499586  0.36208173  2.394438
 -1.00583187 -0.85916731 -0.69663055 -0.89686172  0.97635214  0.26508221
 -0.51292188]
[ 0.06988599 -1.46841752 -0.93851463 -0.20696359 -0.79744721 -0.41763453
  2.80375634 -0.85916731  1.43548113  0.82852939 -0.64911323  0.26508221
  1.12302895]
[ 0.40075247  0.68100522 -0.93851463 -0.20696359 -0.58486691 -0.41763453
 -1.00583187 -0.81543681  1.43548113  1.00106851 -0.64911323  2.22410436
  1.12302895]
[ 0.62133012  0.68100522 -0.93851463 -0.09273778  0.13017594 -0.41763453
  0.89896224 -0.24694024  1.43548113  0.31091206  0.97635214  0.26508221
  1.12302895]
[-0.26098049  0.68100522  1.97312292 -0.77809263 -1.16463138 -0.41763453
 -1.00583187  1.76466298 -0.69663055 -0.89686172 -0.64911323 -0.71442887
 -2.14887271]
[ 0.9521966  -1.46841752 -0.93851463  1.04952029  3.10630021 -0.41763453
 -1.00583187  0.19036481 -0.69663055  2.55392051 -0.64911323  2.22410436
  1.12302895]
[-1.03300228  0.68100522 -0.93851463 -1.57767328 -0.73947076 -0.41763453
```

```
 -1.00583187 -0.07201822  1.43548113  1.69122495 -0.64911323 -0.71442887
 -0.51292188]
[-0.7021358   0.68100522 -0.93851463 -0.54964101 -0.46891401 -0.41763453
 -1.00583187  1.58974096 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-0.04040284  0.68100522  0.03203122 -1.34922166  1.21240295 -0.41763453
  0.89896224  0.27782582 -0.69663055 -0.89686172  0.97635214 -0.71442887
  1.12302895]
[-0.04040284 -1.46841752  1.00257707 -1.23499586 -0.62351787 -0.41763453
  0.89896224  0.36528682 -0.69663055  0.48345117 -0.64911323 -0.71442887
 -0.51292188]
[-0.15069166 -1.46841752  1.00257707 -0.20696359 -0.58486691 -0.41763453
 -1.00583187 -1.51512489 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -3.78482354]
[-1.25357993  0.68100522 -0.93851463 -0.66386682 -1.33856072 -0.41763453
 -1.00583187 -1.29647236  1.43548113  1.25987717 -0.64911323 -0.71442887
  1.12302895]
[ 1.39335191 -1.46841752  1.00257707  1.1637461   0.59398751 -0.41763453
  0.89896224  0.97751389 -0.69663055 -0.89686172  0.97635214  0.26508221
 -0.51292188]
[-1.36386876  0.68100522  0.03203122 -0.66386682  0.9418462  -0.41763453
  0.89896224  0.54020884 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-0.04040284  0.68100522  1.00257707 -0.3783023   0.51668558 -0.41763453
 -1.00583187  0.1029038  -0.69663055 -0.46551394 -2.27457861  0.26508221
 -0.51292188]
[-0.04040284 -1.46841752  0.03203122  0.02148802  0.80656782  2.394438
 -1.00583187  0.40901733  1.43548113 -0.89686172  0.97635214  0.26508221
 -0.51292188]
[ 0.62133012  0.68100522 -0.93851463 -0.83520553 -0.31431015  2.394438
  0.89896224  0.45274783  1.43548113  0.31091206  0.97635214  1.24459328
  1.12302895]
[ 1.72421839  0.68100522 -0.93851463 -0.09273778  1.46363422 -0.41763453
 -1.00583187 -1.77750792 -0.69663055  1.17360762 -0.64911323  2.22410436
 -0.51292188]
[-0.15069166 -1.46841752 -0.93851463  0.36416545 -0.23700823 -0.41763453
 -1.00583187  0.45274783 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-1.1432911   0.68100522  1.00257707 -0.09273778 -0.25633371 -0.41763453
  0.89896224  1.28362743  1.43548113 -0.55178349  0.97635214 -0.71442887
 -0.51292188]
[ 1.06248543  0.68100522 -0.93851463  0.76395577 -0.66216884 -0.41763453
 -1.00583187 -0.7717063  -0.69663055  0.82852939 -0.64911323  1.24459328
 -2.14887271]
[ 0.84190778 -1.46841752 -0.93851463  0.47839125  2.85506894 -0.41763453
 -1.00583187  0.32155632 -0.69663055  0.13837295 -0.64911323 -0.71442887
 -0.51292188]
[ 0.18017482  0.68100522  1.97312292 -0.66386682 -1.029353   -0.41763453
```

```
 -1.00583187  0.54020884 -0.69663055  0.74225984 -0.64911323 -0.71442887
  1.12302895]
[-0.15069166  0.68100522 -0.93851463  0.59261706 -0.39161208 -0.41763453
 -1.00583187 -1.69004691  1.43548113 -0.89686172  0.97635214 -0.71442887
  1.12302895]
[-0.37126932  0.68100522  1.00257707 -1.80612489 -0.46891401 -0.41763453
  0.89896224 -0.29067075  1.43548113  0.13837295 -0.64911323 -0.71442887
 -0.51292188]
[ 0.84190778  0.68100522 -0.93851463 -0.66386682  0.40073269 -0.41763453
  0.89896224 -2.21481297  1.43548113  0.65599028 -0.64911323  1.24459328
  1.12302895]
[ 2.49624017  0.68100522 -0.93851463 -0.3783023   1.11577554 -0.41763453
 -1.00583187  0.54020884  1.43548113 -0.89686172  0.97635214  2.22410436
 -0.51292188]
[ 0.18017482 -1.46841752 -0.93851463  0.13571383  3.14495118 -0.41763453
 -1.00583187  0.01544279  1.43548113  0.74225984 -0.64911323  1.24459328
  1.12302895]
[ 0.62133012  0.68100522 -0.93851463  0.76395577  0.69061493 -0.41763453
 -1.00583187 -0.33440125  1.43548113  1.51868584 -0.64911323  1.24459328
  1.12302895]
[ 1.17277425 -1.46841752 -0.93851463  1.04952029 -0.41093757 -0.41763453
 -1.00583187 -1.55885539 -0.69663055 -0.03416616 -0.64911323  2.22410436
  1.12302895]
[-0.7021358   0.68100522 -0.93851463 -0.09273778  0.18815239  2.394438
 -1.00583187  0.01544279  1.43548113 -0.89686172  0.97635214  1.24459328
  1.12302895]
[ 0.18017482 -1.46841752 -0.93851463  3.90516547  0.80656782  2.394438
 -1.00583187 -0.7279758   1.43548113  2.55392051 -2.27457861  1.24459328
  1.12302895]
[ 0.18017482  0.68100522 -0.93851463 -0.09273778  0.70994041  2.394438
 -1.00583187 -2.03989095  1.43548113  0.48345117 -2.27457861 -0.71442887
  1.12302895]
[-1.25357993  0.68100522  1.00257707 -0.09273778  1.32835584 -0.41763453
  0.89896224  0.54020884 -0.69663055  0.74225984  0.97635214  0.26508221
 -0.51292188]
[ 1.17277425 -1.46841752  1.00257707  1.62064933  2.19800254 -0.41763453
 -1.00583187  0.05917329 -0.69663055 -0.20670527  0.97635214 -0.71442887
 -0.51292188]
[-1.36386876 -1.46841752 -0.93851463 -1.69189909  0.36208173 -0.41763453
 -1.00583187 -1.20901135 -0.69663055 -0.37924438 -0.64911323 -0.71442887
 -0.51292188]
[-1.91531289 -1.46841752  1.00257707 -0.66386682 -0.60419239 -0.41763453
  0.89896224  0.89005288 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[ 1.72421839  0.68100522  0.03203122  1.39219771 -0.02442792 -0.41763453
 -1.00583187 -0.29067075 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[ 1.83450721 -1.46841752  0.03203122  1.62064933  1.07712457 -0.41763453
```

```
  0.89896224  0.54020884 -0.69663055 -0.55178349  0.97635214  1.24459328
 -0.51292188]
[-1.80502406  0.68100522  1.97312292 -0.66386682 -0.29498467 -0.41763453
  0.89896224  1.41481894  1.43548113  2.38138139 -0.64911323 -0.71442887
  1.12302895]
[-0.92271345  0.68100522  1.00257707  1.04952029 -0.29498467 -0.41763453
  0.89896224 -0.11574873 -0.69663055  2.20884228 -0.64911323 -0.71442887
 -0.51292188]
[-1.36386876  0.68100522  1.00257707 -0.09273778 -1.28058427 -0.41763453
  0.89896224  0.01544279 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-1.69473524  0.68100522 -0.93851463 -0.77809263 -0.52689046 -0.41763453
  0.89896224 -0.42186226 -0.69663055  0.13837295 -0.64911323 -0.71442887
  1.12302895]
[ 0.06988599 -1.46841752 -0.93851463  2.7629074   1.56026163 -0.41763453
  2.80375634 -1.42766388  1.43548113  2.03630317 -0.64911323 -0.71442887
 -0.51292188]
[-1.25357993 -1.46841752 -0.93851463  0.02148802  1.83081838  2.394438
 -1.00583187 -0.59678428  1.43548113  1.69122495 -0.64911323 -0.71442887
  1.12302895]
[-0.15069166  0.68100522  1.00257707 -0.09273778 -0.00510244  2.394438
 -1.00583187  1.0212444  -0.69663055 -0.89686172  0.97635214  2.22410436
 -0.51292188]
[-0.92271345 -1.46841752 -0.93851463  0.36416545 -0.06307888 -0.41763453
 -1.00583187  0.1029038   1.43548113 -0.89686172 -0.64911323 -0.71442887
 -0.51292188]
[-0.7021358  -1.46841752  1.00257707 -0.09273778  0.55533655 -0.41763453
  0.89896224 -0.46559277 -0.69663055 -0.7243226   0.97635214 -0.71442887
 -0.51292188]
[-0.15069166  0.68100522 -0.93851463 -0.49252811  0.69061493 -0.41763453
  0.89896224 -2.38973499  1.43548113  0.82852939 -0.64911323  1.24459328
  1.12302895]
[-1.03300228 -1.46841752  0.03203122 -0.09273778 -0.23700823 -0.41763453
 -1.00583187  1.10870541 -0.69663055 -0.37924438 -0.64911323 -0.71442887
 -0.51292188]
[-1.1432911   0.68100522  0.03203122 -0.66386682 -0.50756498 -0.41763453
  0.89896224  0.89005288 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[ 0.29046364  0.68100522 -0.93851463 -0.09273778 -2.2275329  -0.41763453
  0.89896224 -1.51512489  1.43548113  0.13837295 -0.64911323  0.26508221
  1.12302895]
[ 1.39335191  0.68100522 -0.93851463 -0.3783023   0.14950142  2.394438
  0.89896224  0.58393935 -0.69663055 -0.7243226  -0.64911323  1.24459328
  1.12302895]
[-0.26098049  0.68100522 -0.93851463 -0.20696359 -0.81677269  2.394438
  0.89896224  0.27782582  1.43548113 -0.03416616 -0.64911323 -0.71442887
 -3.78482354]
[-0.37126932 -1.46841752  1.00257707 -0.66386682  0.9418462  -0.41763453
```

```
 -1.00583187  0.32155632 -0.69663055 -0.37924438  0.97635214 -0.71442887
 -0.51292188]
[ 1.06248543  0.68100522 -0.93851463 -0.66386682 -0.00510244 -0.41763453
 -1.00583187 -2.34600448  1.43548113  1.00106851 -2.27457861  0.26508221
 -0.51292188]
[ 1.39335191  0.68100522 -0.93851463 -0.66386682 -0.33363564 -0.41763453
 -1.00583187 -0.90289782  1.43548113  1.34614673 -0.64911323  1.24459328
  1.12302895]
[-2.13589054  0.68100522  0.03203122 -0.54964101 -1.04867848 -0.41763453
  0.89896224  1.0649749  -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]]
```

[11]:

```
print(y_test)
```

```
[0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1
 1 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1 0
 0 1]
```

## 0.3 CREATING THE MODEL

[12]:

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear' , random_state = 1)
clf = classifier.fit(X_train , y_train)
```

[13]:

```
y_pred=classifier.predict(x_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.
 ↪reshape(len(y_test),1)),1))
```

```
[[1 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [1 0]
 [1 1]
```

```
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[0 1]
[1 1]
[1 1]
[1 1]
[1 1]
[0 0]
[1 1]
[1 1]
[0 1]
[1 1]
[0 0]
[0 0]
[1 1]
[1 1]
[0 0]
[1 1]
[1 1]
[0 1]
[1 1]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[1 1]
[1 1]
[1 1]
[1 1]
[1 1]
[1 1]
[1 0]
[1 0]
[1 1]
[0 0]
[0 0]
[0 0]
[1 1]
[1 1]
[1 1]
[0 0]
```

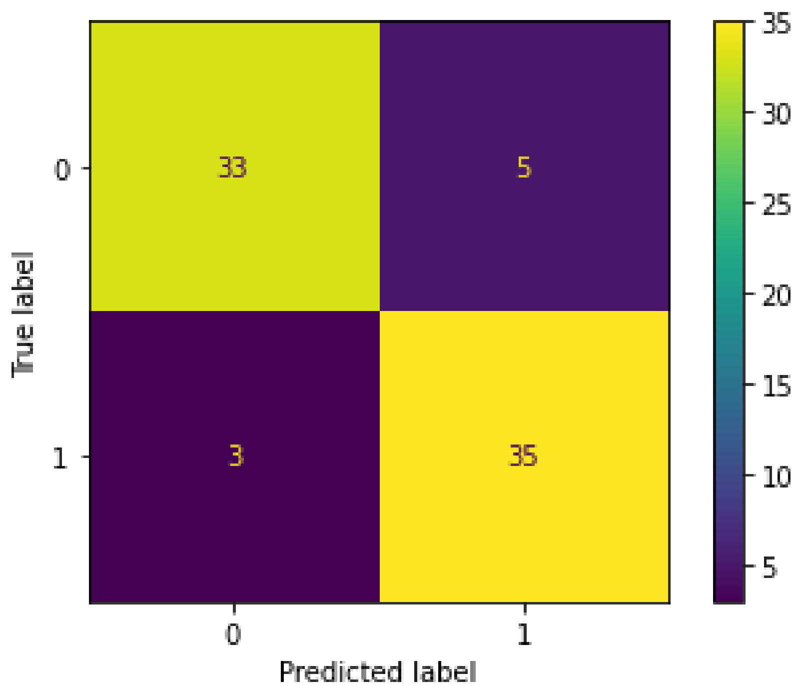
```
[1 1]
[1 1]
[0 0]
[0 0]
[1 0]
[1 1]
[0 0]
[0 0]
[1 1]]
```

[14]:
```python
#confusion matrix is used to check how many datapoints are predicted exactly
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(cm)
print(round(accuracy_score(y_test,y_pred) , 2))
```

```
[[33  5]
 [ 3 35]]
0.89
```

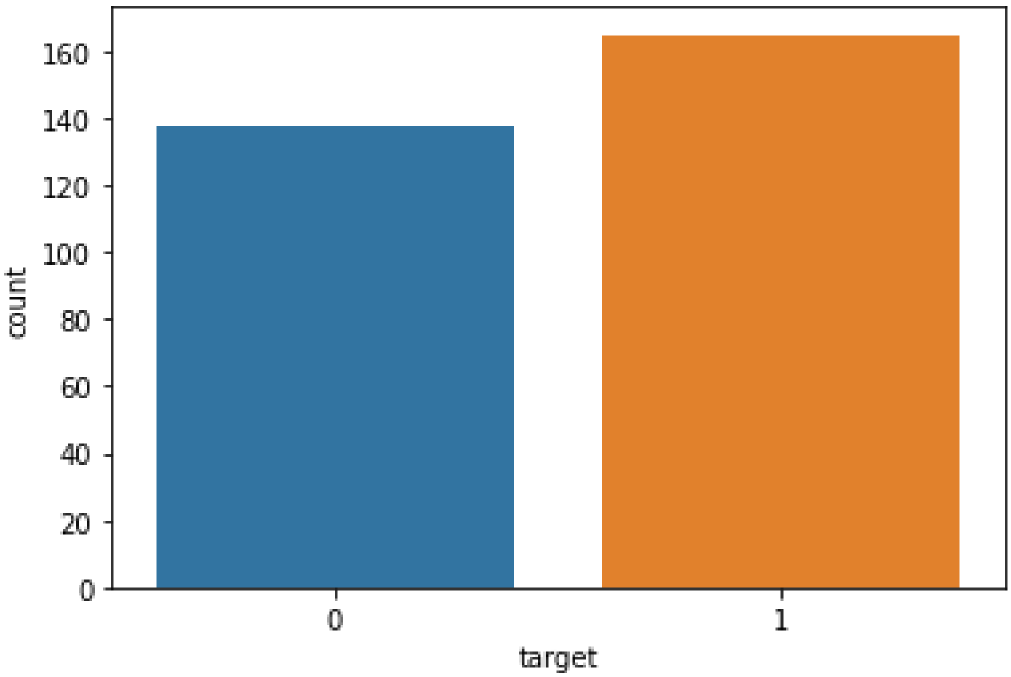
### 0.4 CONFUSION MATRIX

[15]:
```python
from sklearn.metrics import plot_confusion_matrix
a = plot_confusion_matrix(clf , x_test , y_test)
plt.show()
```



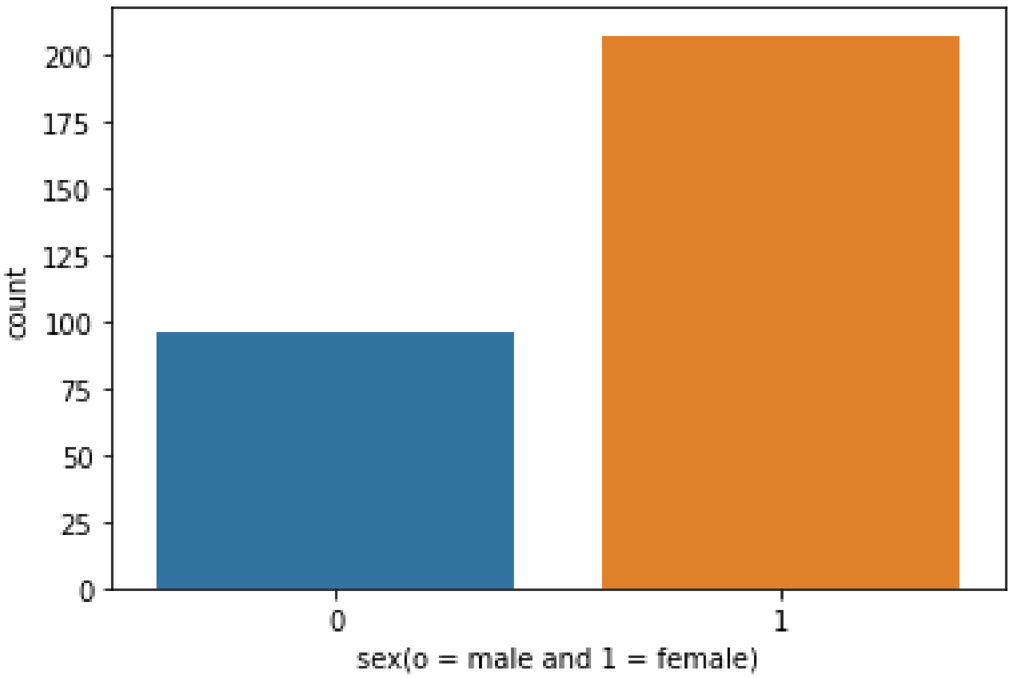
**BAR PLOT FOR COUNT OF PEOPLE DISEASED AND NOT DISEASED**

[16]:
```
import seaborn as sns
sns.countplot(x = 'target' , data = ds)
plt.show()
```



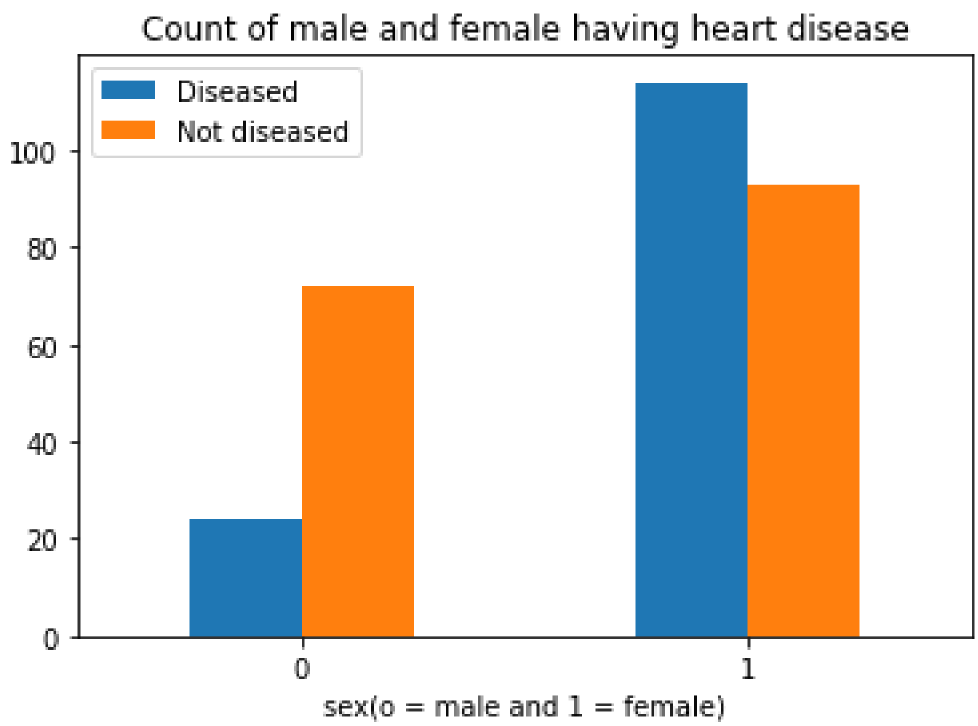

## 0.5 BAR PLOT FOR COUNT OF MALE AND FEMALE

[17]:
```
sns.countplot(x = 'sex' , data = ds)
plt.xlabel("sex(o = male and 1 = female)")
plt.show()
```



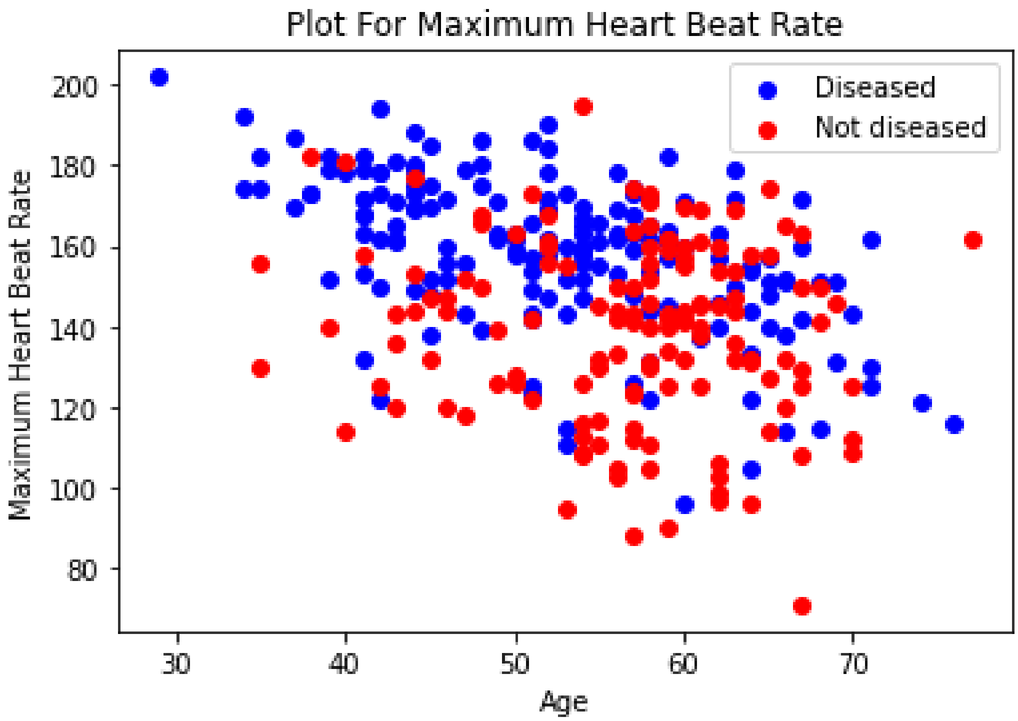

### 0.6 BAR PLOT FOR COUNT OF MALE AND FEMALE HAVING HEART DISEASE

[18]:
```
pd.crosstab(ds.sex , ds.target).plot(kind = 'bar')
plt.title("Count of male and female having heart disease")
plt.xlabel("sex(o = male and 1 = female)")
plt.xticks(rotation=0)
plt.legend(['Diseased', 'Not diseased'])
plt.show()
```




### 0.7 SCATTER PLOT BETWEEN AGE AND MAXIMUM HEART RATE

[19]:
```
plt.scatter(x = ds.age[ds.target==1],y = ds.thalach[ds.target==1] , c= "blue")
plt.scatter(x = ds.age[ds.target==0],y = ds.thalach[ds.target==0] , c= "red")
plt.title('Plot For Maximum Heart Beat Rate')
plt.legend(['Diseased','Not diseased'])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Beat Rate")
plt.show()
```


Plot For Maximum Heart Beat Rate
Diseased
Not diseased
Maximum Heart Beat Rate
200
180
160
140
120
100
80
30
40
50
60
70
Age

[ ]: