

In []:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
```

In []:

```
pwd
```

In []:

```
df = pd.read_csv('../input/heartcsv/Heart.csv')
```

In []:

```
df.head()
```

a) Find Shape of Data ¶

In []:

```
df.shape #303, 15
```

b) Find Missing Values

In []:

```
df.isnull().sum()
```

In []:

```
df.count()
```

c) Find data type of each column

In []:

```
df.info()
```

In []:

```
df.dtypes
```

d) Finding out Zero's

In []:

```
df==0
```

In []:

```
df[df==0]
```

In []:

```
(df == 0).sum()
```

e) Find Mean age of patients

In []:

```
np.mean(df['Age'])
```

In []:

```
df.Age.mean()
```

f) Now extract only Age, Sex, ChestPain, RestBP, Chol. Randomly divide dataset in training (75%) and testing (25%).

In []:

```
df.columns
```

In []:

```
data = df[['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol']]
```

In []:

```
#Cross validation
```

In []:

```
train,test = train_test_split(data,test_size=0.25,random_state=1)
```

In []:

```
train.shape
```

In []:

```
test.shape
```

Through the diagnosis test I predicted 100 report as COVID positive, but only 45 of those were actually positive. Total 50 people in my sample were actually COVID positive. I have total 500 samples.

Create confusion matrix based on above data and find

- I. Accuracy
- II. Precision
- III. Recall
- IV. F-1 score

In []:

```
actual = np.concatenate((np.ones(45), np.zeros(450), np.ones(5)))  
actual
```

In []:

In []:

```
# run = np.array([1,0,1,1,1])
```

In []:

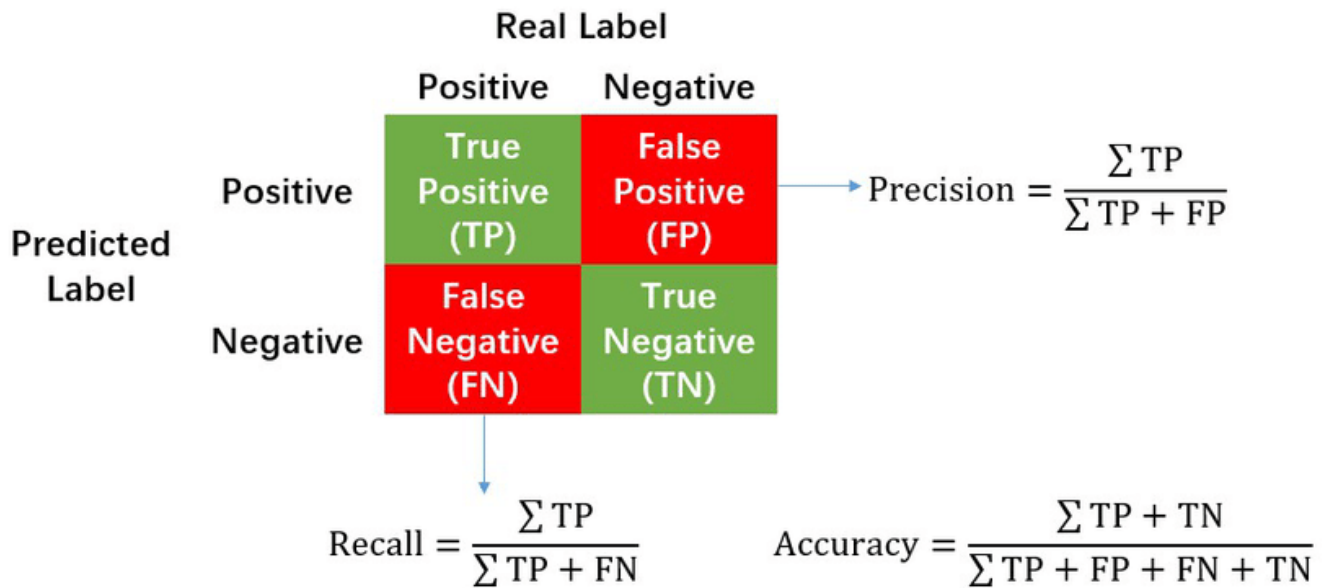
In []:

```
predicted = np.concatenate((np.ones(100), np.zeros(400)))  
predicted
```

In []:

```
type(predicted)
```

Confusion Matrix



In []:

```
from sklearn.metrics import ConfusionMatrixDisplay
```

In []:

```
ConfusionMatrixDisplay.from_predictions(actual,predicted)
```

In []:

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
```

In []:

```
print(classification_report(actual,predicted))
```

In []:

```
accuracy_score(actual,predicted)
```