

```
In [ ]: Name:Aditya Desai  
Roll No: A-19
```

```
In [1]: import pandas as pd  
import numpy as np  
from matplotlib import pyplot as plt  
%matplotlib inline
```

```
In [2]: df = pd.read_csv("/home/kj-comp/Aditya Desai/GCR/DB/Social_Network_Ads(1).csv")  
df.head(10)
```

```
Out[2]:   User ID  Gender  Age  EstimatedSalary  Purchased
```

| | | | | | |
|---|----------|--------|----|--------|---|
| 0 | 15624510 | Male | 19 | 19000 | 0 |
| 1 | 15810944 | Male | 35 | 20000 | 0 |
| 2 | 15668575 | Female | 26 | 43000 | 0 |
| 3 | 15603246 | Female | 27 | 57000 | 0 |
| 4 | 15804002 | Male | 19 | 76000 | 0 |
| 5 | 15728773 | Male | 27 | 58000 | 0 |
| 6 | 15598044 | Female | 27 | 84000 | 0 |
| 7 | 15694829 | Female | 32 | 150000 | 1 |
| 8 | 15600575 | Male | 25 | 33000 | 0 |
| 9 | 15727311 | Female | 35 | 65000 | 0 |

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 400 entries, 0 to 399  
Data columns (total 5 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   User ID          400 non-null    int64    
 1   Gender           400 non-null    object    
 2   Age              400 non-null    int64    
 3   EstimatedSalary  400 non-null    int64    
 4   Purchased        400 non-null    int64  
dtypes: int64(4), object(1)  
memory usage: 15.8+ KB
```

```
In [4]: df.describe()
```

Out[4]:

| | User ID | Age | EstimatedSalary | Purchased |
|--------------|--------------|------------|-----------------|------------|
| count | 4.000000e+02 | 400.000000 | 400.000000 | 400.000000 |
| mean | 1.569154e+07 | 37.655000 | 69742.500000 | 0.357500 |
| std | 7.165832e+04 | 10.482877 | 34096.960282 | 0.479864 |
| min | 1.556669e+07 | 18.000000 | 15000.000000 | 0.000000 |
| 25% | 1.562676e+07 | 29.750000 | 43000.000000 | 0.000000 |
| 50% | 1.569434e+07 | 37.000000 | 70000.000000 | 0.000000 |
| 75% | 1.575036e+07 | 46.000000 | 88000.000000 | 1.000000 |
| max | 1.581524e+07 | 60.000000 | 150000.000000 | 1.000000 |

In [5]:

```
X = df.iloc[:,[2,3]].values  
y = df.iloc[:,4].values
```

In [6]:

```
X
```

```
Out[6]: array([[ 19, 19000],  
   [ 35, 20000],  
   [ 26, 43000],  
   [ 27, 57000],  
   [ 19, 76000],  
   [ 27, 58000],  
   [ 27, 84000],  
   [ 32, 150000],  
   [ 25, 33000],  
   [ 35, 65000],  
   [ 26, 80000],  
   [ 26, 52000],  
   [ 20, 86000],  
   [ 32, 18000],  
   [ 18, 82000],  
   [ 29, 80000],  
   [ 47, 25000],  
   [ 45, 26000],  
   [ 46, 28000],  
   [ 48, 29000],  
   [ 45, 22000],  
   [ 47, 49000],  
   [ 48, 41000],  
   [ 45, 22000],  
   [ 46, 23000],  
   [ 47, 20000],  
   [ 49, 28000],  
   [ 47, 30000],  
   [ 29, 43000],  
   [ 31, 18000],  
   [ 31, 74000],  
   [ 27, 137000],  
   [ 21, 16000],  
   [ 28, 44000],  
   [ 27, 90000],  
   [ 35, 27000],  
   [ 33, 28000],  
   [ 30, 49000],  
   [ 26, 72000],  
   [ 27, 31000],  
   [ 27, 17000],  
   [ 33, 51000],  
   [ 35, 108000],  
   [ 30, 15000],  
   [ 28, 84000],  
   [ 23, 20000],  
   [ 25, 79000],  
   [ 27, 54000],  
   [ 30, 135000],  
   [ 31, 89000],  
   [ 24, 32000],  
   [ 18, 44000],  
   [ 29, 83000],  
   [ 35, 23000],  
   [ 27, 58000],  
   [ 24, 55000],
```

```
[ 23, 48000],  
[ 28, 79000],  
[ 22, 18000],  
[ 32, 117000],  
[ 27, 20000],  
[ 25, 87000],  
[ 23, 66000],  
[ 32, 120000],  
[ 59, 83000],  
[ 24, 58000],  
[ 24, 19000],  
[ 23, 82000],  
[ 22, 63000],  
[ 31, 68000],  
[ 25, 80000],  
[ 24, 27000],  
[ 20, 23000],  
[ 33, 113000],  
[ 32, 18000],  
[ 34, 112000],  
[ 18, 52000],  
[ 22, 27000],  
[ 28, 87000],  
[ 26, 17000],  
[ 30, 80000],  
[ 39, 42000],  
[ 20, 49000],  
[ 35, 88000],  
[ 30, 62000],  
[ 31, 118000],  
[ 24, 55000],  
[ 28, 85000],  
[ 26, 81000],  
[ 35, 50000],  
[ 22, 81000],  
[ 30, 116000],  
[ 26, 15000],  
[ 29, 28000],  
[ 29, 83000],  
[ 35, 44000],  
[ 35, 25000],  
[ 28, 123000],  
[ 35, 73000],  
[ 28, 37000],  
[ 27, 88000],  
[ 28, 59000],  
[ 32, 86000],  
[ 33, 149000],  
[ 19, 21000],  
[ 21, 72000],  
[ 26, 35000],  
[ 27, 89000],  
[ 26, 86000],  
[ 38, 80000],  
[ 39, 71000],  
[ 37, 71000],
```

```
[ 38, 61000],  
[ 37, 55000],  
[ 42, 80000],  
[ 40, 57000],  
[ 35, 75000],  
[ 36, 52000],  
[ 40, 59000],  
[ 41, 59000],  
[ 36, 75000],  
[ 37, 72000],  
[ 40, 75000],  
[ 35, 53000],  
[ 41, 51000],  
[ 39, 61000],  
[ 42, 65000],  
[ 26, 32000],  
[ 30, 17000],  
[ 26, 84000],  
[ 31, 58000],  
[ 33, 31000],  
[ 30, 87000],  
[ 21, 68000],  
[ 28, 55000],  
[ 23, 63000],  
[ 20, 82000],  
[ 30, 107000],  
[ 28, 59000],  
[ 19, 25000],  
[ 19, 85000],  
[ 18, 68000],  
[ 35, 59000],  
[ 30, 89000],  
[ 34, 25000],  
[ 24, 89000],  
[ 27, 96000],  
[ 41, 30000],  
[ 29, 61000],  
[ 20, 74000],  
[ 26, 15000],  
[ 41, 45000],  
[ 31, 76000],  
[ 36, 50000],  
[ 40, 47000],  
[ 31, 15000],  
[ 46, 59000],  
[ 29, 75000],  
[ 26, 30000],  
[ 32, 135000],  
[ 32, 100000],  
[ 25, 90000],  
[ 37, 33000],  
[ 35, 38000],  
[ 33, 69000],  
[ 18, 86000],  
[ 22, 55000],  
[ 35, 71000],
```

```
[ 29, 148000],  
[ 29, 47000],  
[ 21, 88000],  
[ 34, 115000],  
[ 26, 118000],  
[ 34, 43000],  
[ 34, 72000],  
[ 23, 28000],  
[ 35, 47000],  
[ 25, 22000],  
[ 24, 23000],  
[ 31, 34000],  
[ 26, 16000],  
[ 31, 71000],  
[ 32, 117000],  
[ 33, 43000],  
[ 33, 60000],  
[ 31, 66000],  
[ 20, 82000],  
[ 33, 41000],  
[ 35, 72000],  
[ 28, 32000],  
[ 24, 84000],  
[ 19, 26000],  
[ 29, 43000],  
[ 19, 70000],  
[ 28, 89000],  
[ 34, 43000],  
[ 30, 79000],  
[ 20, 36000],  
[ 26, 80000],  
[ 35, 22000],  
[ 35, 39000],  
[ 49, 74000],  
[ 39, 134000],  
[ 41, 71000],  
[ 58, 101000],  
[ 47, 47000],  
[ 55, 130000],  
[ 52, 114000],  
[ 40, 142000],  
[ 46, 22000],  
[ 48, 96000],  
[ 52, 150000],  
[ 59, 42000],  
[ 35, 58000],  
[ 47, 43000],  
[ 60, 108000],  
[ 49, 65000],  
[ 40, 78000],  
[ 46, 96000],  
[ 59, 143000],  
[ 41, 80000],  
[ 35, 91000],  
[ 37, 144000],  
[ 60, 102000],
```

```
[ 35, 60000],  
[ 37, 53000],  
[ 36, 126000],  
[ 56, 133000],  
[ 40, 72000],  
[ 42, 80000],  
[ 35, 147000],  
[ 39, 42000],  
[ 40, 107000],  
[ 49, 86000],  
[ 38, 112000],  
[ 46, 79000],  
[ 40, 57000],  
[ 37, 80000],  
[ 46, 82000],  
[ 53, 143000],  
[ 42, 149000],  
[ 38, 59000],  
[ 50, 88000],  
[ 56, 104000],  
[ 41, 72000],  
[ 51, 146000],  
[ 35, 50000],  
[ 57, 122000],  
[ 41, 52000],  
[ 35, 97000],  
[ 44, 39000],  
[ 37, 52000],  
[ 48, 134000],  
[ 37, 146000],  
[ 50, 44000],  
[ 52, 90000],  
[ 41, 72000],  
[ 40, 57000],  
[ 58, 95000],  
[ 45, 131000],  
[ 35, 77000],  
[ 36, 144000],  
[ 55, 125000],  
[ 35, 72000],  
[ 48, 90000],  
[ 42, 108000],  
[ 40, 75000],  
[ 37, 74000],  
[ 47, 144000],  
[ 40, 61000],  
[ 43, 133000],  
[ 59, 76000],  
[ 60, 42000],  
[ 39, 106000],  
[ 57, 26000],  
[ 57, 74000],  
[ 38, 71000],  
[ 49, 88000],  
[ 52, 38000],  
[ 50, 36000],
```

```
[ 59, 88000],  
[ 35, 61000],  
[ 37, 70000],  
[ 52, 21000],  
[ 48, 141000],  
[ 37, 93000],  
[ 37, 62000],  
[ 48, 138000],  
[ 41, 79000],  
[ 37, 78000],  
[ 39, 134000],  
[ 49, 89000],  
[ 55, 39000],  
[ 37, 77000],  
[ 35, 57000],  
[ 36, 63000],  
[ 42, 73000],  
[ 43, 112000],  
[ 45, 79000],  
[ 46, 117000],  
[ 58, 38000],  
[ 48, 74000],  
[ 37, 137000],  
[ 37, 79000],  
[ 40, 60000],  
[ 42, 54000],  
[ 51, 134000],  
[ 47, 113000],  
[ 36, 125000],  
[ 38, 50000],  
[ 42, 70000],  
[ 39, 96000],  
[ 38, 50000],  
[ 49, 141000],  
[ 39, 79000],  
[ 39, 75000],  
[ 54, 104000],  
[ 35, 55000],  
[ 45, 32000],  
[ 36, 60000],  
[ 52, 138000],  
[ 53, 82000],  
[ 41, 52000],  
[ 48, 30000],  
[ 48, 131000],  
[ 41, 60000],  
[ 41, 72000],  
[ 42, 75000],  
[ 36, 118000],  
[ 47, 107000],  
[ 38, 51000],  
[ 48, 119000],  
[ 42, 65000],  
[ 40, 65000],  
[ 57, 60000],  
[ 36, 54000],
```

```
[ 58, 144000],  
[ 35, 79000],  
[ 38, 55000],  
[ 39, 122000],  
[ 53, 104000],  
[ 35, 75000],  
[ 38, 65000],  
[ 47, 51000],  
[ 47, 105000],  
[ 41, 63000],  
[ 53, 72000],  
[ 54, 108000],  
[ 39, 77000],  
[ 38, 61000],  
[ 38, 113000],  
[ 37, 75000],  
[ 42, 90000],  
[ 37, 57000],  
[ 36, 99000],  
[ 60, 34000],  
[ 54, 70000],  
[ 41, 72000],  
[ 40, 71000],  
[ 42, 54000],  
[ 43, 129000],  
[ 53, 34000],  
[ 47, 50000],  
[ 42, 79000],  
[ 42, 104000],  
[ 59, 29000],  
[ 58, 47000],  
[ 46, 88000],  
[ 38, 71000],  
[ 54, 26000],  
[ 60, 46000],  
[ 60, 83000],  
[ 39, 73000],  
[ 59, 130000],  
[ 37, 80000],  
[ 46, 32000],  
[ 46, 74000],  
[ 42, 53000],  
[ 41, 87000],  
[ 58, 23000],  
[ 42, 64000],  
[ 48, 33000],  
[ 44, 139000],  
[ 49, 28000],  
[ 57, 33000],  
[ 56, 60000],  
[ 49, 39000],  
[ 39, 71000],  
[ 47, 34000],  
[ 48, 35000],  
[ 48, 33000],  
[ 47, 23000],
```

```
[ 45, 45000],  
[ 60, 42000],  
[ 39, 59000],  
[ 46, 41000],  
[ 51, 23000],  
[ 50, 20000],  
[ 36, 33000],  
[ 49, 36000]]))
```

In [7]: y

```
In [10]: 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_
```

```
In [11]: from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
X_train = sc.fit_transform(X_train)  
X_test = sc.transform(X_test)
```

In [12]: x train

```
Out[12]: array([[ 0.58164944, -0.88670699],  
[-0.60673761,  1.46173768],  
[-0.01254409, -0.5677824 ],  
[-0.60673761,  1.89663484],  
[ 1.37390747, -1.40858358],  
[ 1.47293972,  0.99784738],  
[ 0.08648817, -0.79972756],  
[-0.01254409, -0.24885782],  
[-0.21060859, -0.5677824 ],  
[-0.21060859, -0.19087153],  
[-0.30964085, -1.29261101],  
[-0.30964085, -0.5677824 ],  
[ 0.38358493,  0.09905991],  
[ 0.8787462 , -0.59677555],  
[ 2.06713324, -1.17663843],  
[ 1.07681071, -0.13288524],  
[ 0.68068169,  1.78066227],  
[-0.70576986,  0.56295021],  
[ 0.77971394,  0.35999821],  
[ 0.8787462 , -0.53878926],  
[-1.20093113, -1.58254245],  
[ 2.1661655 ,  0.93986109],  
[-0.01254409,  1.22979253],  
[ 0.18552042,  1.08482681],  
[ 0.38358493, -0.48080297],  
[-0.30964085, -0.30684411],  
[ 0.97777845, -0.8287207 ],  
[ 0.97777845,  1.8676417 ],  
[-0.01254409,  1.25878567],  
[-0.90383437,  2.27354572],  
[-1.20093113, -1.58254245],  
[ 2.1661655 , -0.79972756],  
[-1.39899564, -1.46656987],  
[ 0.38358493,  2.30253886],  
[ 0.77971394,  0.76590222],  
[-1.00286662, -0.30684411],  
[ 0.08648817,  0.76590222],  
[-1.00286662,  0.56295021],  
[ 0.28455268,  0.07006676],  
[ 0.68068169, -1.26361786],  
[-0.50770535, -0.01691267],  
[-1.79512465,  0.35999821],  
[-0.70576986,  0.12805305],  
[ 0.38358493,  0.30201192],  
[-0.30964085,  0.07006676],  
[-0.50770535,  2.30253886],  
[ 0.18552042,  0.04107362],  
[ 1.27487521,  2.21555943],  
[ 0.77971394,  0.27301877],  
[-0.30964085,  0.1570462 ],  
[-0.01254409, -0.53878926],  
[-0.21060859,  0.1570462 ],  
[-0.11157634,  0.24402563],  
[-0.01254409, -0.24885782],  
[ 2.1661655 ,  1.11381995],  
[-1.79512465,  0.35999821],
```

```
[ 1.86906873,  0.12805305],  
[ 0.38358493, -0.13288524],  
[-1.20093113,  0.30201192],  
[ 0.77971394,  1.37475825],  
[-0.30964085, -0.24885782],  
[-1.6960924 , -0.04590581],  
[-1.00286662, -0.74174127],  
[ 0.28455268,  0.50496393],  
[-0.11157634, -1.06066585],  
[-1.10189888,  0.59194336],  
[ 0.08648817, -0.79972756],  
[-1.00286662,  1.54871711],  
[-0.70576986,  1.40375139],  
[-1.29996338,  0.50496393],  
[-0.30964085,  0.04107362],  
[-0.11157634,  0.01208048],  
[-0.30964085, -0.88670699],  
[ 0.8787462 , -1.3505973 ],  
[-0.30964085,  2.24455257],  
[ 0.97777845,  1.98361427],  
[-1.20093113,  0.47597078],  
[-1.29996338,  0.27301877],  
[ 1.37390747,  1.98361427],  
[ 1.27487521, -1.3505973 ],  
[-0.30964085, -0.27785096],  
[-0.50770535,  1.25878567],  
[-0.80480212,  1.08482681],  
[ 0.97777845, -1.06066585],  
[ 0.28455268,  0.30201192],  
[ 0.97777845,  0.76590222],  
[-0.70576986, -1.49556302],  
[-0.70576986,  0.04107362],  
[ 0.48261718,  1.72267598],  
[ 2.06713324,  0.18603934],  
[-1.99318916, -0.74174127],  
[-0.21060859,  1.40375139],  
[ 0.38358493,  0.59194336],  
[ 0.8787462 , -1.14764529],  
[-1.20093113, -0.77073441],  
[ 0.18552042,  0.24402563],  
[ 0.77971394, -0.30684411],  
[ 2.06713324, -0.79972756],  
[ 0.77971394,  0.12805305],  
[-0.30964085,  0.6209365 ],  
[-1.00286662, -0.30684411],  
[ 0.18552042, -0.3648304 ],  
[ 2.06713324,  2.12857999],  
[ 1.86906873, -1.26361786],  
[ 1.37390747, -0.91570013],  
[ 0.8787462 ,  1.25878567],  
[ 1.47293972,  2.12857999],  
[-0.30964085, -1.23462472],  
[ 1.96810099,  0.91086794],  
[ 0.68068169, -0.71274813],  
[-1.49802789,  0.35999821],  
[ 0.77971394, -1.3505973 ],
```

```
[ 0.38358493, -0.13288524],  
[-1.00286662,  0.41798449],  
[-0.01254409, -0.30684411],  
[-1.20093113,  0.41798449],  
[-0.90383437, -1.20563157],  
[-0.11157634,  0.04107362],  
[-1.59706014, -0.42281668],  
[ 0.97777845, -1.00267957],  
[ 1.07681071, -1.20563157],  
[-0.01254409, -0.13288524],  
[-1.10189888, -1.52455616],  
[ 0.77971394, -1.20563157],  
[ 0.97777845,  2.07059371],  
[-1.20093113, -1.52455616],  
[-0.30964085,  0.79489537],  
[ 0.08648817, -0.30684411],  
[-1.39899564, -1.23462472],  
[-0.60673761, -1.49556302],  
[ 0.77971394,  0.53395707],  
[-0.30964085, -0.33583725],  
[ 1.77003648, -0.27785096],  
[ 0.8787462 , -1.03167271],  
[ 0.18552042,  0.07006676],  
[-0.60673761,  0.8818748 ],  
[-1.89415691, -1.40858358],  
[-1.29996338,  0.59194336],  
[-0.30964085,  0.53395707],  
[-1.00286662, -1.089659 ],  
[ 1.17584296, -1.43757673],  
[ 0.18552042, -0.30684411],  
[ 1.17584296, -0.74174127],  
[-0.30964085,  0.07006676],  
[ 0.18552042,  2.09958685],  
[ 0.77971394, -1.089659 ],  
[ 0.08648817,  0.04107362],  
[-1.79512465,  0.12805305],  
[-0.90383437,  0.1570462 ],  
[-0.70576986,  0.18603934],  
[ 0.8787462 , -1.29261101],  
[ 0.18552042, -0.24885782],  
[-0.4086731 ,  1.22979253],  
[-0.01254409,  0.30201192],  
[ 0.38358493,  0.1570462 ],  
[ 0.8787462 , -0.65476184],  
[ 0.08648817,  0.1570462 ],  
[-1.89415691, -1.29261101],  
[-0.11157634,  0.30201192],  
[-0.21060859, -0.27785096],  
[ 0.28455268, -0.50979612],  
[-0.21060859,  1.6067034 ],  
[ 0.97777845, -1.17663843],  
[-0.21060859,  1.63569655],  
[ 1.27487521,  1.8676417 ],  
[-1.10189888, -0.3648304 ],  
[-0.01254409,  0.04107362],  
[ 0.08648817, -0.24885782],
```

```
[-1.59706014, -1.23462472],  
[-0.50770535, -0.27785096],  
[ 0.97777845,  0.12805305],  
[ 1.96810099, -1.3505973 ],  
[ 1.47293972,  0.07006676],  
[-0.60673761,  1.37475825],  
[ 1.57197197,  0.01208048],  
[-0.80480212,  0.30201192],  
[ 1.96810099,  0.73690908],  
[-1.20093113, -0.50979612],  
[ 0.68068169,  0.27301877],  
[-1.39899564, -0.42281668],  
[ 0.18552042,  0.1570462 ],  
[-0.50770535, -1.20563157],  
[ 0.58164944,  2.01260742],  
[-1.59706014, -1.49556302],  
[-0.50770535, -0.53878926],  
[ 0.48261718,  1.83864855],  
[-1.39899564, -1.089659 ],  
[ 0.77971394, -1.37959044],  
[-0.30964085, -0.42281668],  
[ 1.57197197,  0.99784738],  
[ 0.97777845,  1.43274454],  
[-0.30964085, -0.48080297],  
[-0.11157634,  2.15757314],  
[-1.49802789, -0.1038921 ],  
[-0.11157634,  1.95462113],  
[-0.70576986, -0.33583725],  
[-0.50770535, -0.8287207 ],  
[ 0.68068169, -1.37959044],  
[-0.80480212, -1.58254245],  
[-1.89415691, -1.46656987],  
[ 1.07681071,  0.12805305],  
[ 0.08648817,  1.51972397],  
[-0.30964085,  0.09905991],  
[ 0.08648817,  0.04107362],  
[-1.39899564, -1.3505973 ],  
[ 0.28455268,  0.07006676],  
[-0.90383437,  0.38899135],  
[ 1.57197197, -1.26361786],  
[-0.30964085, -0.74174127],  
[-0.11157634,  0.1570462 ],  
[-0.90383437, -0.65476184],  
[-0.70576986, -0.04590581],  
[ 0.38358493, -0.45180983],  
[-0.80480212,  1.89663484],  
[ 1.37390747,  1.28777882],  
[ 1.17584296, -0.97368642],  
[ 1.77003648,  1.83864855],  
[-0.90383437, -0.24885782],  
[-0.80480212,  0.56295021],  
[-1.20093113, -1.5535493 ],  
[-0.50770535, -1.11865214],  
[ 0.28455268,  0.07006676],  
[-0.21060859, -1.06066585],  
[ 1.67100423,  1.6067034 ],
```

```
[ 0.97777845,  1.78066227],  
[ 0.28455268,  0.04107362],  
[-0.80480212, -0.21986468],  
[-0.11157634,  0.07006676],  
[ 0.28455268, -0.19087153],  
[ 1.96810099, -0.65476184],  
[-0.80480212,  1.3457651 ],  
[-1.79512465, -0.59677555],  
[-0.11157634,  0.12805305],  
[ 0.28455268, -0.30684411],  
[ 1.07681071,  0.56295021],  
[-1.00286662,  0.27301877],  
[ 1.47293972,  0.35999821],  
[ 0.18552042, -0.3648304 ],  
[ 2.1661655 , -1.03167271],  
[-0.30964085,  1.11381995],  
[-1.6960924 ,  0.07006676],  
[-0.01254409,  0.04107362],  
[ 0.08648817,  1.05583366],  
[-0.11157634, -0.3648304 ],  
[-1.20093113,  0.07006676],  
[-0.30964085, -1.3505973 ],  
[ 1.57197197,  1.11381995],  
[-0.80480212, -1.52455616],  
[ 0.08648817,  1.8676417 ],  
[-0.90383437, -0.77073441],  
[-0.50770535, -0.77073441],  
[-0.30964085, -0.91570013],  
[ 0.28455268, -0.71274813],  
[ 0.28455268,  0.07006676],  
[ 0.08648817,  1.8676417 ],  
[-1.10189888,  1.95462113],  
[-1.6960924 , -1.5535493 ],  
[-1.20093113, -1.089659 ],  
[-0.70576986, -0.1038921 ],  
[ 0.08648817,  0.09905991],  
[ 0.28455268,  0.27301877],  
[ 0.8787462 , -0.5677824 ],  
[ 0.28455268, -1.14764529],  
[-0.11157634,  0.67892279],  
[ 2.1661655 , -0.68375498],  
[-1.29996338, -1.37959044],  
[-1.00286662, -0.94469328],  
[-0.01254409, -0.42281668],  
[-0.21060859, -0.45180983],  
[-1.79512465, -0.97368642],  
[ 1.77003648,  0.99784738],  
[ 0.18552042, -0.3648304 ],  
[ 0.38358493,  1.11381995],  
[-1.79512465, -1.3505973 ],  
[ 0.18552042, -0.13288524],  
[ 0.8787462 , -1.43757673],  
[-1.99318916,  0.47597078],  
[-0.30964085,  0.27301877],  
[ 1.86906873, -1.06066585],  
[-0.4086731 ,  0.07006676],
```

```
[ 1.07681071, -0.88670699],
[-1.10189888, -1.11865214],
[-1.89415691,  0.01208048],
[ 0.08648817,  0.27301877],
[-1.20093113,  0.33100506],
[-1.29996338,  0.30201192],
[-1.00286662,  0.44697764],
[ 1.67100423, -0.88670699],
[ 1.17584296,  0.53395707],
[ 1.07681071,  0.53395707],
[ 1.37390747,  2.331532 ],
[-0.30964085, -0.13288524],
[ 0.38358493, -0.45180983],
[-0.4086731 , -0.77073441],
[-0.11157634, -0.50979612],
[ 0.97777845, -1.14764529],
[-0.90383437, -0.77073441],
[-0.21060859, -0.50979612],
[-1.10189888, -0.45180983],
[-1.20093113,  1.40375139]])
```

In [13]: `from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train,y_train)`

Out[13]: `LogisticRegression(random_state=0)`

In [14]: `y_pred = classifier.predict(X_test)`

In [15]: `y_pred`

Out[15]: `array([0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0,
0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1])`

In [16]: `from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test , y_pred)`

In [17]: `cm`

Out[17]: `array([[65, 3],
[8, 24]])`

In [18]: `c1_report = classification_report(y_test,y_pred)`

In [19]: `c1_report`

Out[19]: `'precision recall f1-score support\n\n 0 0.96 0.92 0.92 68\n 1 0.89 0.75 0.82 32\n accuracy 0.89 100\nmacro avg 0.89 0.89 0.89 100\nweighted avg 0.85 0.87 0.87 100'`

```
In [20]: tp , fn ,fp , tn = confusion_matrix(y_test,y_pred,labels=[0,1]).reshape(-1)
print('Outcome values : \n' , tp , fn , fp ,tn)
```

Outcome values :
65 3 8 24

```
In [21]: accuracy_cm = (tp+tn)/(tp+fp+tn+fn)
precision_cm = tp/(tp+fp)
recall_cm = tp/(tp+fn)
f1_score = 2/((1/recall_cm)+(1/precision_cm))
```

```
In [22]: print("Accuracy :",accuracy_cm)
print("Precision :",precision_cm)
print("Recall :",recall_cm)
print("F1-Score :",f1_score)
```

Accuracy : 0.89
Precision : 0.8904109589041096
Recall : 0.9558823529411765
F1-Score : 0.9219858156028368

```
In [ ]:
```