

In [28]:

```
import pandas, matplotlib.pyplot as plot, numpy
```

In [29]:

```
data_frame = pandas.read_csv('DataSets/social_network.csv')
```

In [30]:

```
data_frame
```

Out[30]:

	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
...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [31]:

```
x = data_frame.iloc[:, [2, 3]].values
```

In [32]:

```
y = data_frame.iloc[:, 4].values
```

In [33]:

```
from sklearn.model_selection import train_test_split
```

In [34]:

```
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.25, random_state=0)
```

In [35]:

```
from sklearn.preprocessing import StandardScaler
```

In [36]:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [37]:

```
from sklearn.linear_model import LogisticRegression
```

In [38]:

```
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, Y_train)
```

Out[38]:

▼	LogisticRegression
LogisticRegression(random_state=0)	

In [39]:

```
Y_predict = classifier.predict(X_test)
```

In [40]:

```
from sklearn.metrics import confusion_matrix, classification_report
```

In [41]:

```
confusionMatrix = confusion_matrix(Y_test, Y_predict)
```

In [42]:

```
confusionMatrix
```

Out[42]:

```
array([[65,  3],
       [ 8, 24]], dtype=int64)
```

In [43]:

```
print(confusionMatrix)
```

```
[[65  3]
 [ 8 24]]
```

In [44]:

```
TP, FP, FN, TN = confusionMatrix[0][0], confusionMatrix[0][1], confusionMatrix[1][0], confusionMatrix[1][1]
```

In [45]:

```
Total = TP + FP + FN + TN
```

In [46]:

```
Accuracy = (TN + TP) / Total
```

In [47]:

```
Accuracy
```

Out[47]:

```
0.89
```

In [48]:

```
Error_rate = (FN + FP)/Total
```

In [49]:

```
Error_rate
```

Out[49]:

0.11

In [50]:

```
report = classification_report(Y_test, Y_predict)
```

In [51]:

```
report
```

Out[51]:

		precision	recall	f1-score	support\n\n	0	0.89	0.96
0.92	68\n		1	0.89	0.75	0.81	32\n\n	accuracy
0.89	100\n	macro avg		0.89	0.85	0.87	100\n	
89	0.89	0.89	100\n'				weighted avg	

In []: