

```
In [2]: import pandas as pd
```

```
In [3]: import numpy as np
```

```
In [4]: df=pd.read_csv("Social_Network_Ads.csv")
```

```
In [5]: df=pd.DataFrame(df)
```

```
In [6]: df
```

```
Out[6]:
```

	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 [7]: from sklearn.model_selection import train_test_split
```

```
In [8]: X = df[['Age','EstimatedSalary']]
```

```
In [9]: Y = df[['Purchased']]
```

```
In [11]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
```

```
In [12]: print(X_train.shape)
```

(300, 2)

```
In [13]: print(Y_train.shape)
```

(300, 1)

```
In [14]: print(X_test.shape)
```

(100, 2)

```
In [15]: print(Y_test.shape)
(100, 1)
```

```
In [16]: from sklearn.preprocessing import StandardScaler
```

```
In [17]: sc = StandardScaler()
```

```
In [18]: X_train=sc.fit_transform(X_train)
```

```
In [19]: X_test=sc.transform(X_test)
```

```
In [20]: from sklearn.linear_model import LogisticRegression
```

```
In [21]: Reg = LogisticRegression()
```

```
In [23]: Reg.fit(X_train, Y_train)
```

```
/home/student/.local/lib/python3.8/site-packages/sklearn/utils/validation.py:1183: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
Out[23]: ▼ LogisticRegression
LogisticRegression()
```

```
In [24]: Y_pred = Reg.predict(X_test)
```

```
In [25]: print(Y_pred)
[0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0
 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0
 0 1 1 1
 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
```

```
In [26]: from sklearn.metrics import confusion_matrix,classification_report
```

```
In [27]: cm = confusion_matrix(Y_test, Y_pred)
```

```
In [28]: cl_report=classification_report(Y_test,Y_pred)
```

```
In [29]: print(cm)
```

```
[[70  5]
 [ 7 18]]
```

In [30]: `print(cl_report)`

	precision	recall	f1-score	support
0	0.91	0.93	0.92	75
1	0.78	0.72	0.75	25
accuracy			0.88	100
macro avg	0.85	0.83	0.84	100
weighted avg	0.88	0.88	0.88	100

In [31]: `from sklearn.metrics import precision_score, recall_score, accuracy_score`

In [32]: `PS=precision_score(Y_test,Y_pred)`

In [33]: `print(PS)`

0.782608695652174

In [34]: `AS=accuracy_score(Y_test,Y_pred)`

In [36]: `print(AS)`

0.88

In [37]: `RS=recall_score(Y_test,Y_pred)`

In [38]: `print(RS)`

0.72

In [39]: `error_rate=1-AS`

In [40]: `print(error_rate)`

0.12

In []: