

In [1]:

```
import pandas as pd;
import numpy as ny;
```

In [2]:

```
dataset = pd.read_csv('Social_Network_Ads.csv')
```

In [3]:

```
dataset.head()
```

Out[3]:

	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

In [4]:

```
dataset['Gender'].replace('Male',1,inplace=True);
dataset['Gender'].replace('Female',0,inplace=True);
```

In [5]:

```
dataset['Gender']
```

Out[5]:

```
0      1
1      1
2      0
3      0
4      1
..
395    0
396    1
397    0
398    1
399    0
Name: Gender, Length: 400, dtype: int64
```

In [6]:

```
X= dataset.iloc[:,1:4]
Y= dataset.iloc[:, -1]
X
```

Out[6]:

	Gender	Age	EstimatedSalary
0	1	19	19000
1	1	35	20000
2	0	26	43000
3	0	27	57000
4	1	19	76000
...	...	...	...
395	0	46	41000
396	1	51	23000
397	0	50	20000
398	1	36	33000
399	0	49	36000

400 rows × 3 columns

In [7]:

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

In [8]:

```
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.20,random_state
```

In [9]:

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

In [10]:

```
logistic_model = LogisticRegression()
```

In [11]:

```
logistic_model.fit(x_train,y_train)
```

Out[11]:

```
LogisticRegression()
```

In [12]:

```
y_pred = logistic_model.predict(x_test)
```

In [13]:

```
y_pred
```

Out[13]:

```
array([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])
```

In [14]:

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

In [18]:

```
accuracy = accuracy_score(y_test,y_pred)*100
confusion_mat = confusion_matrix(y_test,y_pred)
```

In [20]:

```
print(accuracy)
print(confusion_mat)
```

```
56.25
[[45  0]
 [35  0]]
```

In [23]:

```
TP = confusion_mat[0][0];
TN =confusion_mat[1][1];
FP = confusion_mat[1][0];
FN = confusion_mat[0][1];

Precision = (TP)/(TP+FP);

Recall = (TP)/(TP+FN)

Error = (FP+FN)/(TP+TN+FN+FP)
```

In [25]:

```
print("TP : ",TP)
print("FP : ",FP)
print("TN : ",TN)
print("FN : ",FN)
print("Precision : ", Precision)
print("Recall : ", Recall)
print("Error : ", Error)
```

```
TP : 45
FP : 35
TN : 0
FN : 0
Precision : 0.5625
Recall : 1.0
Error : 0.4375
```

In [ ]: