Logistic Regression

Libraries

In [32]:

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
```

Import dataset

In [33]:

```
dataset=pd.read_csv("F:/Data set/Social_Network_Ads.csv")
```

In [34]:

```
dataset.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 [35]:

```
x=dataset.iloc[:,[2,3]].values
y=dataset.iloc[:,-1].values
```

Feature Scaling

In [36]:

```
sc=StandardScaler()
x=sc.fit_transform(x)
```

Split the Train set and Test set

In [37]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,random_state=0)
```

Training the Logistic Regression model on train set

In [38]:

```
classifier=LogisticRegression(random_state=0)
classifier.fit(x_train,y_train)
```

Out[38]:

```
LogisticRegression(random_state=0)
```

Predict the test set results

In [39]:

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

Accuracy

In [40]:

```
cm=confusion_matrix(y_test,y_pred)
print(cm)
acc=(sum(np.diag(cm))/len(y_test))
print(acc)
```

```
[[74  5]
 [11 30]]
0.8666666666666667
```

or else

In [41]:

```
accuracy=accuracy_score(y_test,y_pred)
print(accuracy)
```

```
0.8666666666666667
```