In [12]:

**import** pandas **as** pd

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.metrics **import** confusion\_matrix, accuracy\_score, precision\_score, recall\_sc

In [17]:

df **=** pd**.**read\_csv("Social\_Network\_Ads.csv") df["Gender"]**.**replace({"Male":0,"Female":1},inplace**=True**)

In [18]:

df

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[18]: | **User ID** | **Gender** | **Age** | **EstimatedSalary** | **Purchased** |
|  | **0** 15624510 | 0 | 19 | 19000 | 0 |
|  | **1** 15810944 | 0 | 35 | 20000 | 0 |
|  | **2** 15668575 | 1 | 26 | 43000 | 0 |
|  | **3** 15603246 | 1 | 27 | 57000 | 0 |
|  | **4** 15804002 | 0 | 19 | 76000 | 0 |
|  | **...** ... | ... | ... | ... | ... |
|  | **395** 15691863 | 1 | 46 | 41000 | 1 |
|  | **396** 15706071 | 0 | 51 | 23000 | 1 |
|  | **397** 15654296 | 1 | 50 | 20000 | 1 |
|  | **398** 15755018 | 0 | 36 | 33000 | 0 |
|  | **399** 15594041 | 1 | 49 | 36000 | 1 |

400 rows × 5 columns

In [21]:

Out[21]:

In [22]:

x **=** df[['User ID', 'Gender', 'Age', 'EstimatedSalary']]*# independent*

y **=** df[['Purchased']] *# it depndent on x variable*

df**.**columns

Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')

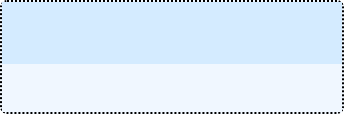
In [23]:

x\_train,x\_test,y\_train,y\_test **=** train\_test\_split(x,y,test\_size**=**0.25,random\_state**=**29)

In [24]:

model **=** LogisticRegression() model**.**fit(x\_train,y\_train)

Out[24]:



▾ LogisticRegression

LogisticRegression()

C:\Users\adesh\DA\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarn ing: 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)

In [25]:

y\_pred **=** model**.**predict(x\_test)

In [26]:

y\_pred

Out[26]:

In [27]:

Out[27]:

In [28]:

Out[28]:

In [29]:

cm **=** confusion\_matrix(y\_test,y\_pred) cm

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| array([0, | 0, | 0, | 0, | 1, | 0, | 1, | 0, | 1, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 1, | 0, | 0, | 0, |
| 1, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 0, |
| 0, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 1, | 0, |
| 0, | 1, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 1, | 0, | 1, | 1, | 0, | 0, |

0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int64)

model**.**score(x\_train,y\_train)

0.7833333333333333

model**.**score(x,y)

0.785

Out[29]:

In [33]:

array([[64, 5],

[16, 15]], dtype=int64)

*#print(confusion\_matrix.* *doc* *)*

*#or*

tn, fp, fn, tp **=** confusion\_matrix(y\_test,y\_pred)**.**ravel()

In [34]:

print(tn,fp,fn,tp)

64 5 16 15

In [35]:

a **=** accuracy\_score(y\_test,y\_pred) a

Out[35]:

In [36]:

e **=** 1**-**a e

0.79

Out[36]:

In [37]:

Out[37]:

In [38]:

Out[38]:

In [ ]:

0.20999999999999996

precision\_score(y\_test,y\_pred)

0.75

recall\_score(y\_test,y\_pred)

0.4838709677419355