```
from google.colab import drive
drive.mount("/content/gdrive")
```

```
     Mounted at /content/gdrive
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/Dataset/Social_Network Ads.csv')
dataset=df
print(dataset)
```

```
           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 x 5 columns]
```

```
print("Number of rows is = ", dataset.shape[0], " \nNumber of columns is = " , dataset.shape[
```

```
     Number of rows is =  400
     Number of columns is =  5
```

```
dataset.columns
```

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

```
dataset.describe()
```

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864

```
dataset.isnull().sum()
```

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

```
correlation = pd.DataFrame(dataset.corr().Purchased)
correlation
```

	Purchased
User ID	0.007120
Age	0.622454
EstimatedSalary	0.362083
Purchased	1.000000

```
dataset.drop('User ID', axis = 1, inplace=True)
```

```
dataset['Purchased'].value_counts()
```

```
0    257
1    143
Name: Purchased, dtype: int64
```

```
pd.crosstab(dataset.Gender,dataset.Purchased,normalize = "index")
```

Purchased	0	1
Gender		
Female	0.622549	0.377451
Male	0.663265	0.336735

```
dataset = pd.get_dummies(dataset, columns = ['Gender'], drop_first=True)
```

```
dataset.head()
```

	Age	EstimatedSalary	Purchased	Gender_Male
0	19	19000	0	1
1	35	20000	0	1
2	26	43000	0	0
3	27	57000	0	0
4	19	76000	0	1

```
X = dataset.drop(['Purchased'],axis = 1)
y = dataset['Purchased']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

```
from sklearn.preprocessing import StandardScaler

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

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0, solver='lbfgs' )
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

print(X_test[:10])
print('-'*15)
print(y_pred[:10])
```

```
[[-0.80480212  0.50496393  1.02020406]
 [-0.01254409 -0.5677824  -0.98019606]
 [-0.30964085  0.1570462   1.02020406]
 [-0.80480212  0.27301877 -0.98019606]
 [-0.30964085 -0.5677824  -0.98019606]
 [-1.10189888 -1.43757673  1.02020406]
 [-0.70576986 -1.58254245 -0.98019606]
 [-0.21060859  2.15757314  1.02020406]
 [-1.99318916 -0.04590581 -0.98019606]
 [ 0.8787462  -0.77073441  1.02020406]]
---------------
[0 0 0 0 0 0 0 1 0 1]
```

```
print(y_pred[:20])
print(y_test[:20])
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0]
132    0
309    0
341    0
196    0
246    0
60     0
155    0
261    1
141    0
214    0
37     0
134    0
113    0
348    0
12     0
59     0
293    0
140    0
206    1
199    0
Name: Purchased, dtype: int64
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[65  3]
 [ 7 25]]
```

```
from sklearn import metrics
print(metrics.classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.96      0.93        68
           1       0.89      0.78      0.83        32

    accuracy                           0.90       100
   macro avg       0.90      0.87      0.88       100
weighted avg       0.90      0.90      0.90       100
```

● ✕