```
import pandas as pd
import numpy as np
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_score
```

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

```
df
```

```
      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]
```

```
df.columns
```

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

```
df['Gender'] = df['Gender'].replace({'Female': 0, 'Male': 1})
```

```
df
```

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

[400 rows x 5 columns]
```

```
x=df[['User ID', 'Gender', 'Age', 'EstimatedSalary']]
```

```
y=df['Purchased']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,rand
om_state=42)
```

```
model=LogisticRegression()
```

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

```
LogisticRegression()
```

```
y_predict=model.predict(x_test)
y_predict
```

```
array([0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0,
0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0], dtype=int64)
```

```
model.score(x_train,y_train)
```

```
0.84
```

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

```
0.85
```

```
model.score(x_test,y_test)
```

```
0.88
```

```
cm=confusion_matrix(y_test,y_predict)
```

```
cm
```

```
array([[61,  2],
       [10, 27]], dtype=int64)
```

```
tn,fp,fn,tp=confusion_matrix(y_test,y_predict).ravel()
```

```
print(tn,fp,fn,tp)
```

```
61 2 10 27
```

```
a=accuracy_score(y_test,y_predict)
```

```
e=1-a
```

```
e
```

```
0.12

precision_score(y_test,y_predict)

0.9310344827586207

recall_score(y_test,y_predict)

0.7297297297297297
```