```python
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

```python
In [2]: df=pd.read_csv('Social_Network_Ads.csv')
```

```python
In [3]: df.head()
```
Out[3]:

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

```python
In [4]: X = df.iloc[:, [0, 1]].values    #k
        y= df.iloc[:, 2].values
```

```python
In [5]: print(X[:3, :])
        print('-'*15)
        print(y[:3])

        [[   19 19000]
         [   35 20000]
         [   26 43000]]
        ---------------
        [0 0 0]
```

```python
In [6]: from sklearn.model_selection import train_test_split    #c
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, ra
```

```python
In [7]: from sklearn.preprocessing import StandardScaler
        sc = StandardScaler()
        X_train = sc.fit_transform(X_train)
        X_test = sc.transform(X_test) #a
```

```python
In [8]: print(X_train[:3])
        print('-'*15)
        print(X_test[:3])

        [[ 0.58164944 -0.88670699]
         [-0.60673761  1.46173768]
         [-0.01254409 -0.5677824 ]]
        ---------------
        [[-0.80480212  0.50496393]
         [-0.01254409 -0.5677824 ]
         [-0.30964085  0.1570462 ]]
```

```python
In [9]:  from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression(random_state = 0)
         classifier.fit(X_train, y_train)
```

Out[9]:
```
▾           LogisticRegression
LogisticRegression(random_state=0)
```

```python
In [10]: y_pred = classifier.predict(X_test) #1
```

```python
In [11]: print(X_test[:10])
         print('-'*15)
         print(y_pred[:10])
```

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

```python
In [12]: print(y_pred[:20]) #v
         print(y_test[:20])
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0]
[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0]
```

```python
In [13]: from sklearn.metrics import confusion_matrix,classification_report
         cm = confusion_matrix(y_test, y_pred)
         cm
```

Out[13]:
```
array([[65,  3],
       [ 8, 24]], dtype=int64)
```

```python
In [14]: cl_report=classification_report(y_test,y_pred)
         cl_report
```

Out[14]:
```
'              precision    recall  f1-score   support\n\n           0
0.89      0.96      0.92        68\n           1       0.89      0.75      0.
81        32\n\n    accuracy                           0.89       100\n    mac
ro avg       0.89      0.85      0.87       100\nweighted avg       0.89
0.89      0.89       100\n'
```