Data Analytics II

Implement logistic regression using Python/R to perform classiﬁcation on Social\_Network\_Ads.csv dataset.

Compute Confusion matrix to ﬁnd TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

**In [9]:**

import pandas as pd import numpy as np

import matplotlib.pyplot as plt

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

%matplotlib inline warnings.filterwarnings('ignore')

**In [4]: df**

=

pd.read\_csv('Social\_Network\_Ads.csv')

**In [5]:**

**Out [5]:**

**In [6]:**

**Out [6]:**

df.head()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

df.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 |
| min | 1.556669e+07 | 18.000000 | 15000.000000 | 0.000000 |
| 25% | 1.562676e+07 | 29.750000 | 43000.000000 | 0.000000 |
| 50% | 1.569434e+07 | 37.000000 | 70000.000000 | 0.000000 |
| 75% | 1.575036e+07 | 46.000000 | 88000.000000 | 1.000000 |

User ID Age EstimatedSalary Purchased

**In [8]:**

max 1.581524e+07 60.000000 150000.000000 1.000000

# input

x = df.iloc[:, [2, 3]].values

# output

y = df.iloc[:, 4].values

**In [10]:**

**X\_train, X\_test, y\_train, y\_test train\_test\_split(x, y, te**

**=**

**In [14]:**

from sklearn.preprocessing import StandardScaler

sc\_x X\_train X\_test

**=**

**=**

=

StandardScaler() sc\_x.fit\_transform(X\_train)

sc\_x.transform(X\_test)

print (X\_train[0:10, :])

|  |  |
| --- | --- |
| **[[ 0.58164944** | **-0.88670699]** |
| **[-0.60673761** | **1.46173768]** |
| **[-0.01254409** | **-0.5677824 ]** |
| **[-0.60673761** | **1.89663484]** |
| **[ 1.37390747** | **-1.40858358]** |
| **[ 1.47293972** | **0.99784738]** |
| **[ 0.08648817** | **-0.79972756]** |
| **[-0.01254409** | **-0.24885782]** |
| **[-0.21060859** | **-0.5677824 ]** |
| **[-0.21060859** | **-0.19087153]]** |

**In [15]:**

from sklearn.linear\_model import LogisticRegression

classifier LogisticRegression(random\_state 0)

=

=

classifier.fit(X\_train, y\_train)

**Out [15]: LogisticRegression(random\_state=0)**

=

**In [16]:**

y\_pred

classifier.predict(X\_test)

**In [17]:**

from sklearn.metrics import confusion\_matrix

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

=

print ("Confusion Matrix : \n", cm)

**Confusion Matrix : [[ 0 68]**

**[ 0 32]]**

**In [18]:**

**from sklearn.metrics import accuracy\_score**

**print ("Accuracy : ", accuracy\_score(y\_test, y\_pred))**

**Accuracy : 0.32**

Compute Confusion matrix to ﬁnd TP, FP, TN, FN, Accuracy, Error rate, Precision,

Recall on the given dataset.

**In [25]:**

**# classification report for precision, recall f1-score and a from sklearn.metrics import classification\_report**

**matrix classification\_report(y\_test, y\_pred,labels=[1,0]) print('Classification report : \n',matrix)**

**=**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification** | **report : precision** | **recall** | **f1-score** | **support** |
| **1** | **0.32** | **1.00** | **0.48** | **32** |
| **0** | **0.00** | **0.00** | **0.00** | **68** |
| **accuracy** |  |  | **0.32** | **100** |
| **macro avg** | **0.16** | **0.50** | **0.24** | **100** |
| **weighted avg** | **0.10** | **0.32** | **0.16** | **100** |