In [1]:

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

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

In [4]:

dataset **=** pd.read\_csv(r"D:\College\TE\SEM-2\Practical\DSBDA\5\Social\_Network\_Ads.csv")

In [5]:

dataset.head()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[5]: |  | | | | | |
|  |  | **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 |

In [6]:

dataset.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 400 entries, 0 to 399

Data columns (total 5 columns):

User ID 400 non-null int64

Gender 400 non-null object

Age 400 non-null int64 EstimatedSalary 400 non-null int64 Purchased 400 non-null int64 dtypes: int64(4), object(1)

memory usage: 15.8+ KB

In [7]:

dataset.isnull().sum()

Out[7]: User ID 0

Gender 0

Age 0

EstimatedSalary 0

Purchased 0

dtype: int64

In [8]:

dataset.shape

Out[8]: (400, 5)

In [10]:

x **=** dataset.iloc[:, [2, 3]].values y **=** dataset.iloc[:, 4].values

In [11]:



print(x)

|  |  |  |
| --- | --- | --- |
| [[ | 19 | 19000] |
| [ | 35 | 20000] |
| [ | 26 | 43000] |
| [ | 27 | 57000] |
| [ | 19 | 76000] |
| [ | 27 | 58000] |
| [ | 27 | 84000] |
| [ | 32 | 150000] |
| [ | 25 | 33000] |
| [ | 35 | 65000] |
| [ | 26 | 80000] |
| [ | 26 | 52000] |
| [ | 20 | 86000] |
| [ | 32 | 18000] |
| [ | 18 | 82000] |
| [ | 29 | 80000] |
| [ | 47 | 25000] |
| [ | 45 | 26000] |
| [ | 46 | 28000] |

In [12]:

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

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x, y, test\_size **=** 0.30, random\_stat

In [13]:

**from** sklearn.preprocessing **import** StandardScaler sc **=** StandardScaler()

x\_train **=** sc.fit\_transform(x\_train) x\_test **=** sc.transform(x\_test)

In [14]:

**from** sklearn.linear\_model **import** LogisticRegression classifier **=** LogisticRegression(random\_state **=** 0)

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

C:\Users\HP\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:432: FutureWa rning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

FutureWarning)

Out[14]: LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,

intercept\_scaling=1, l1\_ratio=None, max\_iter=100, multi\_class='warn', n\_jobs=None, penalty='l2',

random\_state=0, solver='warn', tol=0.0001, verbose=0, warm\_start=False)

In [15]:

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

In [16]:

print(y\_pred)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

In [17]:

print(y\_test)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

In [18]:

**from** sklearn.metrics **import** confusion\_matrix cm **=** confusion\_matrix(y\_test, y\_pred)

print(cm)

[[74 5]

[10 31]]

In [19]:

Accuracy**=**(74**+**31)**/**120 Accuracy

Out[19]: 0.875

In [20]:

Error\_rate**=**(5**+**10)**/**120 Error\_rate

Out[20]: 0.125

In [21]:

**from** sklearn.metrics **import** precision\_score, recall\_score precision\_score(y\_test, y\_pred)

Out[21]: 0.8611111111111112

In [22]:

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

Out[22]: 0.7560975609756098

In [23]:

**from** sklearn.metrics **import** f1\_score f1\_score(y\_test, y\_pred)

Out[23]: 0.8051948051948052

In [ ]: