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

In [2]:
```
sp=pd.read_csv("/home/student/Social_Network_Ads.csv")
```

In [3]:
```
sp.head(6)
```

Out[3]:

	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
5	15728773	Male	27	58000	0

In [4]:
```
sp
```

Out[4]:

	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 × 5 columns

In [5]:
```
from sklearn import preprocessing
sp['Gender'].unique()
```

Out[5]:
```
array(['Male', 'Female'], dtype=object)
```

In [6]:
```
label_encoder=preprocessing.LabelEncoder()
sp['Gender']=label_encoder.fit_transform(sp['Gender'])
```

In [7]:
```
sp['Gender'].unique()
```

Out[7]:
```
array([1, 0])
```

In [8]:
```
features_sp=sp.drop(columns=['Gender'])
```

In [9]:
```
enc=preprocessing.OneHotEncoder()
```

In [10]:
```
sp.head()
```

Out[10]:

	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

In [11]:
```
sp.isnull()
```

Out[11]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
395	False	False	False	False	False
396	False	False	False	False	False
397	False	False	False	False	False
398	False	False	False	False	False
399	False	False	False	False	False

400 rows × 5 columns

In [12]:
```
x=sp.drop(['Purchased'],axis=1)
y=sp['Purchased']
```

In [13]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [14]:
```
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
```

In [16]:
```
logreg.fit(xtrain,ytrain)
```

Out[16]:
```
LogisticRegression()
```

In [17]:
```
LogisticRegression(C=1.0,class_weight=None,dual=False,fit_intercept=True,intercept_scaling=1,max_iter=100,multi_class='ovr',n_jobs=1,penalty='12',random_state=None,solver='liblinear',tol=0.0001,verbose=0,warm_start=False)
ytrain_pred=logreg.predict(xtrain)
ytest_pred=logreg.predict(xtest)
```

In [18]:
```
sp=pd.DataFrame(ytrain_pred,ytrain)
sp=pd.DataFrame(ytest_pred,ytest)
```

In [19]:
```
y_pred=logreg.predict(xtest)
```

In [20]:
```
from sklearn.metrics import precision_score,confusion_matrix,accuracy_score,recall_score
accuracy=accuracy_score(ytest,y_pred)
precision=precision_score(ytest,y_pred,average="micro")
recall=recall_score(ytest,y_pred,average="micro")
cm=confusion_matrix(ytest,y_pred)
```

In [21]:
```
accuracy
```

Out[21]:
```
0.825
```

In [22]:
```
precision
```

Out[22]:
```
0.825
```

In [23]:
```
recall
```

Out[23]:
```
0.825
```

In [24]:
```
cm
```

Out[24]:
```
array([[56,  2],
       [12, 10]])
```

In [ ]: