5. Title of the Assignment:

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 [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
dataset = pd.read_csv('F:/SWATI ENGG/2021-2022/DS and Big Data/PRACTICALS/Codes/Social
dataset.head()
```

In [2]:

```
#X = dataset.iloc[:, [2, 3]].values
#y = dataset.iloc[:, 4].values

#print(X[:3, :])
#print('-'*15)
#print(y[:3])
dataset.shape
```

Out[2]:

```
(400, 5)
```

In [3]:

```
dataset.isnull().sum()
```

Out[3]:

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [4]:

```
from sklearn import preprocessing
le=preprocessing.LabelEncoder()
```

In [5]:

```
dataset['Gender']=le.fit_transform(dataset['Gender'])
```

In [6]:

```
dataset.head()
```

Out[6]:

	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 [7]:

```
#Split dependent variable and independent variables
X = dataset.drop(['Purchased'], axis = 1)
y = dataset['Purchased']
```

In [8]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_sta

print('Dataset shape =',dataset.shape)
print('X_train shape =',X_train.shape)
print('X_test shape =',X_test.shape)
print('y_train shape =',y_train.shape)
print('y_test shape =',y_test.shape)
```

```
Dataset shape = (400, 5)
X_train shape = (300, 4)
X_test shape = (100, 4)
y_train shape = (300,)
y_test shape = (100,)
```

In [9]:

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

In [22]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

In [11]:

```
print(y_pred[:20])
print(y_test[:20])
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0]
132    0
309    0
341    0
196    0
246    0
60     0
155    0
261    1
141    0
214    0
37     0
134    0
113    0
348    0
12     0
59     0
293    0
140    0
206    1
199    0
Name: Purchased, dtype: int64
```

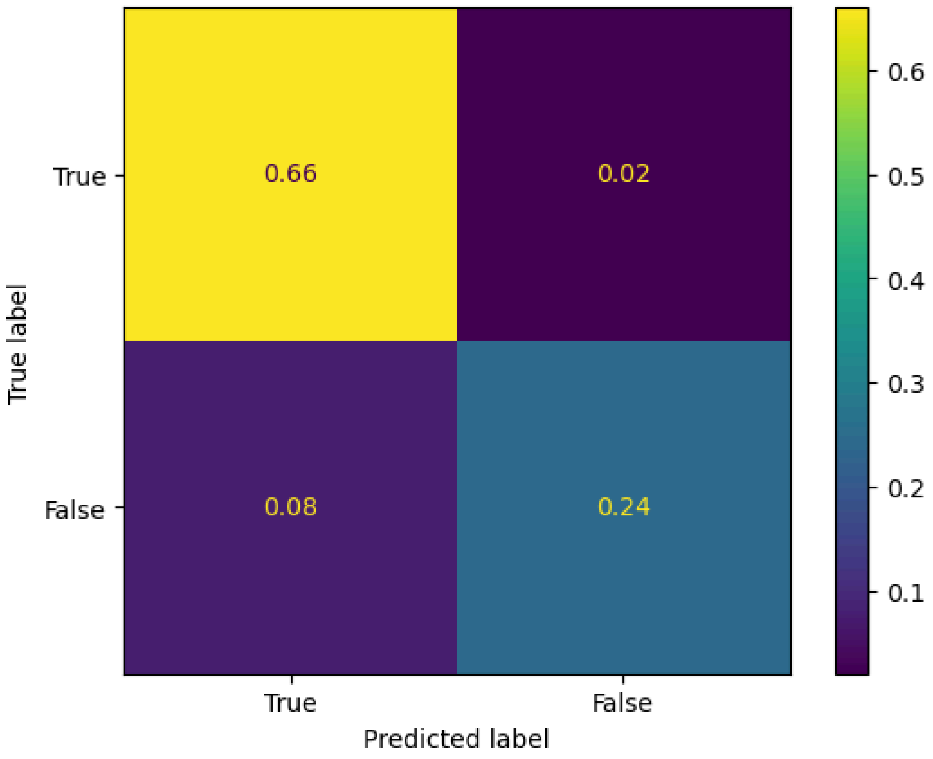
In [24]:

```
from sklearn import metrics
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay,classification_re
#from sklearn.metrics import ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[66  2]
 [ 8 24]]
```

```
In [21]: #from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

         cm = confusion_matrix(y_test, y_pred, normalize='all')
         cmd = ConfusionMatrixDisplay(cm, display_labels=['True','False'])
         cmd.plot()
```

```
Out[21]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27dc85822e8>
```



```
In [27]: print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.97      0.93        68
           1       0.92      0.75      0.83        32

    accuracy                           0.90       100
   macro avg       0.91      0.86      0.88       100
weighted avg       0.90      0.90      0.90       100
```

```
In [ ]:
```