**Assignment No 5**

**Code:**

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import mean\_squared\_error,confusion\_matrix,precision\_recall\_fscore\_support

import numpy as np

from sklearn.preprocessing import MinMaxScaler

import seaborn as sns

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

print(df)

#select features and target variables

X = df[['Age', 'EstimatedSalary']]

y = df['Purchased']

print(X)

print(y)

# fit scaler on training data

norm = MinMaxScaler().fit(X)

# transform training data

X = norm.transform(X)

#splitting data into training and testing set

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=42)

print(y\_train.shape)

#create a linear regression model

model=LogisticRegression()

#fit the model on training data

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

#make prediction on the test data

y\_pred = model.predict(X\_test)

print(y\_pred)

# evaluate the model

mse = mean\_squared\_error(y\_test,y\_pred)

print(mse)

r\_mse = np.sqrt(mse)

print(r\_mse)

plt.scatter(y\_test,y\_pred,color='k')

plt.title('Logistic Regression Visualization')

plt.xlabel('actual values')

plt.ylabel('predicted values')

cf = confusion\_matrix(y\_test, y\_pred)

print(cf)

score = precision\_recall\_fscore\_support(y\_test, y\_pred, average='micro')

print('Precision of Model: ', score[0])

print('Recall of Model: ', score[1])

print('F-Score of Model: ', score[2])

sns.heatmap(cf,annot=True,fmt = 'd',cmap='Blues')

plt.xlabel("Acutual value")

plt.ylabel("predicted values")

plt.title("Confusion Matrics")

plt.show();

**Output:**

    Age  EstimatedSalary  Purchased

0     19            19000          0

1     35            20000          0

2     26            43000          0

3     27            57000          0

4     19            76000          0

..   ...              ...        ...

395   46            41000          1

396   51            23000          1

397   50            20000          1

398   36            33000          0

399   49            36000          1

[400 rows x 3 columns]

     Age  EstimatedSalary

0     19            19000

1     35            20000

2     26            43000

3     27            57000

4     19            76000

..   ...              ...

395   46            41000

396   51            23000

397   50            20000

398   36            33000

399   49            36000

[400 rows x 2 columns]

0      0

1      0

2      0

3      0

4      0

      ..

395    1

396    1

397    1

398    0

399    1

Name: Purchased, Length: 400, dtype: int64

(320,)

[0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0

 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0

 0 0 1 1 0 0]

0.125

0.3535533905932738

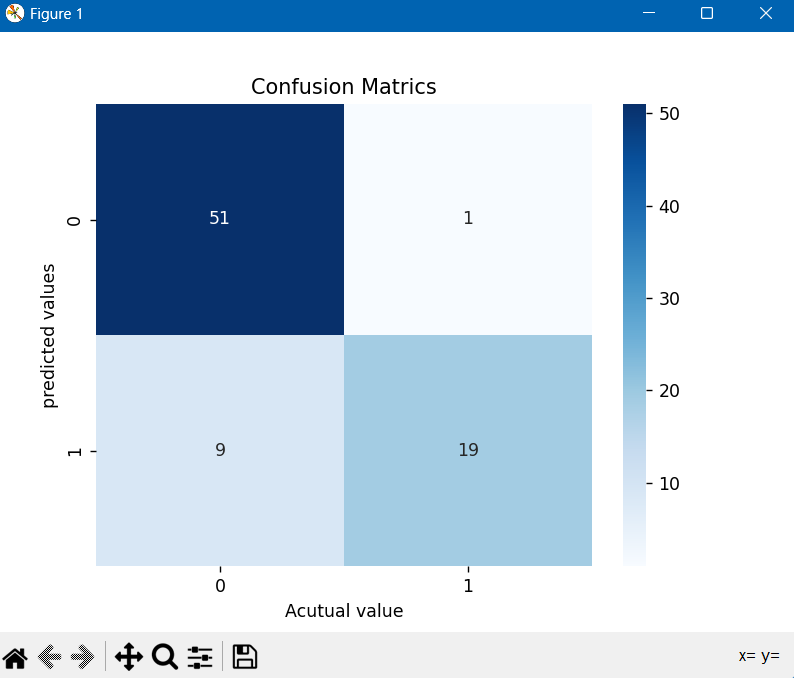
[[51  1]

 [ 9 19]]

Precision of Model:  0.875

Recall of Model:  0.875

F-Score of Model:  0.875

****