Name: Jayesh Rajbhar
Class: TE Comp A
Roll No: 626
Subject : DWDM

Experiment No: 10

Aim: Implementation of Logistic Regression in Python.

Implementation:
Applying Logistic Regression on suv dataset

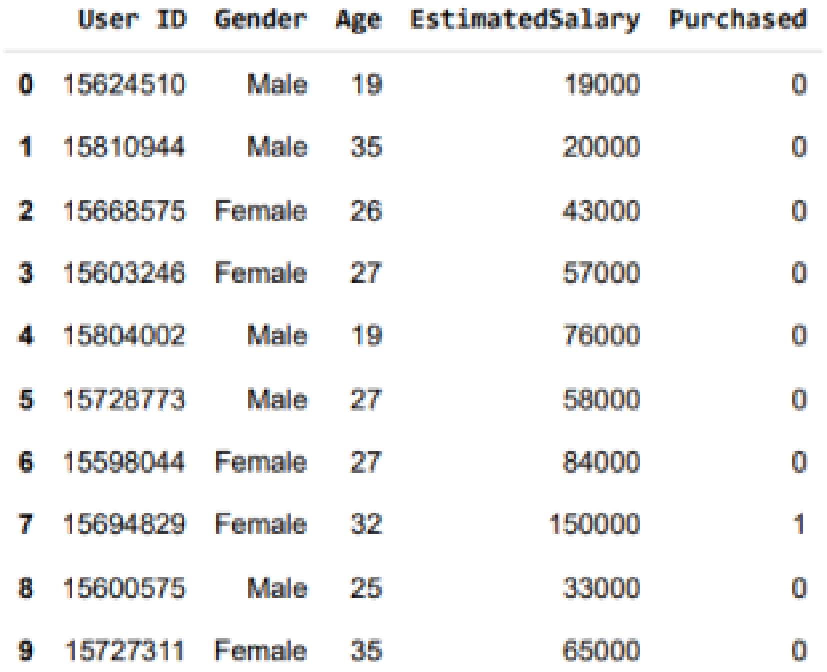
Code:

1) Importing Libraries:

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

2) Loading the dataset:

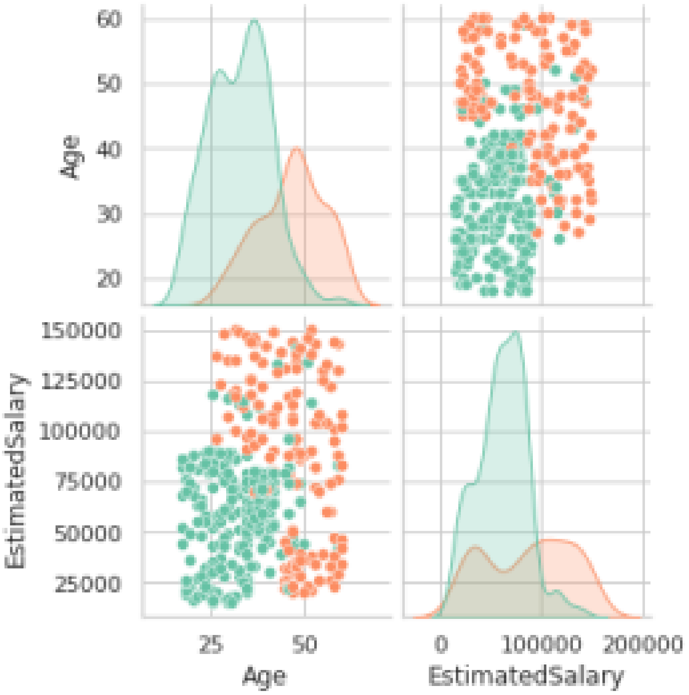
```
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('suv_data.csv')
#Now let’s view our dataset using head(): df.head(10)
```



	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
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

3) Analysing the data:

```
df.shape
(400, 5)
sns.pairplot(df,hue = 'Purchased', palette= 'Set2', vars = ['Age','EstimatedSalary'])
```



4) Slicing (Dependent and Independent Variable):

```
#Extracting Independent and dependent Variable
 x= df.iloc[:, [2,3]].values
 y= df.iloc[:, 4].values
```

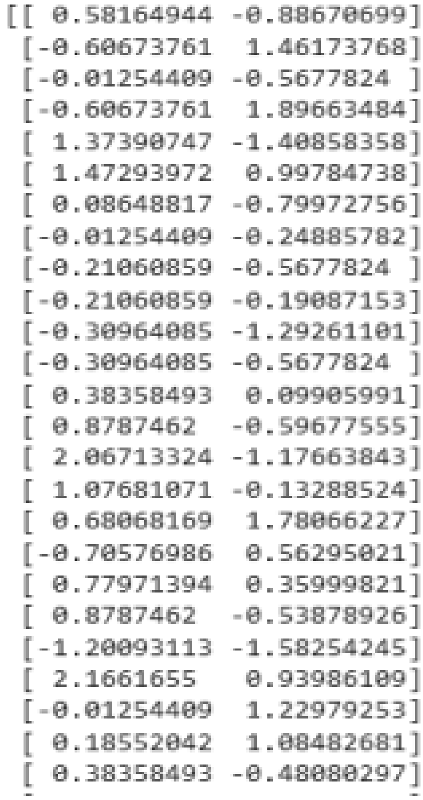
5) Splitting (Training and Testing):

```
 # Splitting the dataset into training and test set.
 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_state=0)
 print('Shape of training data is: ', x_train.shape)
 print('Shape of testing data is: ', x_test.shape)
```

```
 Shape of training data is: (300, 2)
 Shape of testing data is: (100, 2)
```

6) Scaling:

```
 from sklearn.preprocessing import StandardScaler st_x=
StandardScaler()
 x_train= st_x.fit_transform(x_train)
 x_test= st_x.transform(x_test)
 print(x_train)
```



```
[[ 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]
 [-0.30964085 -1.29261101]
 [-0.30964085 -0.5677824 ]
 [ 0.38358493  0.09905991]
 [ 0.8787462  -0.59677555]
 [ 2.06713324 -1.17663843]
 [ 1.07681071 -0.13288524]
 [ 0.68068169  1.78066227]
 [-0.70576986  0.56295021]
 [ 0.77971394  0.35999821]
 [ 0.8787462  -0.53878926]
 [-1.20093113 -1.58254245]
 [ 2.1661655   0.93986109]
 [-0.01254409  1.22979253]
 [ 0.18552042  1.08482681]
 [ 0.38358493 -0.48080297]
```

7) Training:

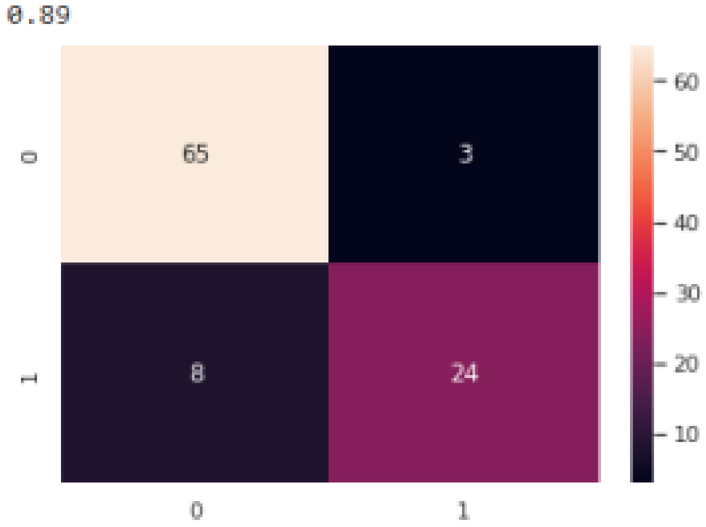
```
from sklearn.linear_model import LogisticRegression
model= LogisticRegression(random_state=0)
model.fit(x_train, y_train)

LogisticRegression(random_state=0)
```

8) Testing (Model Accuracy):

```
y_pred= model.predict(x_test)
from sklearn.metrics import confusion_matrix, accuracy_score y_pred = model.predict(x_test)
y_pred = (y_pred > 0.5)
```

9) Confusion Matrix:

```
# Making the Confusion Matrix:
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
sns.heatmap(cm, annot = True)
print(accuracy_score(y_test, y_pred))
```



```
0.89
```

Conclusion: Hence we successfully implemented Logistic Regression in Python.