**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Machine Learning Program: B.Tech/MBA.Tech**

**Sem: III/V ACAY: 2020-21**

**EXPERIMENT NO. 4**

**Aim:**

1. To load a data set using pandas in python.
2. To be able to use logistic regression using sigmoid function and view predicted output

**Software:**  PYTHON.

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Knowledge of Logistic regression and Sigmoid function |

**Outcome:**

After successful completion of this experiment students will be able to:

1. Load and visualize dataset in Pyhton using Pandas.
2. Implement linear and logistic regression by using sklearn package and hard coding
3. Determine accuracy of the model

**Theory:**

* The data which we will be using for our **linear regression** example is in a .csv file called: ‘sat\_cgpa.
* For loading the data use the command:

df=pd.read\_csv('data\_logistics\_1.csv')

**TO BE COMPLETED BY STUDENTS**

* Students must upload the soft copy of the program in the given format.

|  |
| --- |
| Name of the Experiment: Experiment 4 – Logistic Regression |
| Roll No: N049 Name: Tarun Tanmay |
| Program: MBATech CE Semester: 5 |
| Date of Performance: 14/8/20 Date of Submission: 20/9/20 |

**Step 1: Importing the Relevant Libraries**

#experiment 4

#Logistic regression

import pandas as pd

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

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

### ****Step 2: Loading the Data****

### df=pd.read\_csv('data\_logistics\_1.csv')

### print(df)

**Step 3: Visualizing the Data Frame**

A screenshot of a cell phone

Description automatically generated

### ****Step 4: Exploring the Data****

**x1=df['gmat']**

**y1=df['admitted']**

**x1=np.array(x1)**

**y1=np.array(y1)**

**x1=x1.reshape((-1,1)) #All Rows(-1) and (1)column**

**y1=y1.reshape((-1,1))**

**#y1.shape**

### ****Step 5: Create a linear regression model****

**model=LinearRegression()**

**model.fit(x1,y1)**

**b0=model.intercept\_**

**b1=model.coef\_**

**#predicted output is yhat**

**yhat=b0+(b1\*x1)**

**z=yhat**

**#np.mean(yhat)**

**z=yhat-np.mean(yhat)**

**#sigmoid of z is in the range (0,1)**

**#value of z is in (-inf, +inf)**

**#or mean value of z should be 0**

**sigm=1/(1+np.exp(-z))**

**sh=sigm.shape**

**row=sh[0]**

**col=sh[1]**

**#decision is based on the value of sigmoid**

**output=np.zeros((sh[0],1))**

**for i in range(0,sh[0]):**

**if sigm[i]>0.5:**

**output[i]=1**

**else:**

**output[i]=0**

**Step 6: Plot results**

plt.scatter(yhat,y1)

plt.plot(yhat,sigm)

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated…..

A picture containing keyboard

Description automatically generated

### ****Step 7: Creating Logistic Regression model****

### from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

import seaborn as sn

df=pd.read\_csv('data\_logistics.csv')

df

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25,random\_state=0)

model=LogisticRegression()

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

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

confusion\_matrix=pd.crosstab(y\_test,y\_pred,rownames=['Actual'],colnames=['Predicted'])

sn.heatmap(confusion\_matrix, annot=True)

### ****Step 8: display confusion matrix****

A screenshot of a cell phone

Description automatically generated

### ****Step 10:The Standard Errors****

error=0

for i in range(sh[0]):

if output[i][0]!=y1[i][0]:

error=error+1

error

### A screenshot of a cell phone Description automatically generated

**Step 11: Conclusion**

A screenshot of a cell phone

Description automatically generated