

**Implementation of Logistic Regression**

**ABUBAKAR ASIF**

**#AbubakarAsif FA20-BCE-013**

**#abubakarasif3111@gmail.com**

**#https://github.com/Abubakar3111**

**#https://www.linkedin.com/in/abubakar-asif-b3b94021a/**

**Artificial Intelligence**

**SUBMITTED TO:DR SHOAIB AZMAT**

**DATE:4th Jan,2023**

**# Implementation of Logistic Regression**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import numpy.linalg as LA

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion\_matrix, plot\_confusion\_matrix, mean\_squared\_error

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

data = pd.read\_csv('currency.csv')

print(data.shape)

print(data)

x1 = data['v'].values

x2 = data['s'].values

x3 = data['k'].values

x4 = data['e'].values

Y = data['class'].values

m = len(x1)

x1 = x1.reshape(m)

x2 = x1.reshape(m)

x3 = x1.reshape(m)

x4 = x1.reshape(m)

x0=np.ones(m)

X = np.array([x0,x1,x1\*2,x2,x2\*2,x3,x3\*2,x4,x4\*2]).T ############################linear

v=3.62160

s=8.66610

k=-2.8073

e=-0.44699

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.20)

reg=LogisticRegression()

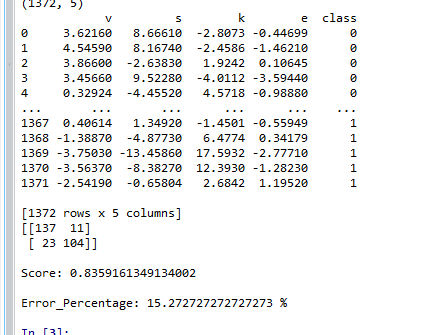
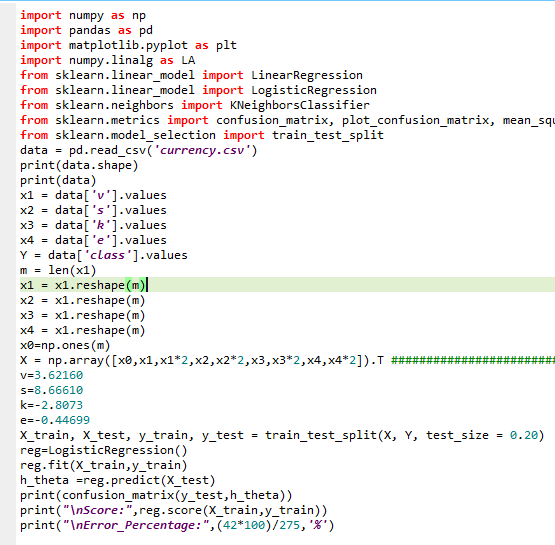
reg.fit(X\_train,y\_train)

h\_theta =reg.predict(X\_test)

print(confusion\_matrix(y\_test,h\_theta))

print("\nScore:",reg.score(X\_train,y\_train))

print("\nError\_Percentage:",(42\*100)/275,'%')



**Second :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import numpy.linalg as LA

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion\_matrix, plot\_confusion\_matrix, mean\_squared\_error

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

data = pd.read\_csv('spine.csv')

print(data.shape)

print(data)

x1 = data['Col1'].values

x2 = data['Col2'].values

x3 = data['Col3'].values

x4 = data['Col4'].values

x5 = data['Col5'].values

x6 = data['Col6'].values

x7 = data['Col7'].values

x8 = data['Col8'].values

x9 = data['Col9'].values

x10 = data['Col10'].values

x11= data['Col11'].values

x12= data['Col12'].values

Y = data['Class\_label'].values

m = len(x1)

x1 = x1.reshape(m)

x2 = x2.reshape(m)

x3 = x3.reshape(m)

x4 = x4.reshape(m)

x5 = x5.reshape(m)

x6 = x6.reshape(m)

x7 = x7.reshape(m)

x8 = x8.reshape(m)

x9 = x9.reshape(m)

x10 = x10.reshape(m)

x11= x11.reshape(m)

x12= x12.reshape(m)

x0=np.ones(m)

X = np.array([x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12]).T ############################linear

x1=63.0278175

x2= 22.55258597

x3= 39.60911701

x4= 40.47523153

x5= 98.67291675

x6= -0.254399986

x7= 0.744503464

x8= 12.5661

x9= 14.5386

x10=15.30468

x11=-28.658501

x12=43.5123

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.20)

reg=LogisticRegression()

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

h\_theta =reg.predict(X\_test)

print(confusion\_matrix(y\_test,h\_theta))

print("\nScore:",reg.score(X\_train,y\_train))

