**G. H. RAISONI COLLEGE OF ENGG., NAGPUR**

**(An Autonomous Institute)**

**Department of Computer Science & Engg.**



**Date: 21-09-2021**

**Practical Subject: Skill Development-2 [BCSP318]**

**Session: 2021-22**

**Student Details:**

| **Roll Number** | 01 |
| --- | --- |
| **Name** | Anand Suralkar |
| **Semester** | 9th |
| **Section** | A |
| **Batch** | CSE |

**Practical Details: Practical Number-9;**

| Practical Aim | Implement Logistic Regression in python |
| --- | --- |
| Theory & Syntax | Logistic Regression using Python  Difficulty Level : Medium  Last Updated : 29 Apr, 2019  Prerequisite: Understanding Logistic Regression  User Database – This dataset contains information of users from a companies database. It contains information about UserID, Gender, Age, EstimatedSalary, Purchased. We are using this dataset for predicting that a user will purchase the company’s newly launched product or not.  Data – User\_Data  Let’s make the Logistic Regression model, predicting whether a user will purchase the product or not.  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  Loading dataset – User\_Data  dataset = pd.read\_csv('...\\User\_Data.csv')  Now, to predict whether a user will purchase the product or not, one needs to find out the relationship between Age and Estimated Salary. Here User ID and Gender are not important factors for finding out this.  # input  x = dataset.iloc[:, [2, 3]].values    # output  y = dataset.iloc[:, 4].values  Splitting the dataset to train and test. 75% of data is used for training the model and 25% of it is used to test the performance of our model.  from sklearn.cross\_validation import train\_test\_split  xtrain, xtest, ytrain, ytest = train\_test\_split(  x, y, test\_size = 0.25, random\_state = 0)  Now, it is very important to perform feature scaling here because Age and Estimated Salary values lie in different ranges. If we don’t scale the features then Estimated Salary feature will dominate Age feature when the model finds the nearest neighbor to a data point in data space.  from sklearn.preprocessing import StandardScaler  sc\_x = StandardScaler()  xtrain = sc\_x.fit\_transform(xtrain)  xtest = sc\_x.transform(xtest)    print (xtrain[0:10, :])  Output :  [[ 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]]  Here once see that Age and Estimated salary features values are sacled and now there in the -1 to 1. Hence, each feature will contribute equally in decision making i.e. finalizing the hypothesis.  Finally, we are training our Logistic Regression model.  from sklearn.linear\_model import LogisticRegression  classifier = LogisticRegression(random\_state = 0)  classifier.fit(xtrain, ytrain)  After training the model, it time to use it to do prediction on testing data.  y\_pred = classifier.predict(xtest)  Let’s test the performance of our model – Confusion Matrix  from sklearn.metrics import confusion\_matrix  cm = confusion\_matrix(ytest, y\_pred)    print ("Confusion Matrix : \n", cm)  Output :  Confusion Matrix :  [[65 3]  [ 8 24]]  Out of 100 :  TruePostive + TrueNegative = 65 + 24  FalsePositive + FalseNegative = 3 + 8  Performance measure – Accuracy  from sklearn.metrics import accuracy\_score  print ("Accuracy : ", accuracy\_score(ytest, y\_pred))  Output :  Accuracy : 0.89 |
| Program | import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  dataset = pd.read\_csv('User\_Data.csv')  # input  x = dataset.iloc[:, [2, 3]].values  # output  y = dataset.iloc[:, 4].values  from sklearn.preprocessing import StandardScaler  from sklearn.model\_selection import train\_test\_split  xtrain, xtest, ytrain, ytest = train\_test\_split(  x, y, test\_size = 0.25, random\_state = 0)  sc\_x = StandardScaler()  xtrain = sc\_x.fit\_transform(xtrain)  xtest = sc\_x.transform(xtest)  from sklearn.linear\_model import LogisticRegression  classifier = LogisticRegression(random\_state = 0)  classifier.fit(xtrain, ytrain)  print(xtrain[0:10, :])  from matplotlib.colors import ListedColormap  X\_set, y\_set = xtest, ytest  X1, X2 = np.meshgrid(np.arange(start=X\_set[:, 0].min() - 1,  stop=X\_set[:, 0].max() + 1, step=0.01),  np.arange(start=X\_set[:, 1].min() - 1,  stop=X\_set[:, 1].max() + 1, step=0.01))  plt.contourf(X1, X2, classifier.predict(  np.array([X1.ravel(), X2.ravel()]).T).reshape(  X1.shape), alpha=0.75, cmap=ListedColormap(('red', 'green')))  plt.xlim(X1.min(), X1.max())  plt.ylim(X2.min(), X2.max())  for i, j in enumerate(np.unique(y\_set)):  plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],  c=ListedColormap(('red', 'green'))(i), label=j)  plt.title('Classifier (Test set)')  plt.xlabel('Age')  plt.ylabel('Estimated Salary')  plt.legend()  plt.show() |
| Output |  |
| Conclusion | Implemented Logistic Regression in python using sklearn , matplotlib and numpy |