|  |  |  |  |
| --- | --- | --- | --- |
|  | Business Intelligence  Practical #10 | | |
|  |  |  |  |
| **Name** | Farman Farooq Kazi | **Roll Number** | 21302B0049 |
| **Class** | TY BSc IT | **Division** | A |
| **Subject/Course:** | Business intelligence | | |
| **Topic** | Logistic Regression | | |
|  |  |  |  |
| **Overview of Logistic Regression**  **What are the steps to perform Logistic Regression?** | | | |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  dataset = pd.read\_csv('/content/User\_Data.csv')  dataset    #Input  x = dataset.iloc[:, [2, 3]].values  #Output  y = dataset.iloc[:, 4].values  from sklearn.model\_selection import train\_test\_split  xtrain, xtest, ytrain, ytest = train\_test\_split(x, y, test\_size = 0.25, random\_state = 0)  from sklearn.preprocessing import StandardScaler  sc\_x = StandardScaler()  xtrain = sc\_x.fit\_transform(xtrain)  xtest = sc\_x.transform(xtest)  print(xtrain[0 : 10, :])    from sklearn.linear\_model import LogisticRegression  classifier = LogisticRegression(random\_state = 0)  classifier.fit(xtrain, ytrain)  y\_pred = classifier.predict(xtest)  from sklearn.metrics import confusion\_matrix  cm = confusion\_matrix(ytest, y\_pred)  print("Confusion Matrix:\n", cm)    from sklearn.metrics import accuracy\_score  print("Accuracy: ", accuracy\_score(ytest, y\_pred))    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.xlim(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() | | | |
|  | | | |