**Pattern Recognition and Machine Learning**

**Name: Rajat Kumar ID: 201811024 Submission Date: 9th Apr. 2019**

**ASSIGNMENT 7**

**(1) Use IRIS data to generate training data and test data. Choose 40 random data points from**

**each class for training data and use remaining 10 data points from each class as test data.**

**(2) Perform k-NN classification on Test data.**

**(3) Use in-built function to perform logistic regression on training data and classify test data.**

**Code:**

|  |
| --- |
| **import pandas as pd import numpy as np from sklearn import datasets from sklearn.neighbors import KNeighborsClassifier from sklearn.model\_selection import StratifiedKFold from sklearn.linear\_model import LogisticRegression    iris=datasets.load\_iris() x=iris.data[:,:4] y=iris.target df=pd.DataFrame(x,columns=['sepal-length','sepal-width','petal-length','petal-width']) df['class']=y from sklearn.model\_selection import train\_test\_split SEED = 201 x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=.20, random\_state=SEED,stratify=y)   skf = StratifiedKFold(n\_splits=3, random\_state=None,shuffle=True) # X is the feature set and y is the target k=3  knnscores=[] for train\_index, test\_index in skf.split(x,y):  #k-1 fold(X\_train,y\_train)  X\_train, X\_test = x[train\_index], x[test\_index]  #kth fold(X\_test,y\_test)  Y\_train, Y\_test = y[train\_index], y[test\_index]  model = KNeighborsClassifier(n\_neighbors=k)  #Fitting the Model  model.fit(X\_train, Y\_train)  Y\_pred = model.predict(X\_test)  knnscores.append(model.score(X\_test,Y\_test)) print('Average 3-Fold Score:' ,np.mean(knnscores))**  **clf = LogisticRegression(random\_state=0, solver='lbfgs', max\_iter=1000,multi\_class='multinomial').fit(x\_train, y\_train) y\_pred=clf.predict(x\_test) from sklearn.metrics import classification\_report print(classification\_report(y\_test,y\_pred,target\_names=['class 0','class 1','class 2']))**  **Output:**  **KNN:**  0.9683333333333332  **Logistic Regression:**    Accuracy Score: 0.9666666666666667 |