Simran Shilky

9918103027

F1

**Assignment 8 Solutions**

Q1)import numpy as np

from scipy.sparse import csr\_matrix

list1=[

[1,0,0],

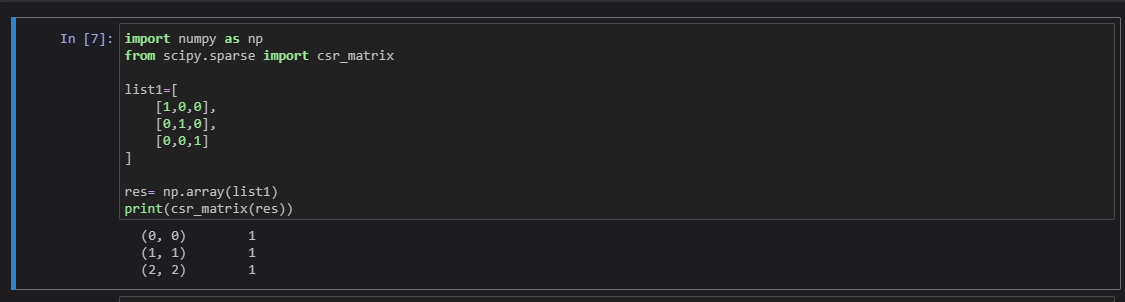
[0,1,0],

[0,0,1]

]

res= np.array(list1)

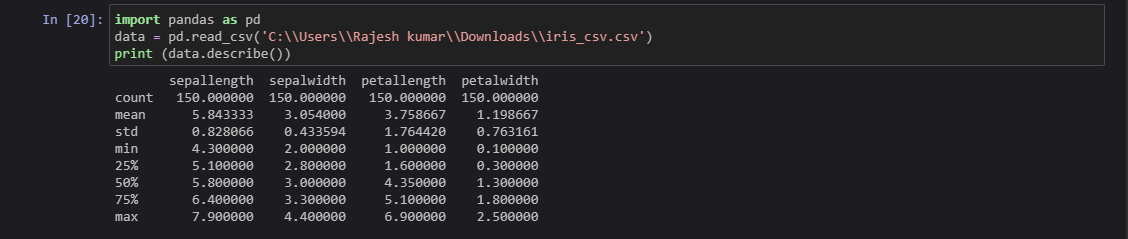
print(csr\_matrix(res))



Q2)import pandas as pd

data = pd.read\_csv('C:\\Users\\Rajesh kumar\\Downloads\\iris\_csv.csv')

print (data.describe())



Q3)import pandas as pd

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

iris = pd.read\_csv("C:\\Users\\Rajesh kumar\\Downloads\\iris\_csv.csv")

X = iris.iloc[:, :-1].values

y = iris.iloc[:, 4].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30)

print("\n70% train data:")

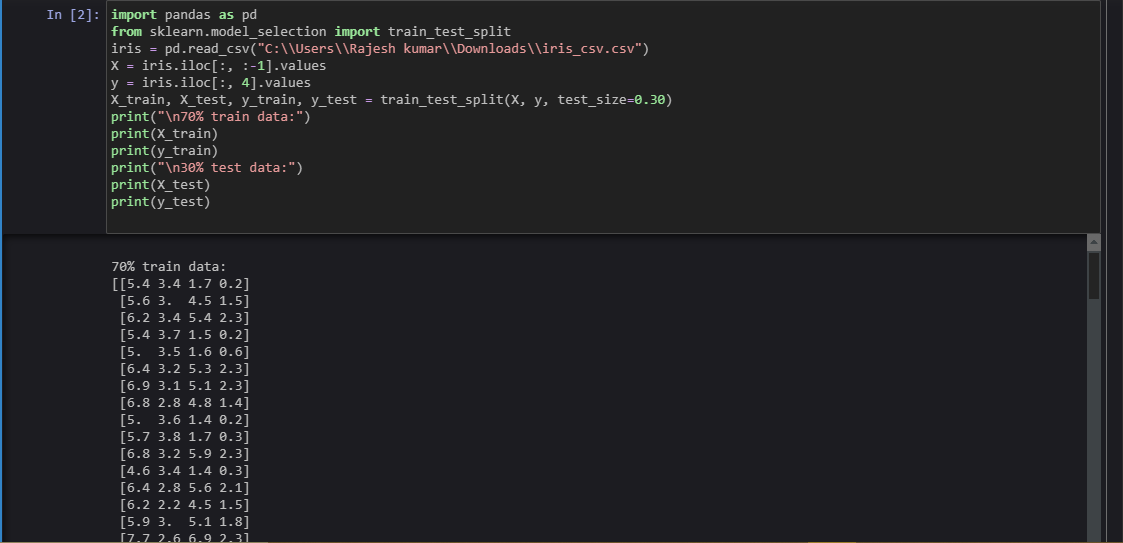
print(X\_train)

print(y\_train)

print("\n30% test data:")

print(X\_test)

print(y\_test)



Q4)import pandas as pd

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

from sklearn.neighbors import KNeighborsClassifier

iris = pd.read\_csv("C:\\Users\\Rajesh kumar\\Downloads\\iris\_csv.csv")

X = iris.iloc[:, :-1].values

y = iris.iloc[:, 4].values

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

#Create KNN Classifier

#Number of neighbors to use by default for kneighbors queries.

knn = KNeighborsClassifier(n\_neighbors=5)

#Train the model using the training sets

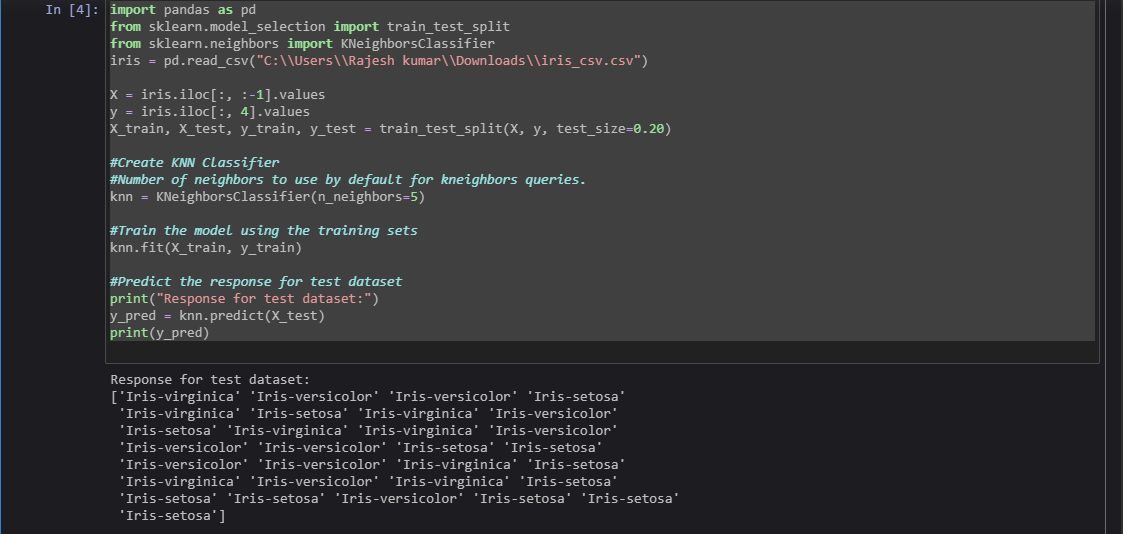
knn.fit(X\_train, y\_train)

#Predict the response for test dataset

print("Response for test dataset:")

y\_pred = knn.predict(X\_test)

print(y\_pred)



Q5)# Importing Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# Importing Data

from sklearn.datasets import load\_boston

boston = load\_boston()

# Input Data

x = boston.data

# Output Data

y = boston.target

# splitting data to training and testing dataset.

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

xtrain, xtest, ytrain, ytest = train\_test\_split(x, y, test\_size =0.2,

random\_state = 0)

print("xtrain shape : ", xtrain.shape)

print("xtest shape : ", xtest.shape)

print("ytrain shape : ", ytrain.shape)

print("ytest shape : ", ytest.shape)

# Fitting Multi Linear regression model to training model

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(xtrain, ytrain)

# predicting the test set results

y\_pred = regressor.predict(xtest)

plt.scatter(ytest, y\_pred, c = 'green')

plt.xlabel("Price: in $1000's")

plt.ylabel("Predicted value")

plt.title("True value vs predicted value : Linear Regression")

plt.show()

