1.Histogram Program

import matplotlib.pyplot as plt

from sklearn import datasets

iris = datasets.load\_iris()

x1= iris.data[:,0]

plt.hist(x1)

plt.xlabel("Sepeal Length")

plt.ylabel("No of Occurances")

plt.title("Histogram Demo - Iris Datasets")

plt.show()

2.Matrix operations

import numpy as np

a = np.array([1, 2, 3])

print("type: %s" %type(a))

print("shape: %s" %a.shape)

print(a[0], a[1], a[2])

a[0] = 5

print(a)

b = np.array([[1,2,3],[4,5,6]])

print("\n shape of b:",b.shape)

print(b[0, 0], b[0, 1], b[1, 0])

a = np.zeros((2,2))

print("All zeros matrix:\n %s" %a)

b = np.ones((1,2))

print("\nAll ones matrix:\n %s" %b)

d = np.eye(2)

print("\n identity matrix: \n%s"%d)

e = np.random.random((2,2))

print("\n random matrix: \n%s"%e)

#vectorized sum

print("Vectorized sum example\n")

x = np.array([[1,2],[3,4]])

print("x:\n %s" %x)

print("sum: %s"%np.sum(x))

print("sum axis = 0: %s" %np.sum(x, axis=0))

print(" sum axis = 1: %s" %np.sum(x, axis=1))

#matrix dot product

a = np.arange(10000)

b = np.arange(10000)

dp = np.dot(a,b)

print("Dot product: %s\n" %dp)

#outer product

op = np.outer(a,b)

print("\n Outer product: %s\n" %op)

#elementwise product

ep = np.multiply(a, b)

print("\n Element Wise product: %s \n" %ep)

Aim: Program to implement K-NN classification using any standard dataset available in the public

domain

Source Code

from sklearn.neighbors import KNeighborsClassifier

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

from sklearn.datasets import load\_iris

irisData = load\_iris()

X = irisData.data

Y = irisData.target

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.45, random\_state=68)

knn = KNeighborsClassifier(n\_neighbors=7)

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

print(knn.predict(X\_test))