Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Course No: CSE 4108

Course Title: Artificial Intelligence Lab

Assignment No: 04

Date of Submission: 13/2/2022

Submitted to:

Mr. Faisal Muhammad Shah
Associate Professor, Department of CSE, AUST.

Mr. Md. Siam Ansary
Lecturer, Department of CSE, AUST.

Submitted by,

Name: Atanu Kumar Saha

Student ID: 17.02.04.003

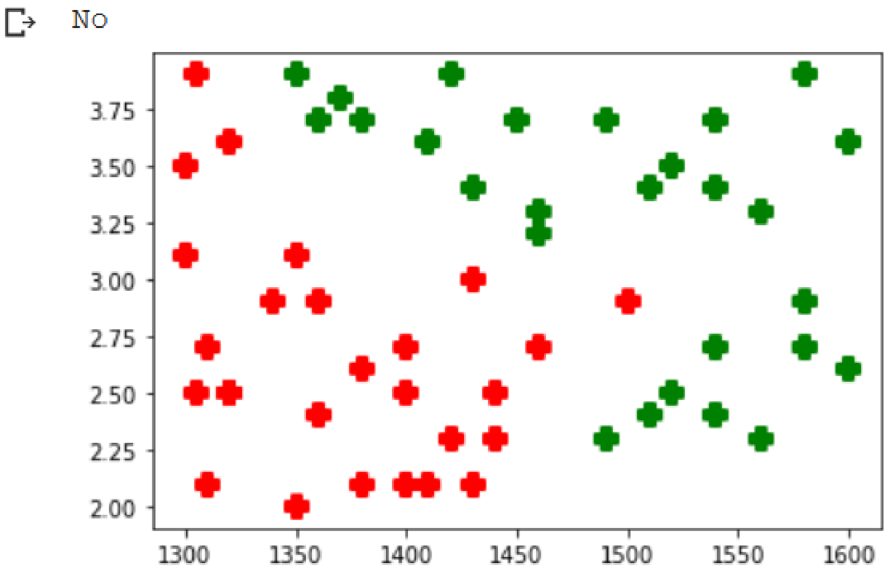
Question : Implement K Nearest Neighbor classifier in Python.

Solution:

Python Code:

```
import matplotlib
import numpy as np
import matplotlib.pyplot as plt

X = [[1580,2.9], [1540,2.7], [1600,2.6], [1580,2.7], [1520,2.5],
     [1540,2.4], [1560,2.3], [1490,2.3], [1510,2.4],
     [1350,3.9], [1360,3.7], [1370,3.8], [1380,3.7], [1410,3.6],
     [1420,3.9], [1430,3.4], [1450,3.7], [1460,3.2],
     [1580,3.9], [1540,3.7], [1600,3.6], [1490,3.7], [1520,3.5],
     [1540,3.4], [1560,3.3], [1460,3.3], [1510,3.4],
     [1340,2.9], [1360,2.4], [1320,2.5], [1380,2.6], [1400,2.1],
     [1320,2.5], [1310,2.7], [1410,2.1], [1305,2.5],
     [1460,2.7], [1500,2.9], [1300,3.5], [1320,3.6], [1400,2.7],
     [1300,3.1], [1350,3.1], [1360,2.9], [1305,3.9],
     [1430,3.0], [1440,2.3], [1440,2.5], [1380,2.1], [1430,2.1],
     [1400,2.5], [1420,2.3], [1310,2.1], [1350,2.0]]

Y =['Yes','Yes','Yes','Yes','Yes','Yes',
    'Yes','Yes','Yes',
    'Yes','Yes','Yes','Yes','Yes','Yes',
    'Yes','Yes','Yes',
    'Yes','Yes','Yes','Yes','Yes','Yes',
    'Yes','Yes','Yes',
    'No','No','No','No','No','No',
    'No','No','No',
    'No','No','No','No','No','No',
    'No','No','No',
    'No','No','No','No','No','No',
    'No','No','No']

for i in range(len(X)):
    if Y[i] == 'Yes':
        plt.scatter(X[i][0], X[i][1], s=100, marker='P',
                    linewidths=2, color='green')
    else:
        plt.scatter(X[i][0], X[i][1], s=100, marker='P',
                    linewidths=2, color='red')

pathsToNeighbor={}

def kNearestNeighbor(point,k):
    findingPathToNeighbor(point)
    sortedPathsToNeighbor = sorted(pathsToNeighbor.items())
    Yes=0
    No=0
    for i in range(k):
        if sortedPathsToNeighbor[i][1]=='No':
            Yes+=1
        else:
            No+=1

    if(Yes>No):
        return 'Yes'
    else:
        return 'No'

def findingPathToNeighbor(point):
    n_of_dimensions=len(point)
    for i in range(len(X)):
        total_d=0
        for j in range(0,n_of_dimensions):
            d=abs(point[j]-X[i][j])
            total_d +=(d * d)
        total_d=np.sqrt(total_d)
        pathsToNeighbor[total_d]=Y[i]

point=[1520,2.3]
print(kNearestNeighbor(point,5))
plt.show()
```

Output:



```
No
```

Question : Implement K-Means-Clustering in Python.

Solution:

<u>Python Code:</u>

```
from google.colab import drive
drive.mount('/content/drive')
```

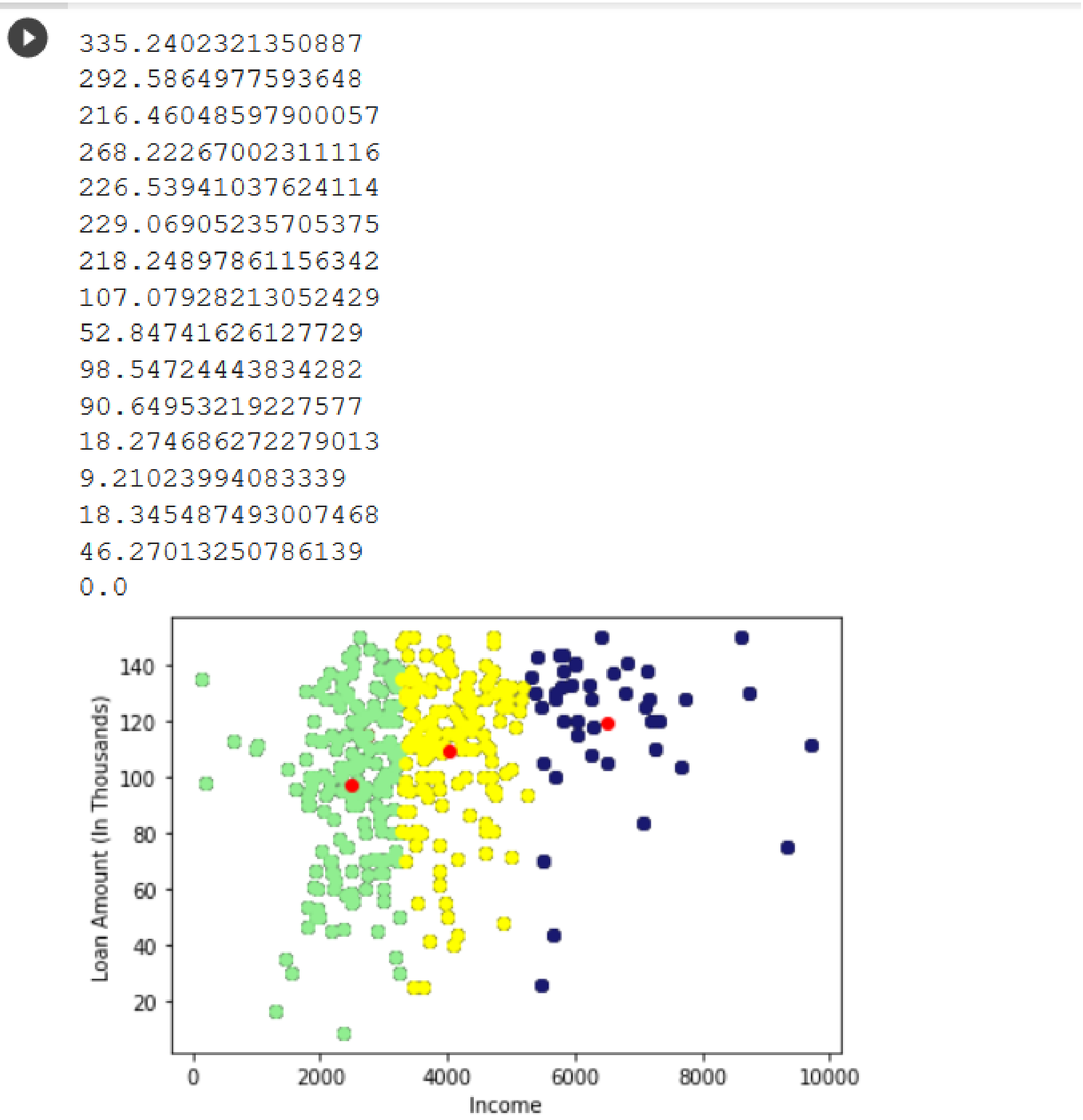
```
Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
import random as rd
import matplotlib.pyplot as plt
pd.options.mode.chained_assignment = None

data=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/input.csv')
data.head()
```

```
X = data[["LoanAmount","ApplicantIncome"]]

plt.scatter(X["ApplicantIncome"],X["LoanAmount"],c='black')
plt.xlabel('AnnualIncome')
plt.ylabel('Loan Amount (In Thousands)')

K=3

Centroids = (X.sample(n=K))
plt.scatter(X["ApplicantIncome"],X["LoanAmount"],c='black')
plt.scatter(Centroids["ApplicantIncome"],Centroids["LoanAmount"],c='red')
plt.xlabel('AnnualIncome')
plt.ylabel('Loan Amount (In Thousands)')

diff = 1
j=0

while(diff!=0):
    XD=X
    i=1
    for index1,row_c in Centroids.iterrows():
        ED=[]
        for index2,row_d in XD.iterrows():
            d1=(row_c["ApplicantIncome"]-row_d["ApplicantIncome"])**2
            d2=(row_c["LoanAmount"]-row_d["LoanAmount"])**2
            d=np.sqrt(d1+d2)
            ED.append(d)
        X[i]=ED
        i=i+1

    C=[]
    for index,row in X.iterrows():
        min_dist=row[1]
        pos=1
        for i in range(K):
            if row[i+1] < min_dist:
                min_dist = row[i+1]
                pos=i+1
        C.append(pos)
    X["Cluster"]=C
    Centroids_new = X.groupby(["Cluster"]).mean()[["LoanAmount","Applicant
Income"]]
```

```
    if j == 0:
        diff=1
        j=j+1
    else:
        diff = (Centroids_new['LoanAmount'] - Centroids['LoanAmount']).sum
() + (Centroids_new['ApplicantIncome']
                    - Centroids['ApplicantIncome']).sum()
        print(diff.sum())
    Centroids = X.groupby(["Cluster"]).mean()[["LoanAmount","ApplicantInco
me"]]

color = ['lightgreen', 'yellow', 'midnightblue']
for k in range(K):
    data = X[X["Cluster"] == k + 1]
    plt.scatter(data["ApplicantIncome"], data["LoanAmount"], c=color[k])
plt.scatter(Centroids["ApplicantIncome"], Centroids["LoanAmount"], c='red'
)
plt.xlabel('Income')
plt.ylabel('Loan Amount (In Thousands)')
plt.show()
```

Output:



```
335.2402321350887
292.5864977593648
216.46048597900057
268.22267002311116
226.53941037624114
229.06905235705375
218.24897861156342
107.07928213052429
52.84741626127729
98.54724443834282
90.64953219227577
18.274686272279013
9.21023994083339
18.345487493007468
46.27013250786139
0.0
```