Implementing K-Nearest Neighbors (KNN) .

Rakib Hossain Rifat
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
160204099@aust.edu

***Abstract*—Main objective of this experiment is implementing K-Nearest Neighbors (KNN) algorithm.**
***Index Terms*—KNN.**

I. Introduction

K-Nearest Neighbors (KNN) is used to classifay data points based on the distance of the given class from the given dataset.

II. Methodology

In this experiment I need to perform several tasks.

A. Plotting Train Data of Both Classes

For this I need to plot training data of both classes from "train.txt", and samples belonging to same class should have same marker and color.

B. Plotting Test Data of Both Classes

After implementing KNN algorithm. The value of K will be taken from user. Classify the test points from "test.txt" with different colored markers according to the predicted class label.

for classifying data set given equation will be used ,

$$(X1 - T1)^2 + (X2 - T2)^2$$

C. Print the top K distances along with their class labels

and i have to print the top K distances along with their class labels and the predicted class to "prediction.txt" for each of the test data

III. Result Analysis

A. Plotting Training Data of Both Classes

Here I have plotted data of both classes with different marker and color.
Class 1= [(7, 7), (7, 4), (6, 4), (7, 5), (7, 6), (6, 7), (6, 6)]
Class 2= [(3, 4), (2, 3), (3, 2), (4, 3), (3, 3), (4, 4), (1, 4)]

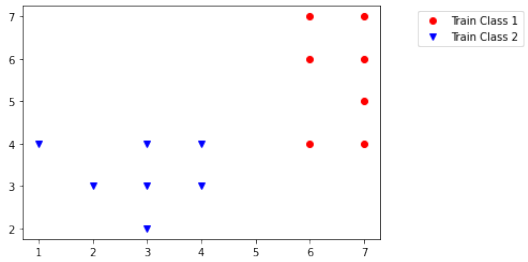

Fig. 1. Training Data of Class One and Class Two

B. Plotting Training Data of Both Classes

Here I have plotted data of both classes with different marker and color.

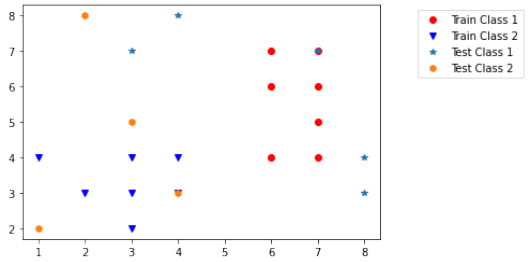

Fig. 2. Test Data of Class One and Class Two

IV. Conclusion

From the experiment it can be stated that data points can be classified using K-Nearest Neighbors (KNN) algorithm correctly.

V. Code

```
# -*- coding: utf-8 -*-
"""160204099_B2_04.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1
    ow8WuE_azifZxA4wVAKoqvmsou5oo2e6
"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import random
import math

"""Variables"""

omega_one=[]
omega_two=[]
Test_omega_one=[]
Test_omega_two=[]
pre_class=[]

print('Enter Value of K:')
K = int(input())

"""**Upload And path setup**"""

train_df=pd.read_csv("/content/train_knn.txt",sep=",
    " ,names=["x1","x2","Y"])
test_df=pd.read_csv("/content/test_knn.txt",sep=","
    ,names=["x1","x2"])
f = open("/content/prediction.txt", "a")

"""**Traning Data Separation**"""

def class_separation(df):
  for i in range(0,len(df)):
    li=[]
    if df['Y'][i]==1:
      li=df['x1'][i],df['x2'][i]
      omega_one.append(li)
    elif df['Y'][i]==2:
      li=df['x1'][i],df['x2'][i]
      omega_two.append(li)

class_separation(train_df)

"""**Plotting Traning Classes**"""

plt.plot(*zip(*omega_one),'ro',label="Train Class 1"
    )
plt.plot(*zip(*omega_two),"bv",label="Train Class 2"
    )
plt.legend(loc="upper center",bbox_to_anchor=(1.25,
    1))
plt.show()

def takefirst(elem):
    return elem[0]

def class_predicition_KNN(x,y,i):
  temp_dict={}
  c1=0
  c2=0
  f.write("Test {} \n".format(i))
  f.write("test point {} {} \n".format(x,y))
  for i in range(0,len(train_df)):

    temp_one=train_df['x1'][i]-x
    temp_two=train_df['x2'][i]-y
    temp_dist=math.pow(temp_one,2)+math.pow(temp_two
    ,2)

    temp_dict["distance {}".format(i)]=[temp_dist,
    train_df["Y"][i]]

  return temp_dict

for i in range(0,len(test_df)):

  c1=0
  c2=0
  temp_dict={}
  li=[]
  temp_dict=class_predicition_KNN(test_df["x1"][i],
    test_df["x2"][i],i)
  li=test_df["x1"][i],test_df["x2"][i]
  a=list(temp_dict.values())
  a.sort(key=takefirst)
  for i in range(0,K):


    f.write("Distance {} class {} \n".format(a[i
    ][0],a[i][1]))
    if a[i][1]==1:
      c1=c1+1
    else:
      c2=c2+1
  if c1>c2:

      f.write("Predicted Class: {} \n".format(1))

      Test_omega_one.append(li)
  else:

      f.write("Predicted Class: {} \n".format(2))
      Test_omega_two.append(li)

f.close()
plt.plot(*zip(*omega_one),'ro',label="Train Class 1"
    )
plt.plot(*zip(*omega_two),"bv",label="Train Class 2"
    )
plt.plot(*zip(*Test_omega_one),"*",label="Test Class
     1")
plt.plot(*zip(*Test_omega_two),"h",label="Test Class
     2")
plt.legend(loc="upper center",bbox_to_anchor=(1.25,
    1))
plt.show()
```