K-Nearest Neighbour Algorithm

Problem Statement: To predict the weight using KNN algorithm without using any packages.

Formula: Here we are using the Euclidian distance formula to calculate the distance between two points. The distance between the two points (x1,y1) and (x2,y2) is given by the formula:

$$[(x2-x1)2+(y2-y1)2]1/2$$

Algorithm:

Step 1: Start

Step 2: Load the train data

Step 3: Load the test data

Step 4: Assign K values

Step 5: Assign target variable

Step 6: Create the variable to store the predicted target values

Step 7: Repeat through the steps:

Find the difference matrix

Compute the distance using Euclidian distance formula

Sort the train data in ascending order w.r.t the distances

Compute the average of the K terms of train dataset and append to predicted targeted values.

Step 8: Display the predicted target values

Step 8: Stop

Code: # -*- coding: utf-8 -*-

“””

@Script-author: Salman

@Script-description: To predict the value using KNN algorithm

@Script-start date: 08/01/2020

@Script-last updated: 13/01/2020

“””

#Set the train and test data

```
train=[[13,14,16],[12,17,14],[11,15,18]]
test=[13,19,17]
diff=[]
```

#Compute the difference matrix

```
for i in range(len(train)):
    im=[]
    for j in range(len(test)):
        im.append(test[j]-train[i][j])
        diff.append(im)
dist=[]
```

#Computing the distance using the formula of Euclidian distance

```
for i in range(len(train)):
    sum=0
    for j in range(len(test)):
        sum=sum+diff[i][j]**2
        dist.append(sum)
```

#Creating a dictionary to link the train data and the calculated distance

```
dict1={}
for i in range(len(dist)):
    dict1[dist[i]]=train[i]
```

```
 #Sorting based on distance
dict1=sorted(dict1.items())
dict1
#Using the K values estimating the predicted value
predict,sum=[],0
for i in range(len(dict1)):
    sum=sum+dict1[i][1][2]
    predict.append(sum/(i+1))
predict
#Estimating the error
error=[]
for i in range(len(predict)):
    error.append((test[2]-predict[i])*100/test[2])
error
#Based on the least error the predicted value is estimated
print("Accurate value is ",predict[error.index(min(error))])
```

OUTPUT:

```
Accurate value is 16.0
```