K-Nearest Neighbour Algorithm

Problem Statement: To predict the weight using KNN algorithm.

Formulas used: Euclidian distance formula has been used to calculate the distance.

Algorithm:

```
Start
Load the train data
Load the test data
Assign k values
Assign target variable
Create the variable to store the predicted targeted values
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 average of the first k terms of train dataset
        Append to predicted targeted values.
Display the predicted targeted values
Stop
```

Code:

```
@script-author:beril thampi
@script-description:To predict the value using knn algorithm

#setting train and test data
train=[[1,2,3],[4,5,6],[7,8,9]]
test=[[9,10,11]]
diff=[]
x,y1,y2=[],[],[]

#Computing the difference matrix
n=len(train[0])-1
for i in range (len(train)):
    x=[]
    for j in range (n):
        x.append(test[0][j]-train[i][j])
    y1.append(x)
print("\nDifference Matrix ",y1)

#Computing distance using euclidian formula
for i in range (len(y1)):
    s=0
    for j in range(len(y1[0])):
        s=s+y1[i][j]**2
    y2.append(s**0.5)
print("\nThe Distance matrix is ",y2)

y3={}
for i in range (len(y2)):
    y3[y2[i]]=train[i]
print("\nThe Combined matrix is ",y3)

#sorting based on distance
y3=sorted(y3.items())
print("\nThe Sorted matrix is ",y3)

predict=[]
avg_sum=0
#Using the k values estimating the predicted value
for i in range (len(y2)):
    avg_sum=avg_sum+y3[i][1][2]
    avg=avg_sum/(i+1)
    predict.append(avg)
print("\nPredicted values are ",predict)
```

```
#Estimaing the error
percent=[]
for i in range (len(predict)):
    percent.append((test[0][n]-predict[i])*100/(test[0][n]))
print("\nPercentage error : ",percent)
print("\nMost accurate value : ",predict[percent.index(min(percent))])
```

Output:

```
Difference Matrix  [[9, 9], [6, 6], [3, 3]]

The Distance matrix is  [12.727922061357855, 8.48528137423857, 4.242640687119285]

The Combined matrix is  {12.727922061357855: [1, 2, 3], 8.48528137423857: [4, 5, 6],
4.242640687119285: [7, 8, 9]}

The Sorted matrix is  [(4.242640687119285, [7, 8, 9]), (8.48528137423857, [4, 5, 6]),
(12.727922061357855, [1, 2, 3])]

Predicted values are  [9.0, 7.5, 6.0]

Percentage error :  [25.0, 37.5, 50.0]

Most accurate value :  9.0
```