K-Nearest Neighbour Algorithm

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

Formulas used: Euclidian distance formula-The distance two points (x1,y1) and (x2,y2) is given by the formula

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

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:

```
# -*- coding: utf-8 -*-
"""

@script-author:arpitha shibu
```

```
@script-description:To predict the value using knn algorithm
@script-start date:07.01.20
@script-last updated:11.01.20


"""
#setting train and test data
train=[[1,2,3],[2,3,4],[4,5,6]]
test=[3,5,7]
diff=[]
#Computing 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 distance using euclidian formula
for i in range(len(train)):
    s=0
    for j in range(len(test)):
        s+=diff[i][j]**2
    dist.append(s)

dict1={} # creating a dictionary to link the train data and the distance calcuated
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,s=[],0
for i in range(len(dict1)):
    s+=dict1[i][1][2]
    predict.append(s/(i+1))
predict
#Estimaing the error
error=[]
for i in range(len(predict)):
    error.append((test[2]-
predict[i])*100/test[2])
error
#based on the least error estimating the
predicted value
print("Accurate value is
",predict[error.index(min(error))])
```

Output:

```
Accurate value is  6.0
```