K-Nearest Neighbour Algorithm

It is also called a lazy learner algorithm, it works on the principal of similar things exist in proximity. It is a simple, supervised machine learning algorithm.

Problem Statement:​ ​To predict the weight using K nearest algorithm without the usage of any packages.

Formulas used:​

There are various methods to calculate distances, one of it is Euclidean distance, it is the distance of two points in a Euclidean space. There is also the manhattan distance, supremum distance etc

Euclidean 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:

Step1: Start

Step 2: Load the train data

Step 3:Load the test data

Step 4:Assign k values which is the number of neighbours

Step 5:Assign target variable

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

Step 7:Repeat through the steps:

Step 8: Find the difference matrix

Step 9: Compute the distance using Euclidean distance formula

Step 10: Sort the train data in ascending order w.r.t the distances

Step 11: Display the predicted targeted values

Step 12: Stop

Code:

#setting train and test data

train=[[9,23,10],[4,13,13],[13,14,16]]

test=[13,19,17]

diff=[]

#Computing the difference matrix

for i in range(len(train)):

l=[]

for j in range(len(test)):

l.append(test[j]-train[i][j])

diff.append(l)

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)

dict={}

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

for i in range(len(dist)):

dict[dist[i]]=train[i]

#sorting based on distance

dict=sorted(dict.items())

dict

#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: predicted value : 11.50