Experiment No:-1 Date:- 09/01/2025

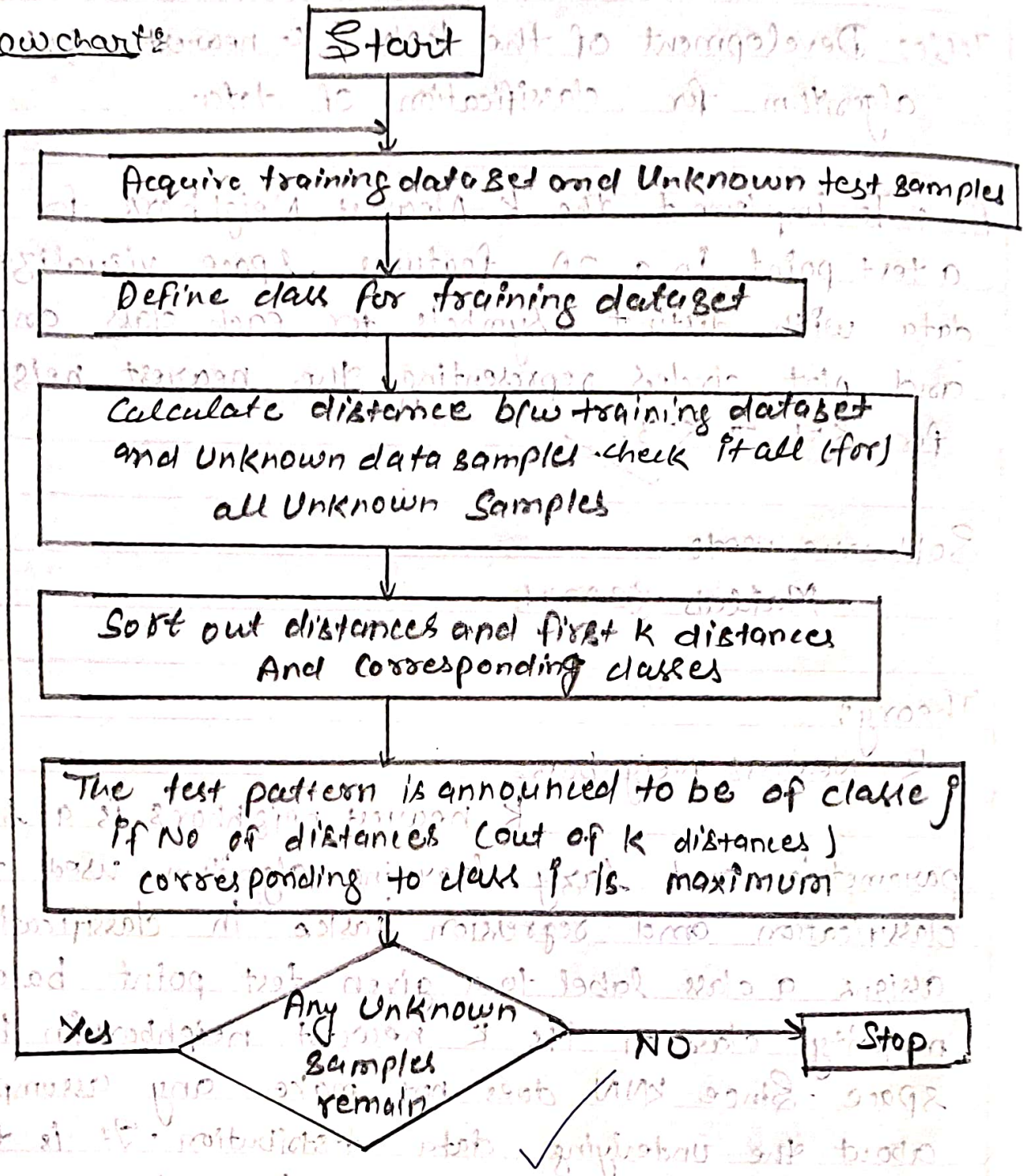
Title: Development of the KNN (K-nearest neighbor) algorithm for classification of data.

Aim: To implement the K-Nearest Neighbors for classify a test point in a 2D features space, visualize the data with distinct symbols for each class and test points and plot circles representing the nearest neighbors for K=1 & K=3.

Software used:
Matlab R2024b.

Theory:
K-Nearest Neighbors:
K nearest neighbors is a simple, non parametric, and lazy learning algorithm used for classification and regression tasks. In classification, KNN assigns a class label to a given test point based on the majority class of its K nearest neighbors in the feature space. Since KNN does not make any assumptions about the underlying data distribution. It is highly versatile and widely used in pattern recognition data mining and machine learning.

Flowchart:


Start
Acquire training dataset and Unknown test samples
Define class for training dataset
Calculate distance b/w training dataset and Unknown data samples check it all (for) all Unknown Samples
Sort out distances and first k distances And corresponding classes
The test pattern is announced to be of class if No of distances (out of k distances) corresponding to class i is maximum
Any Unknown Samples remain
Yes
NO
Stop

Algorithm:

① Input: The dataset (x_i, y_i, c_i) where the (x_i, y_i) is the points in the 2D plane and c_i is the labels class to that specific points.

⇒ for Each points i in the dataset:

→ Compute the Encludian distance dataset and test point.

where $d(x,p) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$

where (x_i, y_i) datapoint (x, y) is testpoint

⇒ Sort the distance in ascending Order.

⇒ Select the top K nearest neighbors.

⇒ Count the class labels of the selected neighbors.

⇒ Assign the class the highest count to the test Sample

⇒ Output the predicted class for the test sample.

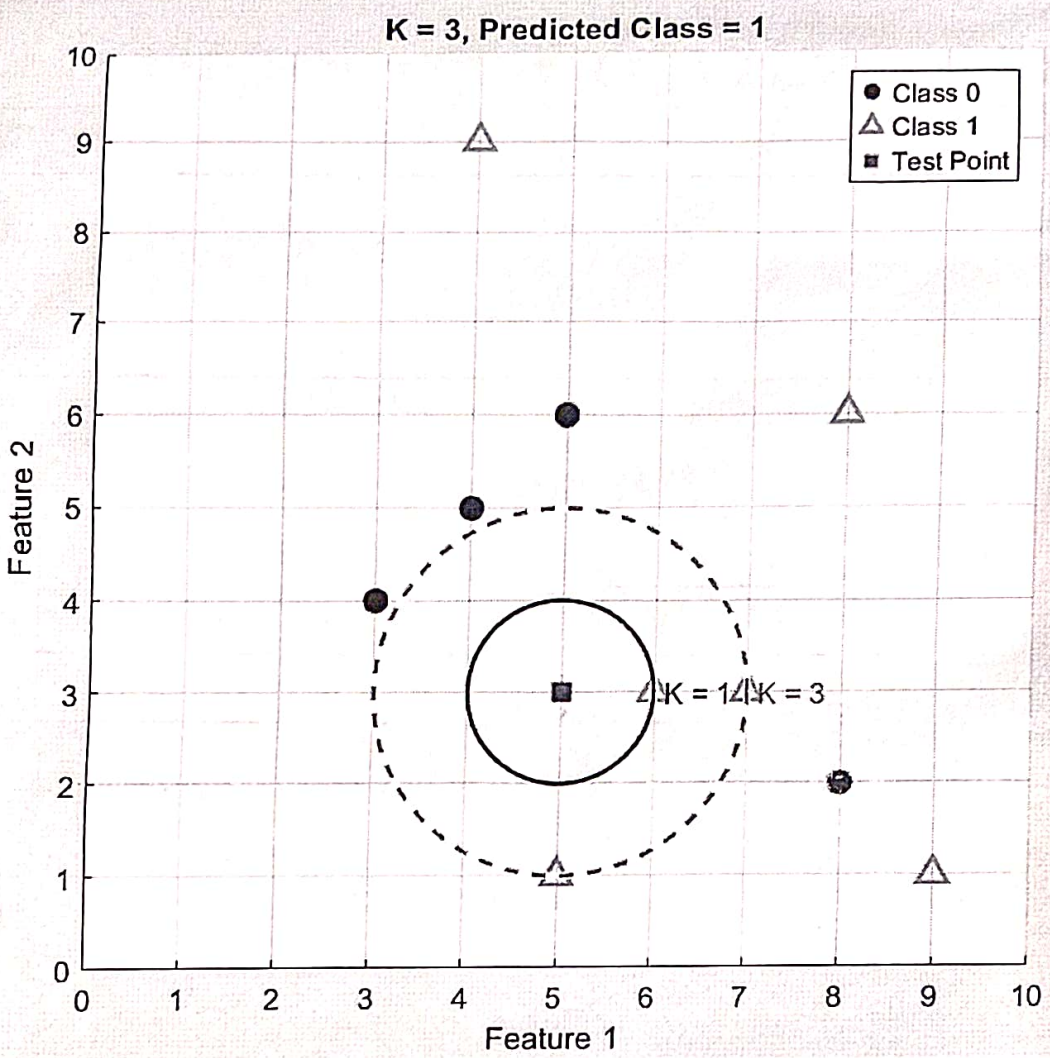
Result: testpoint: (5,3)

database → first class − 0
second class − 1

Datapoint	(3,4)	(4,5)	(5,6)	(6,3)	(5,1)	(8,2)	(7,3)	(4,9)	(9,1)	(8,6)
class	~~2.230~~ 0	0	0	1	1	0	1	1	1	1
distance	2.23	2.23	3	1	2	3.1	2	6.08	4.47	4.24

K=3

So closed 3 neighbor → (6,3), (5,1), (7,3)



Point	Class	Distance
(6, 3)	1	1.00
(5, 1)	1	2.00
(7, 3)	1	2.00
(3, 4)	0	2.24
(4, 5)	0	2.24
(5, 6)	0	3.00
(8, 2)	0	3.16
(8, 6)	1	4.24
(9, 1)	1	4.47
(4, 9)	1	6.08

Visual representation of dataset with the K values.

The final class for testpoint (5,3) is Class -1)

Qualitative Result:

⇒ Visualization of Datapoint:

→ Class 0 datapoint are represented by circle

→ class 1 data point are represented by triangle

→ The test point is represented by a square

→ Circle around the test point are drawn for K=1 and K=3 indicating the distances to the nearest neighbors. and the 3rd nearest neighbor, repectively.

⇒ Classification of the test point:

→ The test point (5,3) lies with the vicinity of data points for both classes.

→ Since the majority of the classes (nearest neighbors) belong to class 1, the algorithm correctly predict the test point as belonging to class 1.

Discussions:-

⇒ for K=1, the nearest neighbor from, class 1, resulting in an Immediate classification as class 1.

→ for K=3, threeneighbors from class 1, leading to a stable classification as class 1 by majority voting.

⇒ Choose an optimal K is crucial, as a very small K can lead to noisy predictions and a very large K can result in bias by including distant points.

=> KNN requires calculating the distances to all points, making it computationally expensive for large datasets.
=> The algorithm's performance can degrade in high-dimensional spaces due to the curse of dimensionality.

Conclusion:

The KNN algorithm was applied to classify a test point based on a dataset with two classes. The experiment The experiment showed that for K=3, the test point (5,3) was correctly predicted as class 1. Different symbols and circles for K=1 & K=3 provided a clear visual understanding of the classification. The results confirmed KNN as an effective method, where proper selection of K is crucial for accuracy.