MODULE-2
Part3
(KNN)

K-Nearest Neighbour Algorithm(K-NN)

- The k-NN algorithm is among the simplest of all machine learning algorithms
- Supervised learning
- KNN is a non-parametric, lazy learning algorithm
  - Non-parametric: KNN is non-parametric because it doesn’t assume a functional form or fixed number of parameters — instead, it learns from the data “as is” and defers decision making until prediction time.
  - Lazy: it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal.
  - In contrast to so called “eager learning” algorithms (which carries out learning without

Why KNN is non-parametric:

1. No Assumptions About Data Distribution:

- KNN does not assume any specific form for the function that maps inputs to outputs.
- Many parametric algorithms, such as linear regression or logistic regression, assume a certain type of relationship (e.g., linear) between the input features and the target variable, and they try to learn the parameters of this assumed relationship.

2. Infinite Parameters: The number of parameters grows with the size of the training data. Each training example can be considered a parameter that influences the prediction. As new data points are added, the model complexity increases. Parametric models, on the other hand, have a fixed number of parameters that do not change regardless of the size of the training dataset.

3. Flexibility and Adaptability:

- Because KNN does not assume a specific form for the data, it is very flexible and can adapt to a wide variety of data shapes and patterns.
- This flexibility makes KNN a powerful tool for complex and non-linear relationships, as it uses the actual data points for making predictions.

4. Memory-Based Learning:

- KNN is a type of memory-based (or instance-based) learning algorithm, where all training data is stored and used for prediction. The model does not summarize the data into a fixed number of parameters.
- Predictions are made based on the entire dataset (or a subset of it), which means the "model" effectively consists of the training data itself.

5. No Training Phase:

- KNN does not involve a training phase where a model is fitted to the data. Instead, it memorizes the training instances and uses them directly to make predictions.
- Parametric models typically involve a training phase where the parameters are estimated based on the training data.

- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

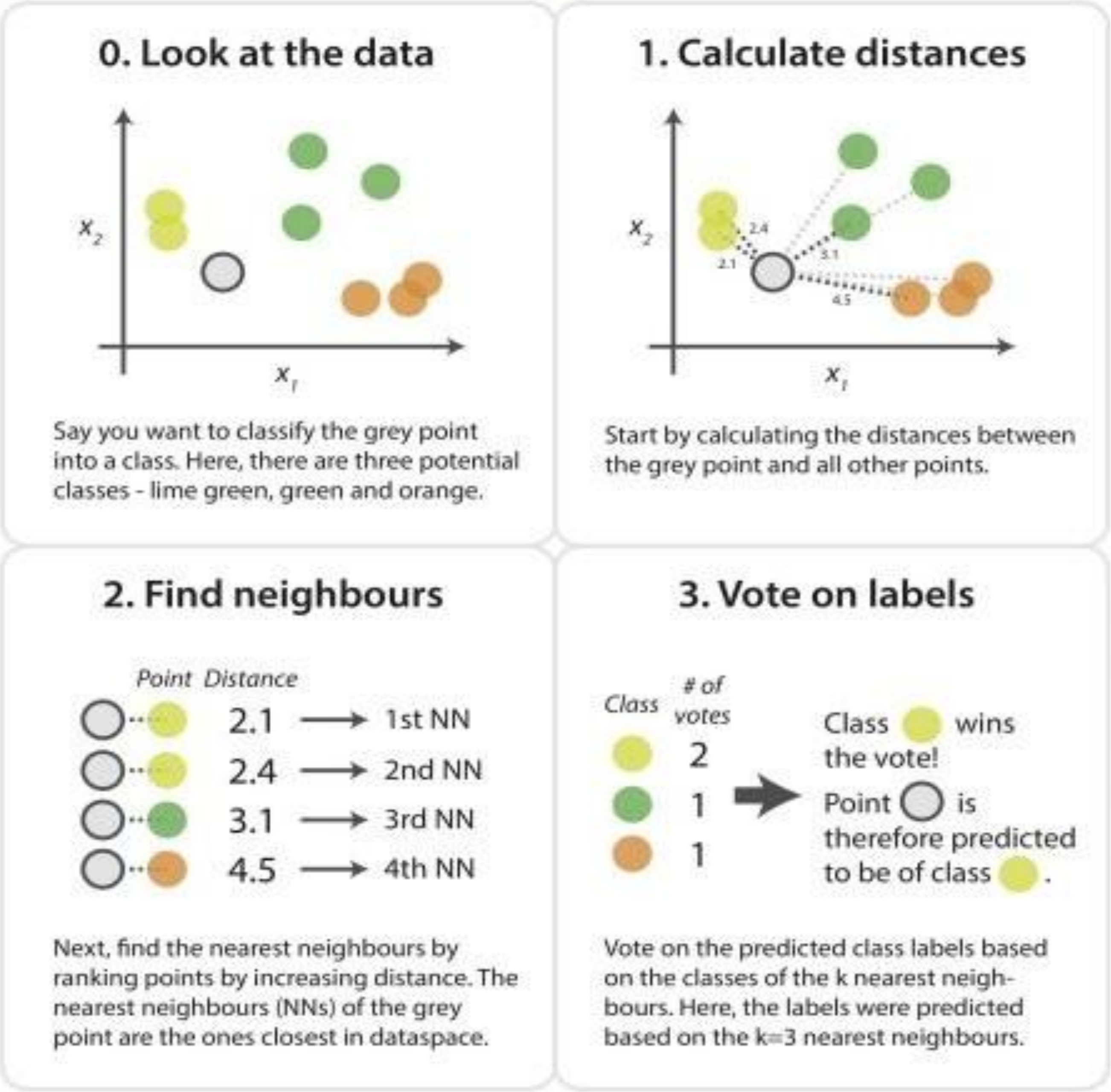
How Does the K-Nearest Neighbors Algorithm Work?

- The K-NN algorithm compares a new data entry to the values in a given data set (with different classes or categories).
- Based on its closeness or similarities in a given range (**K**) of neighbors, the algorithm assigns the new data to a class or category in the data set (training data).

Steps:

- **Step #1 -** Assign a value to **K**.
- **Step #2 -** Calculate the distance between the new data entry and all other existing data entries (you'll learn how to do this shortly). Arrange them in ascending order.
- **Step #3 -** Find the **K** nearest neighbors to the new entry based on the calculated distances.
- **Step #4 -** Assign the new data entry to the majority class in the nearest neighbors.

kNN Algorithm



Algorithm

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Distance functions

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

When to Consider Nearest Neighbour ?

Lots of training data and no training stage, all the work is done during the test stage

Advantages:

- Can be applied to the data from any distribution. For example, data does not have to be separable with a linear boundary
- Training is very fast
- Learn complex target functions
- Don’t lose information

Disadvantages:

- Choosing k may be tricky
- Slow at query time
- Easily fooled by irrelevant attributes
- Need a large number of samples for accuracy

K-NN Summary

- Advantages
  - Learning is extremely simple and intuitive
  - Very flexible decision boundaries
  - Variable-sized hypothesis space
- Disadvantages
  - distance function must be carefully chosen or tuned
  - irrelevant or correlated features have high impact and must be eliminated
  - typically cannot handle high dimensionality computational
  - computational costs: memory and classification-time computation

Strengths	Weaknesses
• Simple and effective • Makes no assumptions about the underlying data distribution • Fast training phase	• Does not produce a model, limiting the ability to understand how the features are related to the class • Requires selection of an appropriate *k* • Slow classification phase • Nominal features and missing data require additional processing

1. Consider the following datasets with attributes such as Sepal Length, sepal width, and class labels as Species. Suppose that a new observation is added with Sepal Length, Sepal Width (5.2,2.1) Find the class label of the new data point. Assume K=5.

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

Step 1: Find Distance

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(x-a)^2 + (y-b)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(5.2-5.3)^2 + (3.1-3.7)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = 0.608$$

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608

Step2:Find the distance of all the data points

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608
5.1	3.8	Setosa	0.707
7.2	3.0	Virginica	2.002
5.4	3.4	Setosa	0.36
5.1	3.3	Setosa	0.22
5.4	3.9	Setosa	0.82
7.4	2.8	Virginica	2.22
6.1	2.8	Verscicolor	0.94
7.3	2.9	Virginica	2.1
6.0	2.7	Verscicolor	0.89
5.8	2.8	Virginica	0.67
6.3	2.3	Verscicolor	1.36
5.1	2.5	Verscicolor	0.60
6.3	2.5	Verscicolor	1.25
5.5	2.4	Verscicolor	0.75

Step 3: Rank the data according to the increasing order of distance

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

K=1(setosa)
K=2(setosa,setosa)
K=3,setosa
K=4(3 setosa,1 versicolor)
K=5(3 setosa,1 versicolor, 1 viginica)

K=3/5
New data point will be under
Class Setosa

2. Consider the following datasets with attributes positions and class labels are C1,C2, and C3. Suppose that a new observation is added with position (5,7). Find the class label of the new data point. Assume K=4.

POINTS	POSITIONS (x,y)	CLASS LABELS
1	(2,11)	C1
2	(2,7)	C2
3	(11,11)	C3
4	(6,2)	C2
5	(6,9)	C1
6	(1,3)	C1
7	(5,11)	C2
8	(4,11)	C2
9	(10,12)	C3
10	(7,5)	C1
11	(9,11)	C3
12	(4,6)	C1
13	(3,6)	C2
14	(3,8)	C3
15	(3,10)	C2
16	(4,12)	C2
New Datapoint (a,b) (5,7)-----predict the class label?		

Step 1: Assume K=4

Step 2: Compute the Euclidean distance of new data(a, b) to each data point(x,y) in the dataset.

POINTS	POSITIONS (x,y)	CLASS LABELS	Distance Measure, $d=\sqrt{(a-x)^2+(b-y)^2}$	
1	(2,11)	C1	$\sqrt{(5-2)^2+(7-11)^2}=$	$\sqrt{(3)^2+(4)^2}=5.00$
2	(2,7)	C2	$\sqrt{(5-2)^2+(7-7)^2}=$	$\sqrt{(3)^2+(0)^2}=3.00$
3	(11,11)	C3	$\sqrt{(5-11)^2+(7-11)^2}=$	$\sqrt{(6)^2+(5)^2}=7.81$
4	(6,2)	C2	$\sqrt{(5-6)^2+(7-2)^2}=$	$\sqrt{(1)^2+(5)^2}=5.10$
5	(6,9)	C1	$\sqrt{(5-6)^2+(7-9)^2}=$	$\sqrt{(1)^2+(2)^2}=2.24$
6	(1,3)	C1	$\sqrt{(5-1)^2+(7-3)^2}=$	$\sqrt{(4)^2+(4)^2}=5.66$
7	(5,11)	C2	$\sqrt{(5-5)^2+(7-11)^2}=$	$\sqrt{(0)^2+(4)^2}=4.00$
8	(4,11)	C2	$\sqrt{(5-4)^2+(7-11)^2}=$	$\sqrt{(1)^2+(4)^2}=4.12$
9	(10,12)	C3	$\sqrt{(5-10)^2+(7-12)^2}=$	$\sqrt{(5)^2+(5)^2}=7.07$
10	(7,5)	C1	$\sqrt{(5-7)^2+(7-5)^2}=$	$\sqrt{(2)^2+(2)^2}=2.83$
11	(9,11)	C3	$\sqrt{(5-9)^2+(7-11)^2}=$	$\sqrt{(4)^2+(4)^2}=5.66$
12	(4,6)	C1	$\sqrt{(5-4)^2+(7-6)^2}=$	$\sqrt{(1)^2+(1)^2}=1.41$
13	(3,6)	C2	$\sqrt{(5-3)^2+(7-6)^2}=$	$\sqrt{(2)^2+(1)^2}=2.24$
14	(3,8)	C3	$\sqrt{(5-3)^2+(7-8)^2}=$	$\sqrt{(2)^2+(1)^2}=2.24$
15	(3,10)	C2	$\sqrt{(5-3)^2+(7-10)^2}=$	$\sqrt{(2)^2+(3)^2}=3.61$
16	(4,12)	C2	$\sqrt{(5-4)^2+(7-12)^2}=$	$\sqrt{(1)^2+(5)^2}=5.10$
New Datapoint (a,b) (5,7)-----predict the class label?				

Step 3: After finding the distance of each point in the dataset to P, we will sort the above points according to their distance from P (5, 7). After sorting, we get the following table.

POINTS	POSITIONS (x,y)	CLASS LABELS	Distance Measure, $d=\sqrt{(a-x)^2+(b-y)^2}$	Distance
12	(4,6)	C1	$\sqrt{(5-4)^2+(7-6)^2} =$	$\sqrt{(1)^2+(1)^2} =1.41$
13	(3,6)	C2	$\sqrt{(5-3)^2+(7-6)^2} =$	$\sqrt{(2)^2+(1)^2} =2.24$
14	(3,8)	C3	$\sqrt{(5-3)^2+(7-8)^2} =$	$\sqrt{(2)^2+(1)^2} =2.24$
5	(6,9)	C1	$\sqrt{(5-6)^2+(7-9)^2} =$	$\sqrt{(1)^2+(2)^2} =2.24$
10	(7,5)	C1	$\sqrt{(5-7)^2+(7-5)^2} =$	$\sqrt{(2)^2+(2)^2} =2.83$
2	(2,7)	C2	$\sqrt{(5-2)^2+(7-7)^2} =$	$\sqrt{(3)^2+(0)^2} =3.00$
15	(3,10)	C2	$\sqrt{(5-3)^2+(7-10)^2} =$	$\sqrt{(2)^2+(3)^2} =3.61$
7	(5,11)	C2	$\sqrt{(5-5)^2+(7-11)^2} =$	$\sqrt{(0)^2+(4)^2} =4.00$
8	(4,11)	C2	$\sqrt{(5-4)^2+(7-11)^2} =$	$\sqrt{(1)^2+(4)^2} =4.12$
1	(2,11)	C1	$\sqrt{(5-2)^2+(7-11)^2} =$	$\sqrt{(3)^2+(4)^2} =5.00$
16	(4,12)	C2	$\sqrt{(5-4)^2+(7-12)^2} =$	$\sqrt{(1)^2+(5)^2} =5.10$
4	(6,2)	C2	$\sqrt{(5-6)^2+(7-2)^2} =$	$\sqrt{(1)^2+(5)^2} =5.10$
11	(9,11)	C3	$\sqrt{(5-9)^2+(7-11)^2} =$	$\sqrt{(4)^2+(4)^2} =5.66$
6	(1,3)	C1	$\sqrt{(5-1)^2+(7-3)^2} =$	$\sqrt{(4)^2+(4)^2} =5.66$
9	(10,12)	C3	$\sqrt{(5-10)^2+(7-12)^2} =$	$\sqrt{(5)^2+(5)^2} =7.07$
3	(11,11)	C3	$\sqrt{(5-11)^2+(7-11)^2} =$	$\sqrt{(6)^2+(5)^2} =7.81$

Step 4 After sorting, then we rank the data based on the distance value

POINTS	POSITIONS (x,y)	CLASS LABELS	Distance Measure, $d=\sqrt{(a-x)^2+(b-y)^2}$	Distance	Rank
12	(4,6)	C1	$\sqrt{(5-4)^2+(7-6)^2}$ =	$\sqrt{(1)^2+(1)^2}$ =1.41	1
13	(3,6)	C2	$\sqrt{(5-3)^2+(7-6)^2}$ =	$\sqrt{(2)^2+(1)^2}$ =2.24	2
14	(3,8)	C3	$\sqrt{(5-3)^2+(7-8)^2}$ =	$\sqrt{(2)^2+(1)^2}$ =2.24	3
5	(6,9)	C1	$\sqrt{(5-6)^2+(7-9)^2}$ =	$\sqrt{(1)^2+(2)^2}$ =2.24	4
10	(7,5)	C1	$\sqrt{(5-7)^2+(7-5)^2}$ =	$\sqrt{(2)^2+(2)^2}$ =2.83	5
2	(2,7)	C2	$\sqrt{(5-2)^2+(7-7)^2}$ =	$\sqrt{(3)^2+(0)^2}$ =3.00	6
15	(3,10)	C2	$\sqrt{(5-3)^2+(7-10)^2}$ =	$\sqrt{(2)^2+(3)^2}$ =3.61	7
7	(5,11)	C2	$\sqrt{(5-5)^2+(7-11)^2}$ =	$\sqrt{(0)^2+(4)^2}$ =4.00	8
8	(4,11)	C2	$\sqrt{(5-4)^2+(7-11)^2}$ =	$\sqrt{(1)^2+(4)^2}$ =4.12	9
1	(2,11)	C1	$\sqrt{(5-2)^2+(7-11)^2}$ =	$\sqrt{(3)^2+(4)^2}$ =5.00	10
16	(4,12)	C2	$\sqrt{(5-4)^2+(7-12)^2}$ =	$\sqrt{(1)^2+(5)^2}$ =5.10	12
4	(6,2)	C2	$\sqrt{(5-6)^2+(7-2)^2}$ =	$\sqrt{(1)^2+(5)^2}$ =5.10	11
11	(9,11)	C3	$\sqrt{(5-9)^2+(7-11)^2}$ =	$\sqrt{(4)^2+(4)^2}$ =5.66	13
6	(1,3)	C1	$\sqrt{(5-1)^2+(7-3)^2}$ =	$\sqrt{(4)^2+(4)^2}$ =5.66	14
9	(10,12)	C3	$\sqrt{(5-10)^2+(7-12)^2}$ =	$\sqrt{(5)^2+(5)^2}$ =7.07	15
3	(11,11)	C3	$\sqrt{(5-11)^2+(7-11)^2}$ =	$\sqrt{(6)^2+(5)^2}$ =7.81	16

New Datapoint (a,b) (5,7)-----predict the class label?

Step 5:As we have taken k=4, we will now consider the class labels of three points in the dataset nearest to new data point (a,b)

POINTS	POSITIONS (x,y)	CLASS LABELS	Distance Measure, $d=\sqrt{(a-x)^2+(b-y)^2}$	Distance	Rank
12	(4,6)	C1	$\sqrt{(5-4)^2+(7-6)^2} =$	$\sqrt{(1)^2+(1)^2} = 1.41$	1
13	(3,6)	C2	$\sqrt{(5-3)^2+(7-6)^2} =$	$\sqrt{(2)^2+(1)^2} = 2.24$	2
14	(3,8)	C3	$\sqrt{(5-3)^2+(7-8)^2} =$	$\sqrt{(2)^2+(1)^2} = 2.24$	3
5	(6,9)	C1	$\sqrt{(5-6)^2+(7-9)^2} =$	$\sqrt{(1)^2+(2)^2} = 2.24$	4

- In the above table, points 12, 13, 14, and 5 are the closest four neighbors to the new data point (a, b).
- Therefore, we will use the class labels of points 12, 13, 14, and 5 to determine the class label for the new data point.
- The class labels for points 12, 13, 14, and 5 are C1, C2, C3, and C1, respectively.
- Among these points, the majority class label is C1. Hence, we will assign the class label of point P = (5, 7) as C1.

You are given a dataset with the following 6 training points, each having two features (X1, X2) and a class label (Y): A new test point T = (4, 3) is to be classified using KNN with Euclidean distance.

Point	X1	X2	Y
A	1	2	0
B	2	3	1
C	3	1	1
D	6	5	0
E	7	7	1
F	8	6	0

(i) Euclidean distance

Formula:

$$d(T,P) = \sqrt{(4-x_1)^2 + (3-x_2)^2}$$

- $d(T,A) = \sqrt{(4-1)^2 + (3-2)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.162$
- $d(T,B) = \sqrt{(4-2)^2 + (3-3)^2} = \sqrt{4+0} = 2.000$
- $d(T,C) = \sqrt{(4-3)^2 + (3-1)^2} = \sqrt{1+4} = \sqrt{5} \approx 2.236$
- $d(T,D) = \sqrt{(4-6)^2 + (3-5)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.828$
- $d(T,E) = \sqrt{(4-7)^2 + (3-7)^2} = \sqrt{9+16} = \sqrt{25} = 5.000$
- $d(T,F) = \sqrt{(4-8)^2 + (3-6)^2} = \sqrt{16+9} = \sqrt{25} = 5.000$

Sorted by distance:

1. B → 2.000 → Class 1
2. C → 2.236 → Class 1
3. D → 2.828 → Class 0
4. A → 3.162 → Class 0
5. E → 5.000 → Class 1
6. F → 5.000 → Class 0

(ii) KNN with $k = 3$

Neighbors: B(1), C(1), D(0)

- Votes: Class 1 → 2, Class 0 → 1
  ✅ Predicted class = **1** (no tie)

(iii) KNN with $k = 5$

Neighbors: B(1), C(1), D(0), A(0), E(1)

- Votes: Class 1 → 3, Class 0 → 2
  ✅ Predicted class = **1** (no tie)

Compute the Minkowiski distance (p=2.6) from point T to all six training points. K=3

The **Minkowski distance** between two points

$$P = (x_1, x_2, \ldots, x_n), \quad Q = (y_1, y_2, \ldots, y_n)$$

with parameter p is defined as:

$$D_p(P, Q) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

✅ Final Minkowski (p=2.6) Distances:

Point	Class	Distance
B	1	**2.000**
C	1	**2.121**
D	0	**2.611**
A	0	**3.065**
E	1	**4.643**
F	0	**4.643**

Class1

Case where tie occurs

Point	(X_1, X_2)	Class
A	(1,2)	0
B	(2,3)	1
C	(3,1)	1
D	(6,5)	0
E	(7,7)	1
F	(8,6)	0

Test point $T = (4, 3)$. Use Minkowski distance with $p = 2.6$ (the distances were computed earlier):

Point	Distance (p=2.6)	Class
B	2.000000	1
C	2.120954	1
D	2.611023	0
A	3.065180	0
E	4.642909	1
F	4.642909	0

K=4 so tie between class 1 and 0

1) Nearest 4 neighbors (sorted by distance)

The 4 nearest neighbors are:

- B — distance $d_B = 2.000000$, class = 1
- C — distance $d_C = 2.120954$, class = 1
- D — distance $d_D = 2.611023$, class = 0
- A — distance $d_A = 3.065180$, class = 0

2) Ordinary majority voting (tie)

- Class 1 votes: B, C → **2 votes**
- Class 0 votes: D, A → **2 votes**

  So there is a tie (2 vs 2).

3) Distance-weighted voting (tie-breaker)

Use weight for neighbor i: $w_i = \dfrac{1}{d_i^2}$.

Compute each weight (rounded sensibly; I'll keep 4 decimal places):

- $w_B = 1/(2.000000)^2 = 1/4 = \mathbf{0.2500}$
- $w_C = 1/(2.120954)^2 \approx 1/4.498446 \approx \mathbf{0.2223}$
- $w_D = 1/(2.611023)^2 \approx 1/6.817442 \approx \mathbf{0.1467}$
- $w_A = 1/(3.065180)^2 \approx 1/9.395328 \approx \mathbf{0.1064}$

(Notes: I computed each squared distance carefully, then the reciprocal. Small rounding shown.)

Now sum weights by class:

- **Class 1 total weight** = $w_B + w_C = 0.2500 + 0.2223 = \mathbf{0.4723}$
- **Class 0 total weight** = $w_D + w_A = 0.1467 + 0.1064 = \mathbf{0.2531}$

4) Decision

Weighted totals: Class 1 → **0.4723**, Class 0 → **0.2531**.

Thus, under distance-weighted voting, **Class 1 wins**.

Manhattan distance

Point	X1	X2	Y
A	1	2	0
B	2	3	1
C	3	1	1
D	6	5	0
E	7	7	1
F	8	6	0

T=(5,4)

Step 1: Compute Manhattan distances

Formula:

$$d(P,T) = |x_1 - t_1| + |x_2 - t_2|$$

- d(A,T) = |1–5| + |2–4| = 4 + 2 = **6**
- d(B,T) = |2–5| + |3–4| = 3 + 1 = **4**
- d(C,T) = |3–5| + |1–4| = 2 + 3 = **5**
- d(D,T) = |6–5| + |5–4| = 1 + 1 = **2**
- d(E,T) = |7–5| + |7–4| = 2 + 3 = **5**
- d(F,T) = |8–5| + |6–4| = 3 + 2 = **5**

Step 2: Sort neighbors (smallest distance first)

- D → 2 (class 0)
- B → 4 (class 1)
- C → 5 (class 1)
- E → 5 (class 1)
- F → 5 (class 0)
- A → 6 (class 0)

Step 3: K = 3 prediction

Nearest 3 neighbors: **D(0), B(1), C(1)**

- Class 0 count = 1
- Class 1 count = 2

✅ Prediction (k=3) = **Class 1**

Step 4: K = 5 prediction

Nearest 5 neighbors: **D(0), B(1), C(1), E(1), F(0)**

- Class 0 count = 2
- Class 1 count = 3

✅ Prediction (k=5) = **Class 1**

◆ How to resolve a tie?

There are different strategies:

1. **Distance-weighted voting (most common)**
   - Give each neighbor a weight $w_i = 1/d_i$.
   - Sum the weights for each class.
   - Class with the highest total weight wins.

✅ In our dataset (k=6 tie):

- Class 0: D(1/2=0.5) + F(1/5=0.2) + A(1/6≈0.167) = **0.867**
- Class 1: B(1/4=0.25) + C(1/5=0.2) + E(1/5=0.2) = **0.65**

  → Class 0 wins after weighting.