k-Nearest Neighbors (kNN): Classification and Estimation

Dr. Mrityunjoy Barman

Department of CSE, ITER.

December 27, 2025

Learning Objectives

- Understand classification and estimation using kNN
- Learn the algorithmic steps and role of distance
- Visualize decision making and effect of k
- Apply kNN to real-world problems

Classification Problem

- Supervised learning task
- Output variable is categorical
- Examples: spam detection, disease diagnosis

Why kNN?

- Simple and intuitive
- No explicit training phase
- Works for both classification and regression

Algorithm Steps

1. Select number of neighbors k

Algorithm Steps

1. Select number of neighbors k
2. Compute distance to all training points

Algorithm Steps

1. Select number of neighbors k
2. Compute distance to all training points
3. Choose k nearest neighbors

Algorithm Steps

1. Select number of neighbors k
2. Compute distance to all training points
3. Choose k nearest neighbors
4. Majority vote (classification) / averaging (regression)

Explanation with an Example

- Imagine you have a new, unlabeled fruit and you want to classify it as either an "apple" or "orange" based on color and weight data from known fruits. The KNN algorithm follows these steps:
- **Store the Existing Data**: The algorithm simply memorizes the entire dataset of labeled fruits (e.g., color, weight, and whether it is an apple or an orange). There is no explicit "training" phase like in other algorithms (hence it is called a "lazy learner").
- **Introduce New Data**: A new, unlabeled fruit data point is added to the system.
- **Calculate Distance**: The system calculates the distance (or similarity) between this new fruit and all other fruits in the stored dataset. The most common method for this is the Euclidean distance (the straight-line distance on a graph).

Explanation with an Example

- **Find 'K' Nearest Neighbors**: You choose a number K (e.g., K=3) to define how many neighbors the algorithm will consider. It then identifies the K data points that are closest to the new fruit.
- **Majority Vote (Classification)**: The algorithm looks at the labels of these K neighbors. The new fruit is assigned to the class that appears most frequently among its neighbors. For example, if 2 of the 3 closest neighbors are apples and 1 is an orange, the new fruit is classified as an apple.

Distance Measures

Definition

Metric: A distance metric or distance function is a real-valued function d, such that for any points x, y, z, the following conditions are satisfied:

1. $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$;
2. $d(x, y) = d(y, x)$;
3. $d(x, z) \leq d(x, y) + d(y, z)$.

- Euclidean distance: $d(x, y) = \left\{ \sum |x_i - y_i|^2 \right\}^{1/2}$.
- Manhattan distance: $d(x, y) = \sum |x_i - y_i|$.
- Minkowski distance: $d(x, y) = \left\{ \sum |x_i - y_i|^k \right\}^{1/k}, \quad k \in \mathbb{N}$.
- Discrete metric: $d(x, y) = 1$ if $x \neq y$, and $d(x, x) = 0$.

2D Classification Dataset

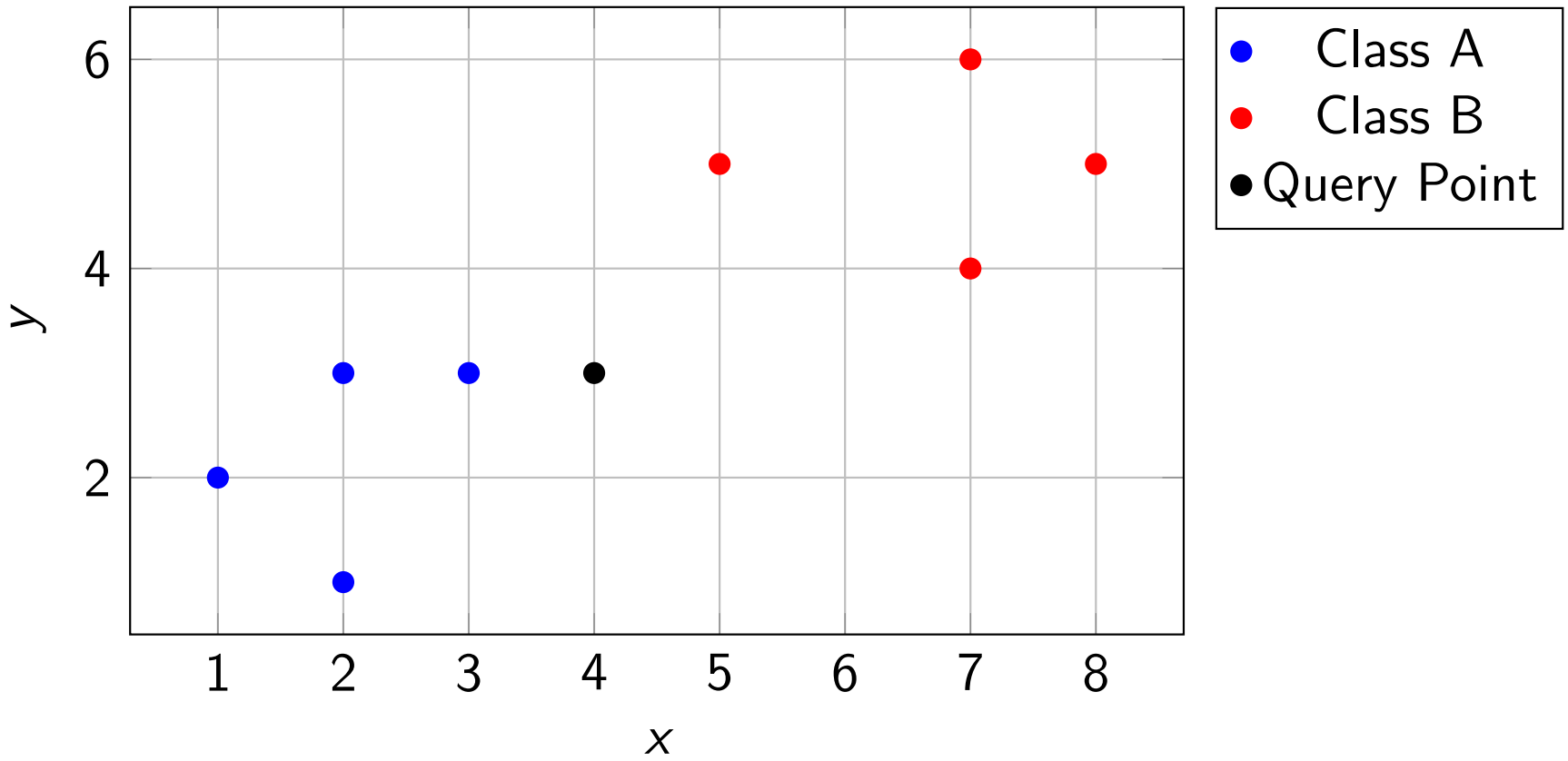


Figure: Given dataset of 2D points.

2D Classification Dataset

Suppose we have a dataset of 2D points $\{A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4\}$, having two classes. The points A_i, marked in blue colour, belong to the **class A** and the other points in red colour belong to the **class B**.

A data point **new** $= (\mathbf{4}, \mathbf{3})$ is the query point which we wish to classify based on the Euclidean distance.

Point	x	y	Distance to Query
A_1	1	2	3.1623
A_2	2	1	2.8284
A_3	2	3	2.0000
A_4	3	3	1.0000
B_1	7	4	3.1623
B_2	5	5	2.2361
B_3	8	5	4.4721
B_4	7	6	4.2426

2D Classification Dataset

For $k = 3$, majority class = **Class A**.

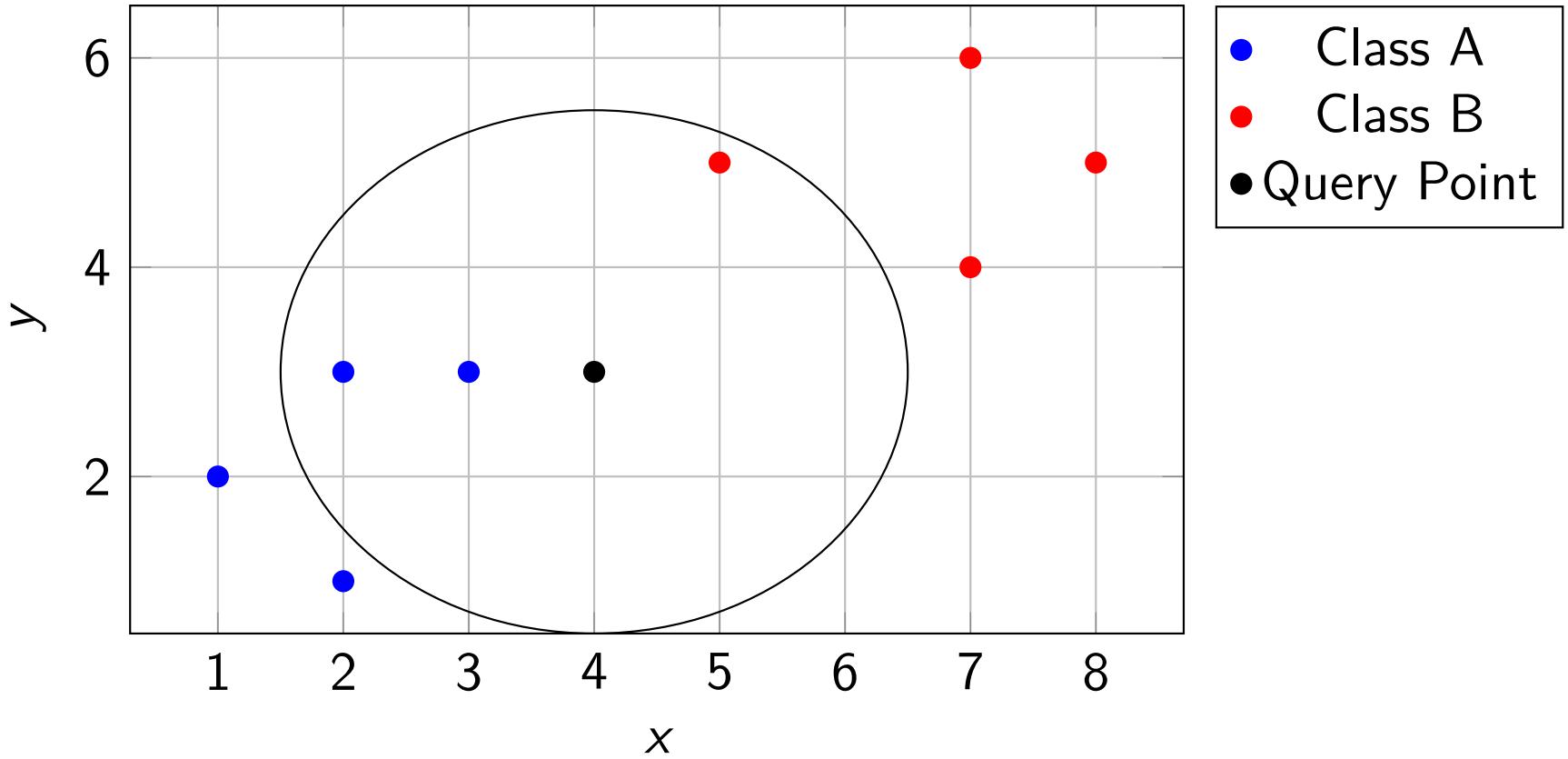


Figure: Nearest neighbours to $new = (4, 3)$ for $k = 3$.

kNN Regression (Estimation)

- Output variable is continuous
- **Prediction = average of the values at the closest neighbours**
- Example: **House price predictions at various locations in Bhubaneswar.**
- **Increased Precision:** Reduces the "smoothing" effect of distant, less relevant neighbors.
- **Outlier Mitigation:** Outliers further from the query point are assigned lower weights, reducing their impact.
- **Smooth Transitions:** Creates a more continuous prediction surface compared to simple averaging.

Introduction to Weighted kNN Regression

- **Standard kNN:** Predictions are a simple average of the k nearest neighbors.
- **Weighted kNN:** Closer neighbours have a higher influence on the prediction than those further away.
- **Goal:** Improve accuracy by prioritizing the most similar data points.

Common Weighting Schemes

Weights (w_i) are calculated based on the distance (d_i) between the query point and the i-th neighbor:

- **Inverse Distance Weighting:**

$$w_i = \frac{1}{d_i + \epsilon}, \quad \epsilon > 0.$$

  One can ignore this constant ϵ if the query point is different from all the data points.

- **Inverse Square Distance Weighting:**

$$w_i = \frac{1}{d_i^2 + \epsilon} \quad \epsilon > 0.$$

- **Rank-based Weighting:** Weights assigned based on proximity rank (1st, 2nd, etc.).

The Regression Formula

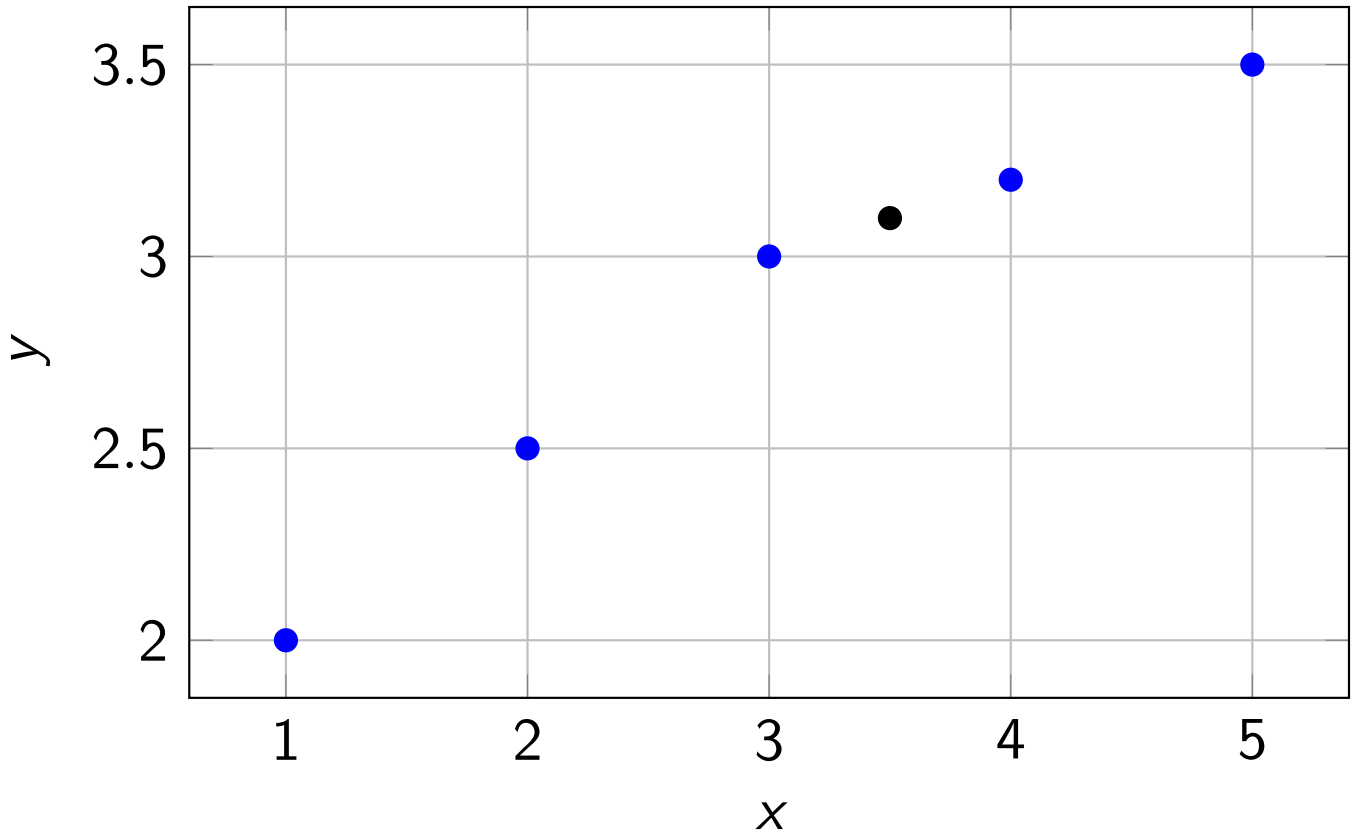
The predicted value $\hat{y}$ is calculated as a **weighted average** of the target values y_i of the k nearest neighbours:

Weighted Average Formula

$$\hat{y} = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i}$$

- This ensures the prediction is normalized relative to the sum of all weights.

Regression Visualization



Feature Scaling

- Distance-based algorithm
- Features must be normalized
- Use standardization or min-max scaling
- $dist([50, 2], [50, 3]) = 1$ vs $dist([50, 2], [40, 3]) = \sqrt{101}$.

The Necessity of Feature Scaling

kNN is a **distance-based algorithm**. Scaling is critical because:

- **Magnitude Bias:** Features with larger numeric ranges (e.g., Annual Income) will dominate features with smaller ranges (e.g., Age) in distance formulas.
- **Distance Equalization:** Scaling ensures every feature contributes proportionally to the similarity measurement.
- **Impact on Weighted Voting:** Since weights (w_i) are derived from distances, unscaled features result in skewed weights that do not reflect true similarity.

Common Scaling Methods

Method	Formula	Best Use Case
Min-Max Scaling $[0,\ 1]$	$\frac{x-x_{min}}{x_{max}-x_{min}}$	Non-Gaussian distributions; no extreme outliers.
z-Score Standardization	$\frac{x-\mu}{\sigma}$	Gaussian distributions; unknown distributions.
Robust Scaling	$\frac{x-Q_2}{IQR}$	Datasets with significant outliers (uses Median and IQR).

Workflow and Best Practices

To maintain model integrity and prevent **Data Leakage**:

1. **Fit on Train Only:** Calculate scaling parameters (mean, std, min, max) using only the training set.
2. **Transform Both:** Apply the learned parameters to both training and testing sets.
3. **Pipeline Integration:** Standardize the scaling and kNN steps into a single workflow to ensure consistency.

2025 Insight: Outlier Sensitivity

In 2025, Robust Scaling is increasingly preferred over Standard Scaling for real-world messy data to prevent outliers from collapsing the feature space.

Summary

- kNN is simple yet powerful
- Works for classification and estimation
- Visualization aids understanding
- Best suited for small–medium datasets

Exercises: Example 1

Example

Consider the following data related to a group of patients obtained from a hospital. Given that Age $\in [10, 60]$, $\sigma(Age) = 15$, $mean(Age) = 45$.

Patient	Age	Gender	Age_{mmn}	Age_z
A	50	Male		
B	20	Male		
C	50	Female		

1. Compute the following distances between the patients using Euclidean metric. You must use the discrete metric $Different(x, y)$ for the nominal variable.

	A	B	C
A	0		
B		0	
C			0

Exercises: Example 1

2. Compute the following distances between the patients (**After the transformation of the "Age" variable using MMN**) using Euclidean metric. Again, you must use the discrete metric *Different*(x, y) for the nominal variable.

	A	B	C
A	0		
B		0	
C			0

3. Compute the following distances between the patients (**After the transformation of the "Age" variable using z-score standardization**) using Euclidean metric.

	A	B	C
A	0		
B		0	
C			0

Exercises: Example 2

Consider the following data about some patients and the type of drugs advised for them.

Patient	Age	Na/K	Age_{mmn}	Na/K_{mmn}	BP
New	17	12.5	0.05	0.25	-
A (Type X)	16.8	12.4	0.0467	0.2471	120
B (Type Y)	17.2	10.5	0.0533	0.1912	122
C (Type Y)	19.5	13.5	0.0917	0.2794	130

1. Compute the distance matrix using the Euclidean metric (**without transformation**).

	New	A	B	C
New	0			

2. Applying kNN, determine the type of drug that should be advised for the New patient.

Exercises: Example 2

3. Compute the distance matrix using the Euclidean metric (**after transformation**).

	New	A	B	C
New	0			

4. Compute the following weights for each of the classes:
   - $w(Type\ X) = \frac{1}{d(New,\ A)^2} = ...$
   - $w(Type\ Y) = \frac{1}{d(New,\ B)^2} + \frac{1}{d(New,\ C)^2} = ...$
5. Following the principle of majority voting (**more weights implies more votes**), determine the type of drug that should be advised for the New patient.

Exercises: Example 2

6. Compute the predicted blood pressure of the New patient using

$$BP(New) = \frac{w(A) * BP(A) + w(B) * BP(B) + w(C) * BP(C)}{w(A) + w(B) + w(C)}$$

$$= .....$$