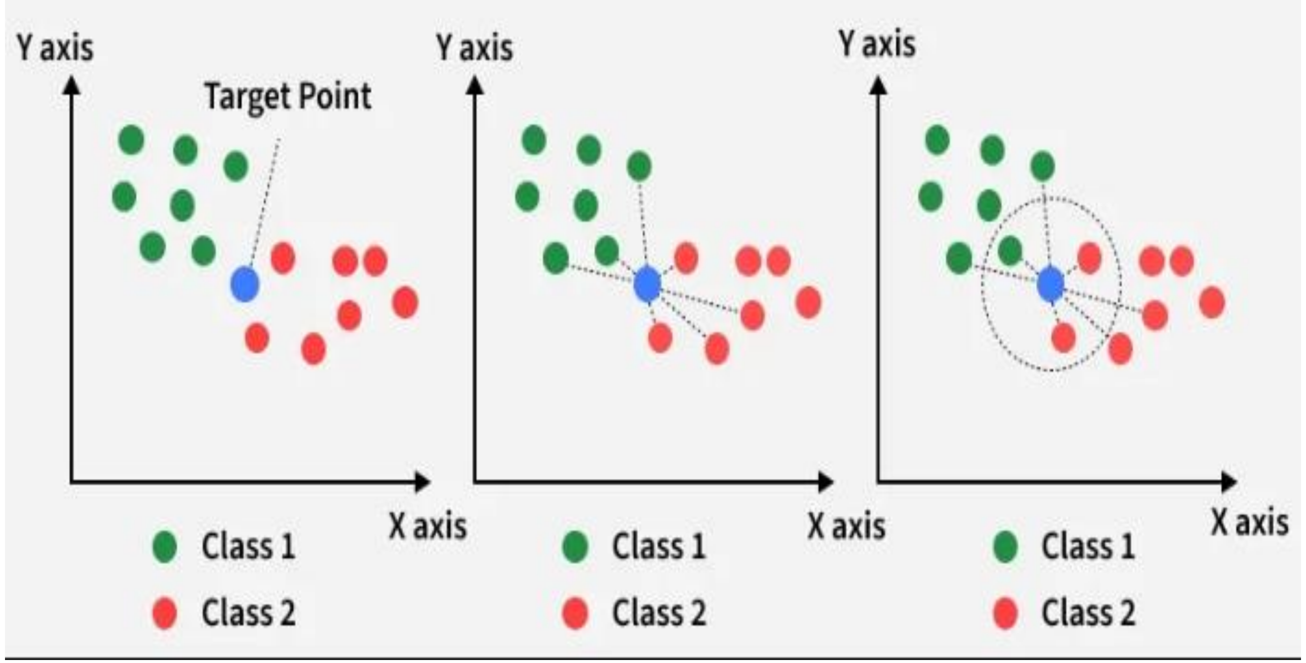
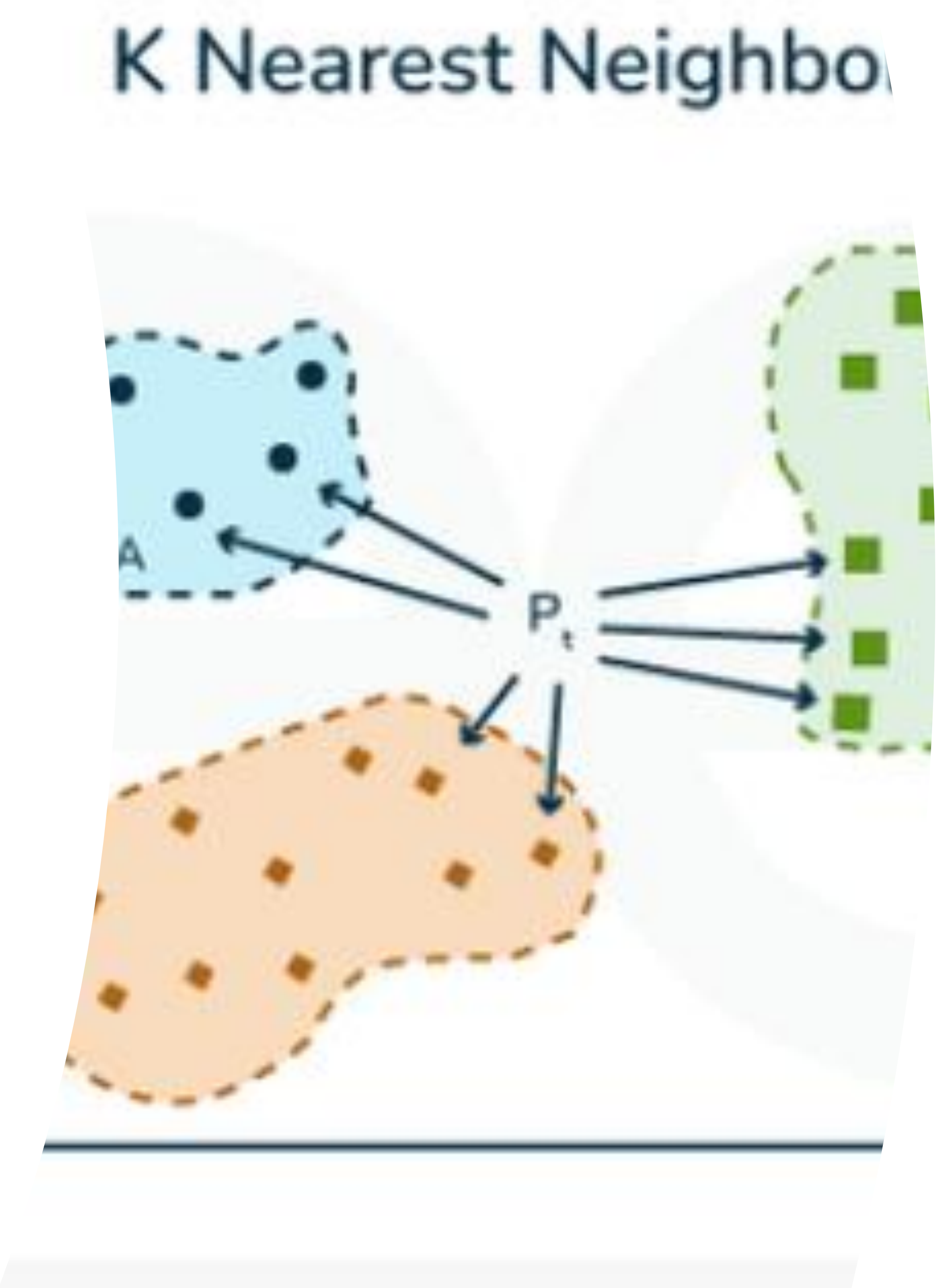
K-Nearest Neighbors (KNN)



- **Type:** Supervised, Non-parametric, Lazy learning algorithm
- **Idea:** Classify a data point based on the **majority class of its K nearest neighbors**
- **Distance Metrics:** Euclidean, Manhattan, Minkowski, Cosine
- **Steps:**
  - Choose value of **K**
  - Compute distance to all training points
  - Select **K nearest neighbors**
  - Predict by **majority vote** (classification) or **average** (regression)
- **Hyperparameter:**
  - Small K → Overfitting
  - Large K → Underfitting
- **Advantages:** Simple, no training phase, works well with small datasets
- **Limitations:** Slow for large data, sensitive to noise & feature scaling
- **Applications:** Recommendation systems, pattern recognition, anomaly detection

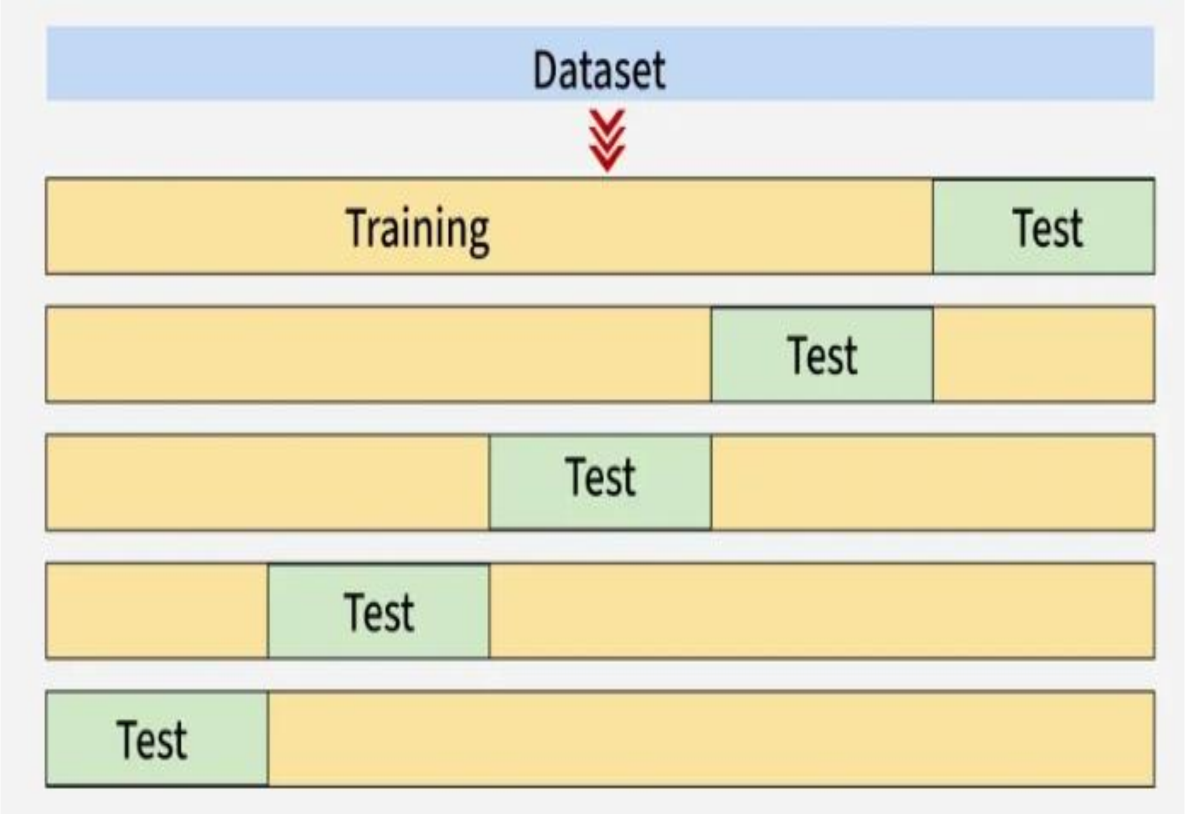
How to Choose K in K-Nearest Neighbors (KNN)



- **Rule of Thumb:**
  - K≈NK \approx \sqrt{N}K≈N (N = number of training samples)
- **Cross-Validation (Best Practice):**
  - Try multiple K values
  - Choose K with **minimum validation error**
- **Bias–Variance Tradeoff:**
  - Small K → Low bias, High variance (overfitting)
  - Large K → High bias, Low variance (underfitting)
- **Odd Value of K:**
  - Prevents ties in **binary classification**
- **Data Characteristics:**
  - Noisy data → Larger K
  - Clean & small data → Smaller K
- **Distance Sensitivity:**
  - Feature scaling affects optimal K
- **Typical Range:**
  - K=3K = 3K=3 to 151515

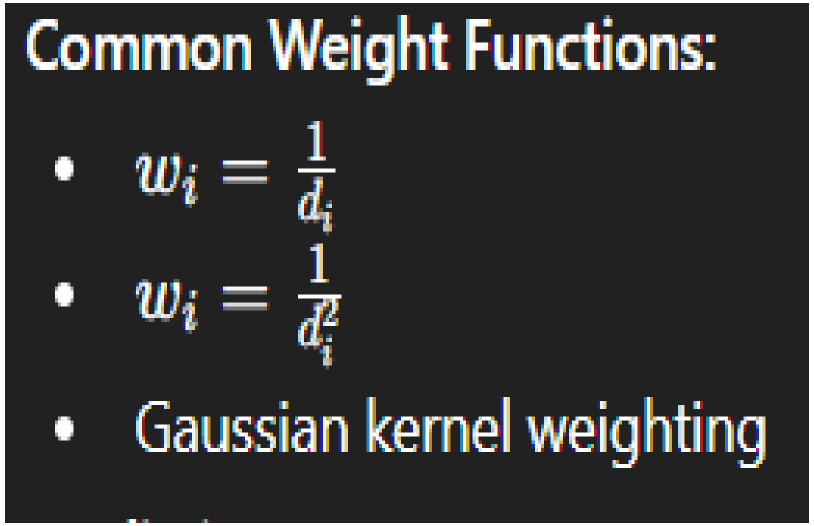
K-Fold Cross-Validation

- **Purpose:** Evaluate model performance reliably on limited data
- **Idea:** Split dataset into **K equal folds**
- **Process:**
  - Divide data into **K subsets (folds)**
  - Use **K−1 folds for training**
  - Use **1 fold for validation**
  - Repeat **K times** (each fold used once as validation)
  - **Average** performance metrics
- **Common Values of K:** 5, 10
- **Advantages:**
  - Better generalization estimate
  - Efficient use of data
  - Reduces variance vs single split
- **Limitations:**
  - Computationally expensive
  - Not ideal for very large datasets
- **Variants:** Stratified K-Fold (class balance), Leave-One-Out (K=N)



Weighted K-Nearest Neighbors (Weighted KNN)

- **Extension of KNN:** Assigns **higher influence to closer neighbors**
- **Idea:** Neighbors contribute with **weights inversely proportional to distance**
- **Prediction:**
    - **Classification:** Class with **maximum weighted vote**
    - **Regression: Weighted average** of neighbor values
- **Why Weighted KNN?:**
    - Reduces impact of distant/noisy neighbors
    - Improves decision boundaries
- **Advantages:**
    - Better accuracy than standard KNN
    - Less sensitive to choice of K
- **Limitations:**
    - Sensitive to distance metric & scaling
    - Slightly higher computation
- **Use Cases:** Noisy datasets, imbalanced data, spatial problems



Common Weight Functions:

- $w_i = \frac{1}{d_i}$
- $w_i = \frac{1}{d_i^2}$
- Gaussian kernel weighting