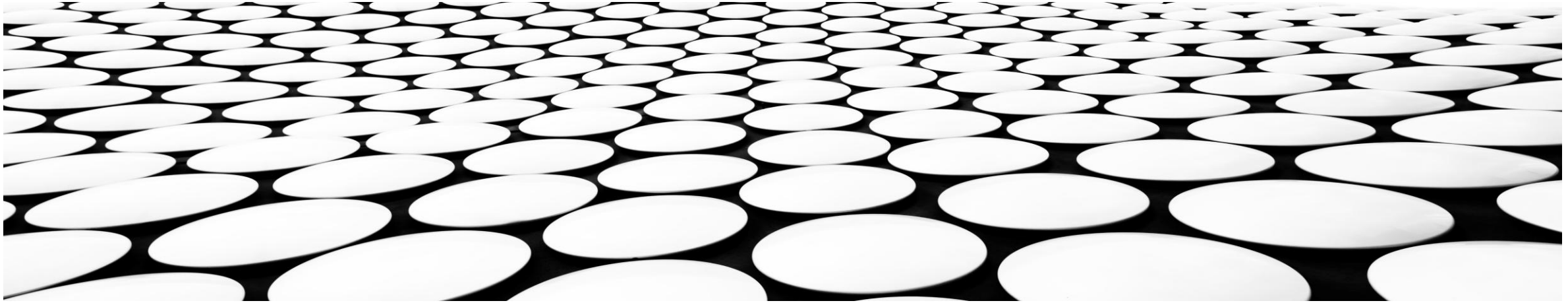
DATA MINING AND PREDICTIVE DATA ANALYTICS

CHAPTER-10

CLASSIFICATION



K-NEAREST NEIGHBOR ALGORITHM

- The **k-Nearest Neighbor (k-NN)** algorithm is one of the simplest and most intuitive algorithms used in classification and prediction tasks.
- It belongs to a class of methods known as instance-based learning or lazy learning.
- Unlike many ML algorithms, k-NN does not build an explicit model during training.
- Instead, it stores the entire training dataset and performs computation only when a new, unseen data point (object) needs to be classified.
- The fundamental idea of k-NN is:

  "Similar data points tend to belong to the same class."

  "Similarity is measured using distance metric"

K-NEAREST NEIGHBOR ALGORITHM

- **To classify a new data point:**
  - Measure its distance from all training points.
  - Select the k nearest neighbors.
  - Assign the class that occurs most frequently among those neighbors.
- **Characteristics of k-NN**
  - Non-parametric: No assumptions about data distribution.
  - Instance-based: Uses stored training data directly for prediction.
  - Lazy learner: No training phase; computation happens at prediction time.
  - Sensitive to:
    - Choice of k
    - Distance metric
    - Feature scaling

DISTANCE FUNCTION

- Why Do We Need a Distance Function?
  - In the k-NN algorithm, a new record is classified based on the most similar existing records.
  - A distance function (or distance metric) is used to measure similarity between two records
  - A distance metric is a function $d(x,y)$ that satisfies the following properties for any points x,y,z
    - Non-negativity: $d(x,y) \geq 0$,and $d(x,y)=0$ if and only if $x=y$
    - Symmetry: $d(x,y)=d(y,x)$
    - Triangle Inequality: $d(x,z) \leq d(x,y)+d(y,z)$

DISTANCE FUNCTION

- **Euclidean Distance (Most Common Distance Function)**
  - The most widely used distance function in k-NN is Euclidean distance, which corresponds to straight-line distance.
  - For two records (objects) x=(x1,x2,…,xm), y=(y1,y2,…,ym) the Euclidean distance is:

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

- Example:
  - Object A: (10,80)
  - Object B: (14,74)
  - Distance between Objects= 7.21

$$d(A, B) = \sqrt{(10 - 14)^2 + (80 - 74)^2}$$

$$= \sqrt{16 + 36} = \sqrt{52} \approx 7.21$$

  - Note: A smaller distance implies greater similarity between the students.

DISTANCE FUNCTION

- **Problem of Scale in Distance Measurement**
  - When attributes have very different numerical ranges, large-scale attributes can dominate the distance.
  - Therefore, we need to normalize the data
- Min–Max Normalization (for continuous data):
  - Scales values into the range [0, 1]
  - Preserves relative ordering

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

- Z-Score Standardization (for continuous data):
  - Centers data around 0
  - Produces values typically between −3 and +3

$$X' = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

DISTANCE FUNCTION

- Distance for Categorical Variables:
  - Euclidean distance is not suitable for categorical attributes.
  - Instead, we use a difference function:

$$\text{different}(x_i, y_i) = \begin{cases} 0, & \text{if } x_i = y_i \\ 1, & \text{otherwise} \end{cases}$$

  - This allows categorical attributes to be included in distance calculations.
- **Mixed-Attribute Distance (Simple Example)**

Student	Study Hours	Department
A	10	CSE
B	6	CSE
C	10	ECE

DISTANCE FUNCTION

- Without Normalization:

$$d(A, B) = \sqrt{(10 - 6)^2 + 0^2} = 4$$

$$d(A, C) = \sqrt{(10 - 10)^2 + 1^2} = 1$$

  - Student C appears more similar to A than B.

Student	Study Hours	Department
A	10	CSE
B	6	CSE
C	10	ECE

- Effect of Min–Max Normalization

$$A = \frac{10 - 0}{20} = 0.5, \quad B = \frac{6 - 0}{20} = 0.3, \quad C = 0.5$$

$$d_{\text{MM}}(A, B) = \sqrt{(0.5 - 0.3)^2 + 0^2} = 0.2$$

$$d_{\text{MM}}(A, C) = \sqrt{(0.5 - 0.5)^2 + 1^2} = 1$$

  - Conclusion reverses: Student B is now closer to A.

Assume the **range of study hours** is:

- $\min(X) = 0$
- $\max(X) = 20$

DISTANCE FUNCTION

- Effect of Z-Score Standardization
  - Assume: Mean study hours = 8, Standard deviation = 2

$$A = \frac{10-8}{2} = 1, \quad B = \frac{6-8}{2} = -1, \quad C = 1$$

$$d_z(A,B) = \sqrt{(1+1)^2 + 0^2} = 2$$

$$d_z(A,C) = \sqrt{(1-1)^2 + 1^2} = 1$$

  - Now Student C is again closer.
- **Key Observations**
  - Distance-based algorithms are highly sensitive to scaling
  - Different normalization methods can change classification results

DISTANCE FUNCTION

- **Practical Guideline**
  - When mixing: Continuous + categorical variables → Min–max normalization is often preferred
  - Only continuous variables → Z-score standardization is commonly used
- **Summary of Distance Metric in k-NN**
  - Distance functions define similarity in k-NN
  - Euclidean distance is the most common choice
  - Attribute scaling is crucial
  - Normalization can change nearest neighbors
  - Understanding normalization is essential for correct classification

KNN - SIMPLE EXAMPLE

- Problem Statement:
  - A university wants to classify students into Pass (P) or Fail (F) based on:
    - Hours of Study per Week
    - Attendance Percentage
  - Given Dataset:

Student	Study Hours	Attendance (%)	Result
S1	8	75	Pass
S2	6	65	Fail
S3	9	80	Pass
S4	5	60	Fail
S5	7	70	Pass

New Student (Unclassified)

- Study Hours = 6.5
- Attendance = 68%

  - We want to predict whether this student will Pass or Fail

KNN - SIMPLE EXAMPLE

- New Student (Unclassified): x=(6.5,68)
- Compute Euclidean distance between New Student and each of the student in the data set. $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$

Student	Computed Euclidean Distance *d(x,y)*	Distance from New Student	Result
S5	**2.06**	Smallest	**Pass**
S2	**3.04**	Small	**Fail**
S1	7.16	Medium	Pass
S4	8.14	Larger	Fail
S3	12.26	Largest	Pass

KNN - SIMPLE EXAMPLE

- **Effect of Choosing Different Values of k**
  - **Case 1: k = 1**
    - Nearest neighbor: S5 (Pass)
    - Prediction: Pass
  - Decision depends on a single nearest point → very sensitive to noise.
  - **Case 2: k = 2**
    - Nearest neighbors: S2 (Fail), S5 (Pass)
    - Tie situation → no clear decision
  - Voting fails when k is even and classes are balanced.

KNN - SIMPLE EXAMPLE

- Case 3: k = 3
  - Nearest neighbors: S2 (Fail), S5 (Pass), S1 (Pass)
  - Majority vote: Pass
  - More stable and reliable decision.
- Choosing k:
  - Small k → sensitive to noise
  - Large k → smoother but may ignore local patterns

COMBINATION FUNCTION

- In k-NN
  - Distance identifies neighbors (k),
  - While the **combination function** determines how neighbors' information is aggregated to make a final classification decision.
- The two most commonly used combination functions in k-NN are:
  - Majority Voting
  - Weighted Voting
- Majority Voting
  - Each neighbor gets one equal vote
  - Distance is ignored after neighbor selection
  - The class with the maximum number of votes is assigned to the new record

KNN - SIMPLE EXAMPLE

- **Weighted Voting**
  - Core idea: Neighbors closer to the new record should have greater influence on the classification (i.e., closer neighbors are more informative than farther ones)
  - Neighbors that are closer or more similar to the new record should be weighted more heavily than more distant neighbors.
  - Decision Rule
    - Votes are weighted by distance
    - The class with the highest total weighted vote is selected
  - Weight Assignment: The weight of each neighbor is inversely proportional to its distance from the new record: $$\text{Weight} = \frac{1}{\text{Distance}}$$

K-NEAREST NEIGHBOR ALGORITHM

- Advantages of k-NN
  - Easy to understand and implement
  - No training phase required
  - Works well for small datasets
  - Naturally handles multi-class problems
- Limitations of k-NN
  - Computationally expensive for large datasets
  - Requires storing entire dataset
  - Sensitive to irrelevant features
  - Performance depends heavily on feature scaling

K-NEAREST NEIGHBOR ALGORITHM

- Summary
  - k-NN classifies data based on similarity.
  - The value of k plays a crucial role in prediction.
  - Distance measurement defines “nearness.”
  - Simple majority voting may fail in some cases.
  - Weighted voting can improve classification accuracy.
- Key Takeaway: k-NN does not learn rules — it predicts (classifies) by remembering examples (unlike other classifiers)
- k-NN memorizes the data and makes predictions based on the closest examples when needed.