K-Nearest Neighbors (KNN)

A Detailed and Descriptive Presentation

February 27, 2025

Table of Contents

Introduction to KNN

Mathematical Formulation

Working of KNN

Example

Advantages and Disadvantages

Applications

Conclusion

Introduction to KNN

What is KNN?

- K-Nearest Neighbors (KNN) is a **supervised machine learning algorithm** used for classification and regression tasks.
- It is a **non-parametric** and **instance-based** learning algorithm, meaning it does not make any assumptions about the underlying data distribution and does not learn a model during training.
- KNN works by finding the *k* nearest data points (neighbors) in the feature space and making predictions based on their labels (for classification) or values (for regression).

Introduction to KNN

Key Characteristics

- **Lazy Learner:** KNN does not explicitly learn a model during training. Instead, it memorizes the training dataset and performs computations at prediction time.
- **Distance-Based:** It relies on a distance metric (e.g., Euclidean, Manhattan) to find the nearest neighbors.
- **Hyperparameter k:** The number of neighbors k is a critical hyperparameter that affects the algorithm's performance.

Mathematical Formulation

Distance Metrics

KNN relies on distance metrics to measure the similarity between data points. Common distance metrics include:

- **Euclidean Distance:**

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

- **Manhattan Distance:**

$$d(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

- **Minkowski Distance:**

$$d(p, q) = \left(\sum_{i=1}^{n} |p_i - q_i|^r \right)^{1/r}$$

(Euclidean is a special case of Minkowski with $r = 2$, and Manhattan with $r = 1$.)

Mathematical Formulation

Prediction Rule

For classification:

$$\hat{y} = \text{mode}(y_{i_1}, y_{i_2}, \ldots, y_{i_k})$$

For regression:

$$\hat{y} = \frac{1}{k} \sum_{j=1}^{k} y_{i_j}$$

Where:

- $\hat{y}$: Predicted label or value.
- y_{i_j}: Label or value of the j-th nearest neighbor.

Working of KNN

Step-by-Step Process

1. **Choose k:** Select the number of neighbors k.
2. **Calculate Distances:** Compute the distance between the query point and all training data points.
3. **Find Nearest Neighbors:** Identify the k nearest neighbors based on the calculated distances.
4. **Majority Voting (Classification):** Assign the class that appears most frequently among the k neighbors.
5. **Averaging (Regression):** Predict the average value of the k neighbors.

Working of KNN

Visualization

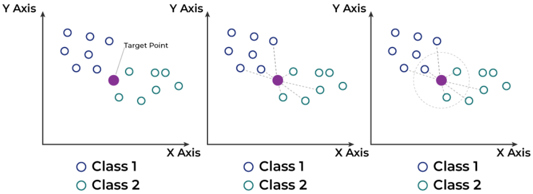

Figure: Step 1: Selecting the optimal value of K
Step 2: Calculating distance
Step 3: Finding Nearest Neighbors
Step 4: Voting for Classification or Taking Average for Regression

Example

Classification Problem

Problem: Classify a new data point $(3, 4)$ using $k = 3$.

Point	Class
(1, 2)	A
(2, 3)	A
(4, 5)	B
(5, 6)	B

Solution:

- Calculate Euclidean distances:

$$d((3,4),(1,2)) = \sqrt{(3-1)^2 + (4-2)^2} = \sqrt{8} \approx 2.83$$

$$d((3,4),(2,3)) = \sqrt{(3-2)^2 + (4-3)^2} = \sqrt{2} \approx 1.41$$

$$d((3,4),(4,5)) = \sqrt{(3-4)^2 + (4-5)^2} = \sqrt{2} \approx 1.41$$

$$d((3,4),(5,6)) = \sqrt{(3-5)^2 + (4-6)^2} = \sqrt{8} \approx 2.83$$

Advantages and Disadvantages

Advantages

- Simple and easy to implement.
- No training phase; the model is ready to use once the data is stored.
- Can be used for both classification and regression.
- Adapts easily to new data.

Advantages and Disadvantages

Disadvantages

- Computationally expensive during prediction, especially for large datasets.
- Sensitive to the choice of k and the distance metric.
- Requires feature scaling for accurate results.
- Struggles with high-dimensional data (curse of dimensionality).

Applications

Real-World Use Cases

- **Recommendation Systems:** Suggest products or content based on user similarity.
- **Image Recognition:** Classify images based on similar features.
- **Medical Diagnosis:** Predict diseases based on patient data.
- **Credit Scoring:** Assess credit risk based on historical data.

Conclusion

- KNN is a simple yet powerful algorithm for classification and regression tasks.
- It is a lazy learner that relies on distance metrics and majority voting.
- Proper tuning of k and feature scaling are crucial for optimal performance.
- Despite its limitations, KNN is widely used in various real-world applications.