K-Nearest Neighbors (KNN) Algorithm

1 Introduction

K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm used for classification and regression tasks. It is a non-parametric method, meaning it makes no assumptions about the underlying data distribution.

2 Algorithm

The KNN algorithm works as follows:

1. Choose the number of neighbors k.
2. Calculate the distance between the query instance and all the training samples.
3. Select the k nearest neighbors based on the calculated distances.
4. For classification, assign the class label that is most frequent among the k nearest neighbors. For regression, compute the average of the values of the k nearest neighbors.

3 Distance Metrics

Common distance metrics used in KNN 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:** $d(p, q) = \left(\sum_{i=1}^{n} |p_i - q_i|^p\right)^{1/p}$

4 Advantages and Disadvantages

4.1 Advantages

- Simple to implement and understand.
- No training phase, making it fast for small datasets.
- Can be used for both classification and regression tasks.

4.2 Disadvantages

- Computationally expensive for large datasets.
- Performance depends on the choice of k and the distance metric.
- Sensitive to irrelevant or redundant features.

5 Implementation

Here is a simple implementation of the KNN algorithm in pseudocode:

Algorithm 1: K-Nearest Neighbors (KNN) Algorithm

Input: Training data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, Test point: $\mathbf{x}$, Number of neighbors: k
Output: Predicted label $\hat{y}$ for test point $\mathbf{x}$
for $i \leftarrow 1$ **to** N **do**
  Compute distance $d_i = \|\mathbf{x} - \mathbf{x}_i\|$;
Sort the distances $\{d_i\}_{i=1}^{N}$ in ascending order and obtain indices of the k smallest distances: $\{i_1, i_2, \ldots, i_k\}$;
for $j \leftarrow 1$ **to** k **do**
  Retrieve the label y_{i_j} of the j-th nearest neighbor;
if *classification* **then**
  $\hat{y} \leftarrow \text{mode}\{y_{i_1}, y_{i_2}, \ldots, y_{i_k}\}$;
else if *regression* **then**
  $\hat{y} \leftarrow \frac{1}{k} \sum_{j=1}^{k} y_{i_j}$;
return $\hat{y}$;

6 Conclusion

KNN is a versatile and intuitive algorithm that can be applied to various machine learning tasks. However, its performance can be significantly affected by the choice of k, the distance metric, and the presence of noisy or irrelevant features.