0 k-Nearest Neighbors (kNN) Method

The k-Nearest Neighbors method is a supervised learning model that predicts based on the nearest records in a multi-dimensional space of training data.

0.1 Hyperparameters

0.1.1 Number of Neighbors (k)

The parameter k determines the number of nearest neighbors from which the model calculates predictions. A higher value helps prevent overfitting.

0.1.2 Distance Metric

A metric on set $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \longrightarrow [0, +\infty)$ such that for every $x, y, z \in \mathcal{X}$, the following properties hold:

1. Non-negativity: $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$,
2. Symmetry: $d(x, y) = d(y, x)$,
3. Triangle Inequality: $d(x, y) + d(y, z) \geq d(x, z)$.

Common metrics include Minkowski L_k distances:

$$\|\mathbf{x} - \mathbf{y}\|_k = d_k(\mathbf{x}, \mathbf{y})_k = \sqrt[k]{\sum_{i=0}^{p-1} |x_i - y_i|^k}$$

Specifically, for $k = 1$, Manhattan distance:

$$d_1(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{p-1} |x_i - y_i|$$

For $k = 2$, Euclidean distance:

$$d_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=0}^{p-1} (x_i - y_i)^2}$$

And for $k = +\infty$, Chebyshev distance:

$$d_\infty(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|$$

Other frequently used metrics include Levenshtein edit distance for strings or cosine distance for vectors based on the angle between them:

$$d(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}$$

There are more sophisticated metrics like Jaccard for sets or Haversine for distance between two points on a sphere.

0.1.3 Neighbor Weights

For regression tasks, neighbor weights can be uniform (each of the k neighbors has the same influence on the prediction):

$$\hat{y} = \frac{1}{k} \sum_{i=0}^{k-1} y_i$$

or weighted by distance (closer neighbors have more influence):

$$\hat{y} = \frac{\sum_{i=0}^{k-1} w_i y_i}{\sum_{i=0}^{k-1} w_i}, \quad \text{where} \quad w_i = \frac{1}{d(\mathbf{x}_i, \boldsymbol{n}_i)}$$

0.2 Applications for Classification and Regression

For training data $\mathbf{X} \in \mathbb{R}^{N,p}$ with the target variable $Y \in \mathbb{R}^N$, the kNN method predicts the value of the target variable for a data point $\mathbf{x} \in \mathbb{R}^p$ by considering the votes (classification) or the average (regression) of k nearest neighbors.

The kNN model has very inexpensive "training"as it only stores the training set in memory. However, predictions can be computationally expensive.

0.3 Data Normalization

Due to different feature ranges, the value ranges in original data contribute unevenly to the distance. This problem can be addressed by normalizing to the interval $[0, 1]$ or $[-1, 1]$, or by standardization.

$$\text{Normalization: } x_i \leftarrow \frac{x_i - \min_x}{\max_x - \min_x} \qquad \text{Standardization: } x_i \leftarrow \frac{x_i - \bar{x}}{\sqrt{s_x^2}}$$