Lesson 8: K-Nearest Neighbors (KNN)

Reviewer

What is K-Nearest Neighbors (KNN)?

K-Nearest Neighbors (KNN) is a ***Supervised Learning*** algorithm.

- ***Supervised Learning***: A ***versatile supervised machine learning algorithm*** primarily used for both ***classification*** and ***regression tasks***.
- ***Non-Parametric & Lazy***: It's ***non-parametric***, meaning it ***makes no underlying assumptions about data distribution***, and a ***"lazy learner"*** as it ***memorizes the entire dataset*** instead of explicit model training.
- ***Prediction by distance***: Predictions are made by identifying and analyzing the ***"K" closest data points*** to a new, unclassified input, forming a local estimate.

Why Use KNN? The Power of distance

KNN operates on a fundamental principle: ***similar things exist in close distance*** within a feature space. This makes it incredibly intuitive and accessible for various applications. It ***assumes inherent similarity*** between nearby data points.

It is remarkably easy to understand and implement, even for beginners. It is highly effective for tasks like ***pattern recognition***, building ***recommendation engines***, and supporting ***data-driven decision-making***.

How Does KNN Work? Step-by-Step

The process involves the following steps:

1. ***Choose K***: Select the ***number of nearest neighbors (K)*** to consider for prediction.
2. ***Calculate Distances***: Measure the distance from the ***new data point*** to all ***training data points***.
3. ***Identify Neighbors***: Find the ***K data points with the smallest distances*** (the closest neighbors).
4. ***Classify or Regress***: For ***classification***, the new point takes the ***majority class*** of its K neighbors. For ***regression***, it's the ***average value***.

Distance Metrics formula: Euclidean Distance

In short, the ***Euclidean Distance*** represents the ***shortest distance*** between two points. You are most likely to use this method when calculating the distance between two rows of data that

have numerical values, such as floating point or integer values.

Imagine the image on the right, you are taking a trip from Barcelona to Berlin. Of course the fastest method of transport is to fly, but how far exactly is this journey?

Definition

Euclidean distance is defined as the ***straight-line distance*** between two points in a plane or space. You can think of it like the ***shortest path*** you would walk if you were to go directly from one point to another.

Formula

The Euclidean Distance (d) between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is:

$$\boldsymbol{d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

Euclidean Distance: Pythagorean Theorem

Enter Pythagoras, a Greek philosopher and inventor of the infamous ***Pythagorean Theorem*** which stated that:

"In a right-angled triangle, the sum of the square of the hypotenuse side is equal to the sum of the squares other two sides."

$$\boldsymbol{a^2 + b^2 = c^2}$$

Derivation

So using this theorem:

1. The distance from Barcelona to Berlin squared (AC^2) is equal to $AB^2 + BC^2$.
2. Therefore AC is equal to $\sqrt{AB^2 + BC^2}$.
3. Since AB is equal to $(x_2 - x_1)$ and likewise BC is equal to $(y_2 - y_1)$, we adjust our formula and we get this, the ***simplest formula for Euclidean Distance***:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance Metrics formula: Manhattan Distance

This is the ***total distance*** you would travel if you could only move along ***horizontal and vertical lines*** like a grid or city streets. It's also called ***"taxicab distance"*** because a taxi can only drive along the grid-like streets of a city.

The ***Manhattan Distance*** is used to calculate the distance between two coordinates in a ***grid-like path***.

Imagine you are on holidays in New York City, you are visiting the Empire State Building and decide to walk to The Morgan Library & Museum by the route below. From the map it is easy to see why Manhattan Distance is also known as city block distance or taxicab geometry.

Visualizing Route

Given two points (x_1, y_1) and (x_2, y_2), the Manhattan Distance d between them is:

$$d = |x_2 - x_1| + |y_2 - y_1|$$

Why the Absolute Value?

The ***absolute value*** ensures that the ***distance is always positive***, regardless of the direction. Think about it: If you were to walk 5 blocks north or 5 blocks south, the effort (or distance) is the same. It's 5 blocks!

Manhattan Distance in Machine Learning

In machine learning, the Manhattan distance is often used in ***clustering algorithms*** or when we need a distance metric between two datasets.

Distance Metrics formula: Minkowski Distance

Minkowski distance is like a ***family of distances***, which includes both Euclidean and Manhattan distances as special cases.

From the formula, when $p = 2$, it becomes the same as the Euclidean distance formula and when $p = 1$, it turns into the Manhattan distance formula. Minkowski distance is essentially a ***flexible formula*** that can represent either Euclidean or Manhattan distance depending on the value of p.

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}}$$

- Minkowski when $p = 1 \rightarrow$ ***Manhattan***
- Minkowski when $p = 2 \rightarrow$ ***Euclidean***

Working with KNN

The K-Nearest Neighbors (KNN) algorithm operates on the ***principle of similarity*** where it predicts the label or value of a new data point by considering the labels or values of its ***K nearest neighbors*** in the training dataset.

Steps how KNN works

Step 1: Selecting the optimal value of K

- K represents the ***number of nearest neighbors*** that needs to be considered while making prediction.

Step 2: Calculating distance

- To measure the similarity between target and training data points ***Euclidean distance*** is widely used.
- Distance is calculated between data points in the dataset and target point.

Step 3: Finding Nearest Neighbors

- The k ***data points with the smallest distances*** to the target point are nearest neighbors.

Step 4: Voting for Classification or Taking Average for Regression

- **Classification:** When you want to classify a data point into a category like spam or not spam, the KNN algorithm looks at the K closest points in the dataset. These closest points are called neighbors. The algorithm then looks at which category the neighbors belong to and picks the one that appears the most. This is called ***majority voting***.
- **Regression:** In regression, the algorithm still looks for the K closest points. But instead of voting for a class in classification, it takes the ***average of the values*** of those K neighbors. This average is the predicted value for the new point for the algorithm.

Real-World Applications of KNN

KNN's versatility makes it a valuable tool across diverse industries.

- ***Medical Diagnosis***: Classifying tumors as benign or malignant based on patient data, aiding early detection.
- ***Recommendation Systems***: Suggesting movies, products, or music to users based on the preferences of similar individuals.
- ***Fraud Detection***: Identifying unusual transactions by comparing new patterns to known legitimate or fraudulent activities.
- ***Image Recognition***: Categorizing objects or scenes in images by finding similarities to previously labeled image data.

Advantages & Disadvantages of KNN

Advantages:

- ***Simplicity & Flexibility***: Intuitive, easy to implement, and makes no data distribution assumptions.
- ***Handles Complexity***: Works well with multi-class problems and adapts to complex decision boundaries.
- ***No Training Phase***: ***"Lazy learner"*** means no explicit training model creation; the data is the model.

Disadvantages:

- ***Computational Cost***: Can be computationally expensive during prediction for large datasets.
- ***Feature Scaling Needed***: Requires data normalization or standardization for optimal performance.
- ***Curse of Dimensionality***: Performance degrades significantly with high-dimensional data as distances become less meaningful.