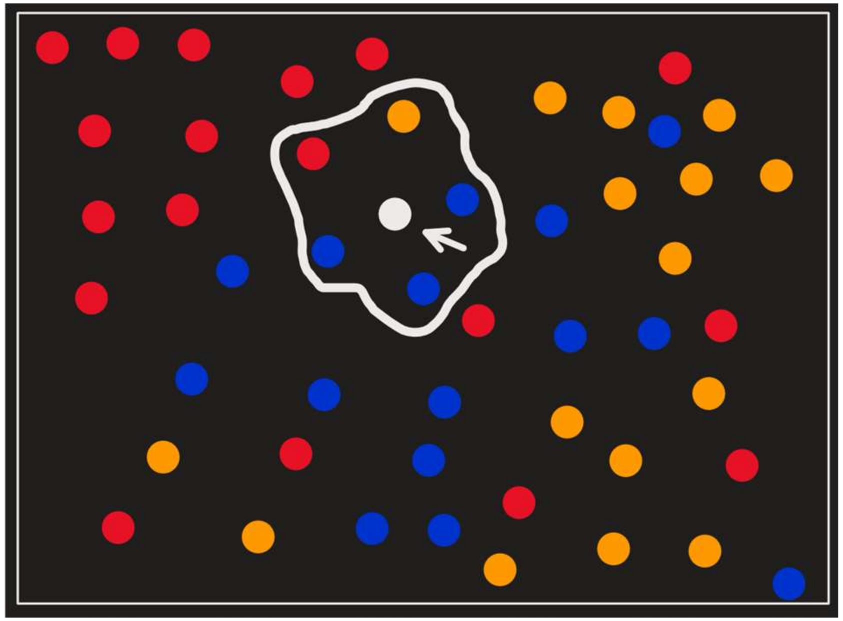
kNN Classification

- kNN stands for k-nearest neighbors.
- The classification of a new data point is determined by the majority vote of the k nearest neighbors.
- The distance between points can be calculated in various ways, in our case, we use the Manhattan distance.
- Example: For K=5 with 3 blue, 1 red, 1 orange neighbors (see right graphic), the test point is classified as 'blue'.

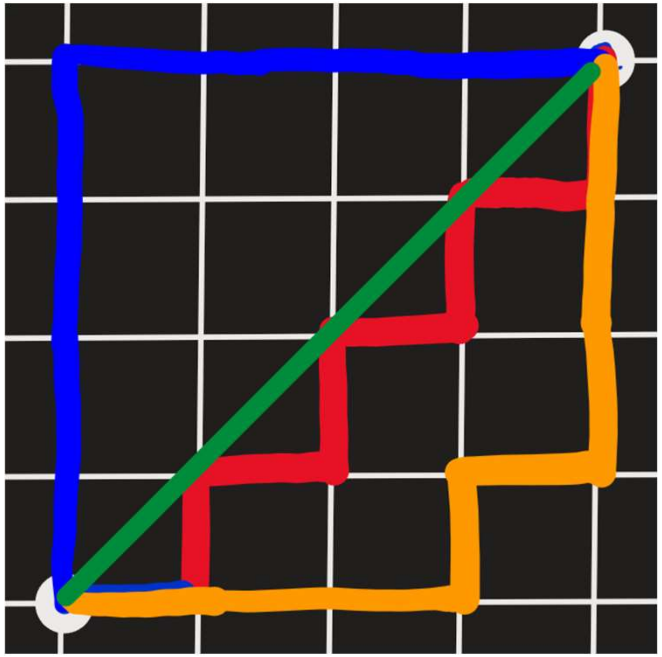


Mannhatten distance

- The Manhattan distance (L1 norm) measures the distance between two points along the axes of a grid.
- Formula:

$$Distance(x, y) = \sum_{i=1}^{n} |x_i - y_i| = |x_2 - x_1| + |y_2 - y_1| + \ldots + |x_n - x_1| + |y_n - y_1|$$

- In the context of kNN, it is used to determine the k nearest neighbors of a test point.
- It is particularly suitable for spatial data with rectangular paths and effectively reflects the physical distance in block-based or urban environments. However, it is less suitable for direct distance measurement between points. In such cases, the Euclidean distance (L2 norm) is a better choice (see right graphic).

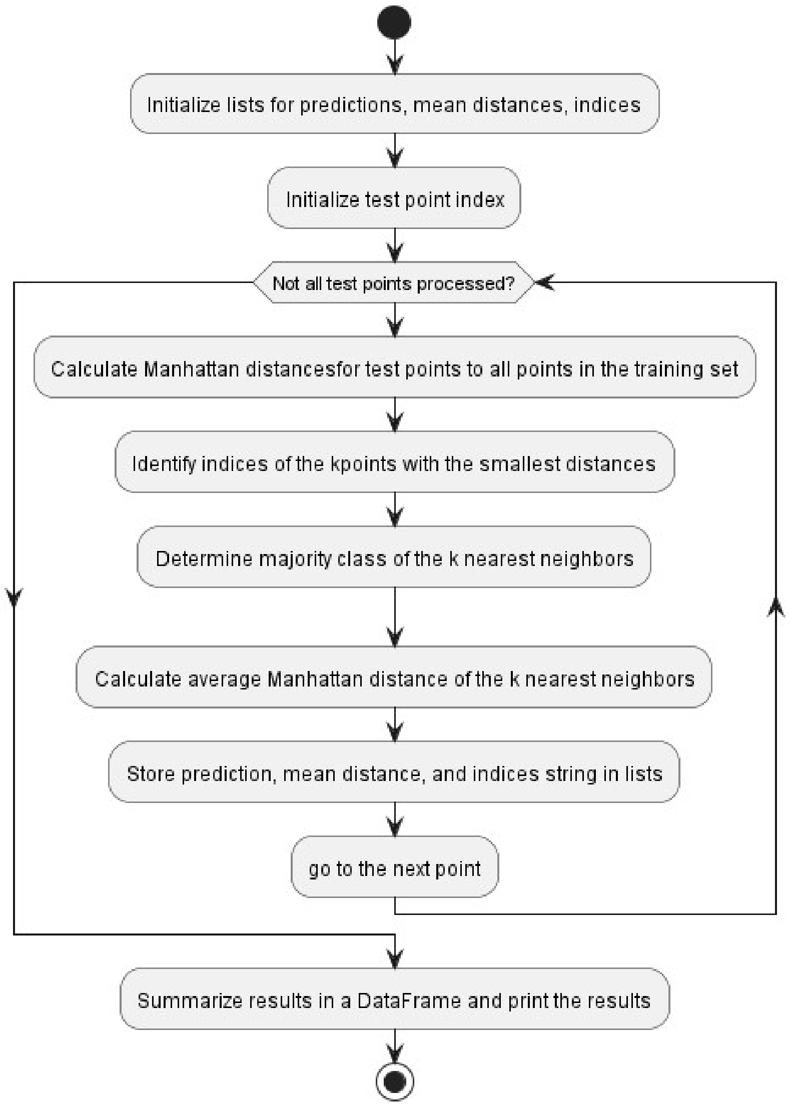


The orange, red, and blue lines are examples of the Manhattan distance (L1 norm). The green line represents the Euclidean distance (L2 norm).

Implementation of the kNN algorithm

- The algorithm takes a dataset (training data), a new data point (test data), and the number of neighbors k as input.
- The k nearest neighbors are identified, and the most frequent class among these neighbors is used as the prediction for the test point.
- Additionally, the average Manhattan distance of the k nearest neighbors is calculated and outputted along with the indices of these neighbors.
- Subsequently, the predictions, average distances, and indices of the k nearest neighbors for each test point are summarized in a DataFrame and outputted.

kNN - algorithm description



Initialize lists for predictions, mean distances, indices

Initialize test point index

Not all test points processed?

Calculate Manhattan distancesfor test points to all points in the training set

Identify indices of the kpoints with the smallest distances

Determine majority class of the k nearest neighbors

Calculate average Manhattan distance of the k nearest neighbors

Store prediction, mean distance, and indices string in lists

go to the next point

Summarize results in a DataFrame and print the results