K-NEAREST NEIGHBOURS (K-NN)

K-NN DEFINITION (1/2)

- KNN falls in the supervised learning family of algorithms.
- The KNN classifier is also a non-parametric and instance-based learning algorithm:
  - **Non-parametric** means it makes no explicit assumptions about the functional form of h, avoiding the dangers of mismodeling the underlying distribution of the data.
  - **Instance-based** learning means that our algorithm doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are subsequently used as "knowledge" for the prediction phase.

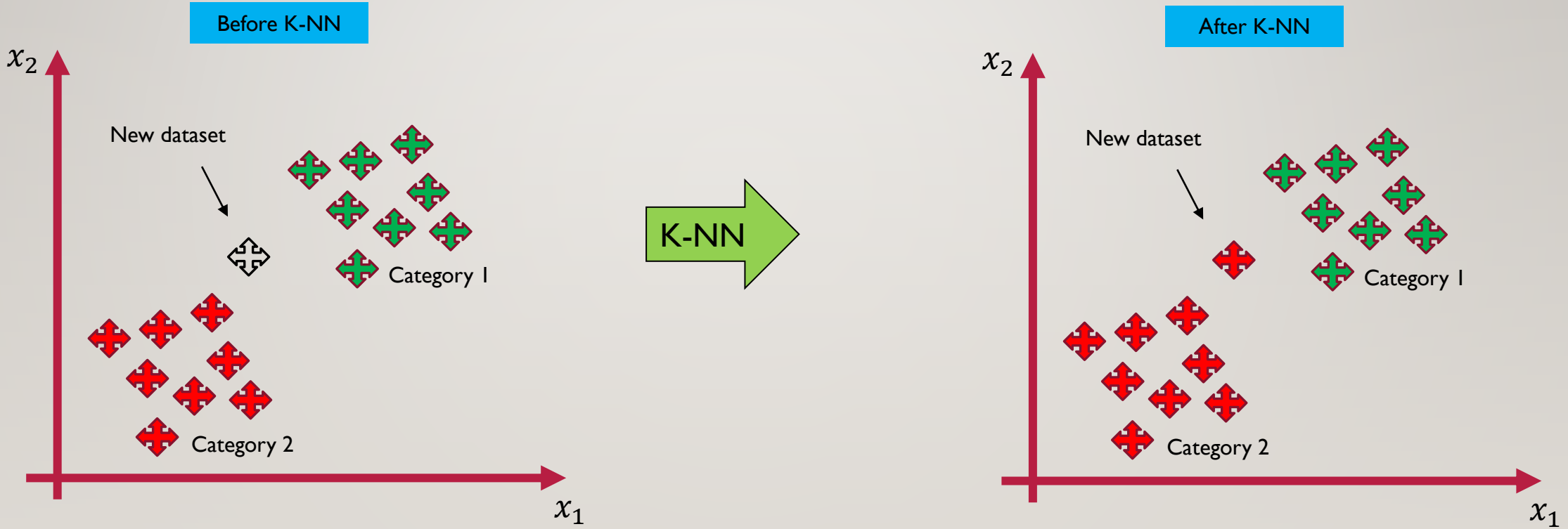
K-NN DEFINITION (2/2)

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

DISTANCE METRICS

- Minkowski Distance
- Euclidean Distance
- Manhattan Distance

K-NN INTUITION



MINKOWSKI DISTANCE

- Minkowski distance is a metric in Normed vector space.
- A Normed vector space is a vector space on which a norm is defined.
- Suppose X is a vector space then a norm on X is a real valued function ||*x*||which satisfies below conditions -
  - **Zero Vector-** Zero vector will have zero length.
  - **Scalar Factor-** The direction of vector doesn't change when you multiply it with a positive number though its length will be changed.
  - **Triangle Inequality-** If distance is a norm then the calculated distance between two points will always be a straight line.

DISTANCE FORMULA

The distance can be calculated using below formula -

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

P VALUE VARIANTS

$p = 1$, *Manhattan Distance*

$p = 2$, *Euclidean Distance*

$p = \infty$, *Chebychev Distance*

HOW DOES IT WORKS?

STEP 1: Choose the number K of neighbors.

STEP 2: Take the K nearest neighbors of the new data point, according to the Euclidean distance.

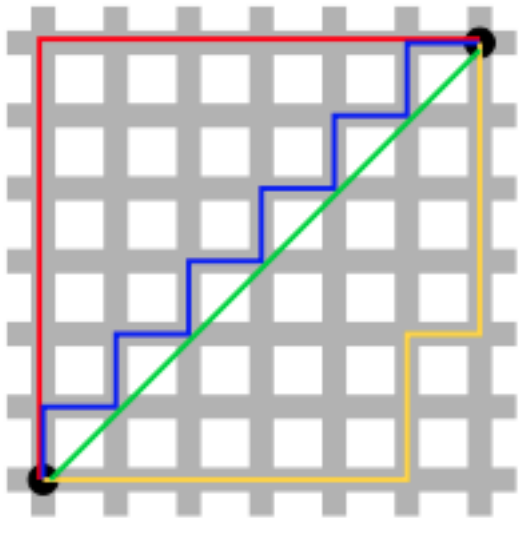
STEP 3: Among these k neighbors, count the number of data points in each category.

STEP 4: Assign the new data point to the category where you counted the most neighbors.

Your model is ready

MANHATTAN DISTANCE

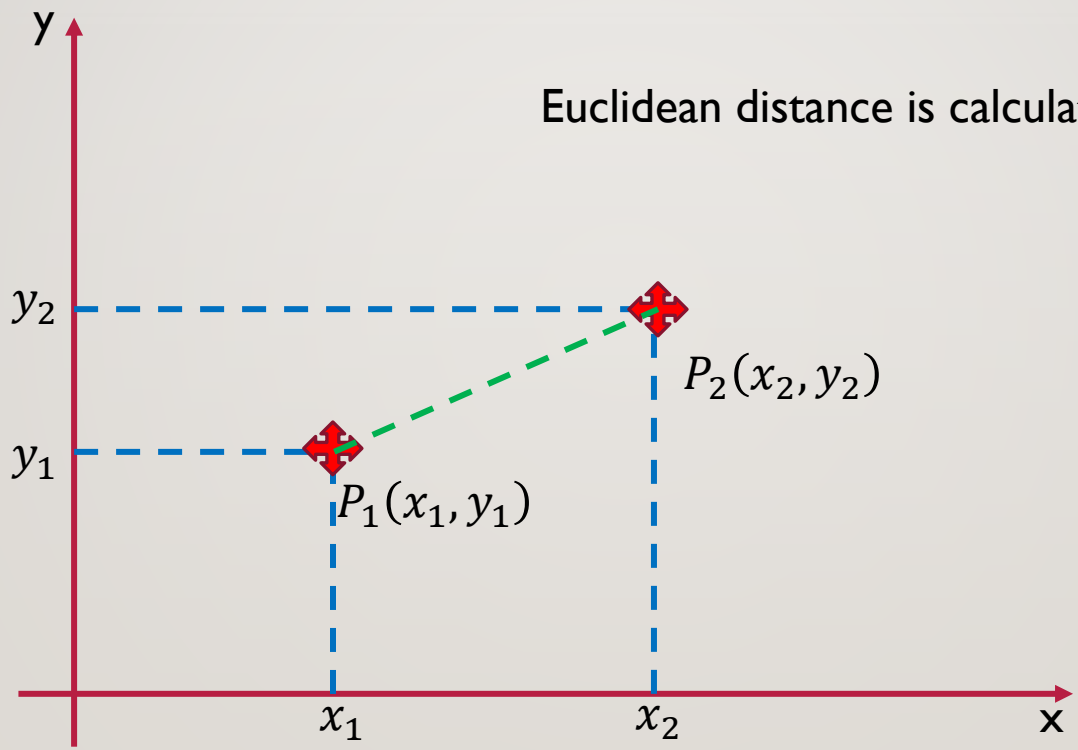
$$d = \sum_{i=1}^{n} |x_i - y_i|$$



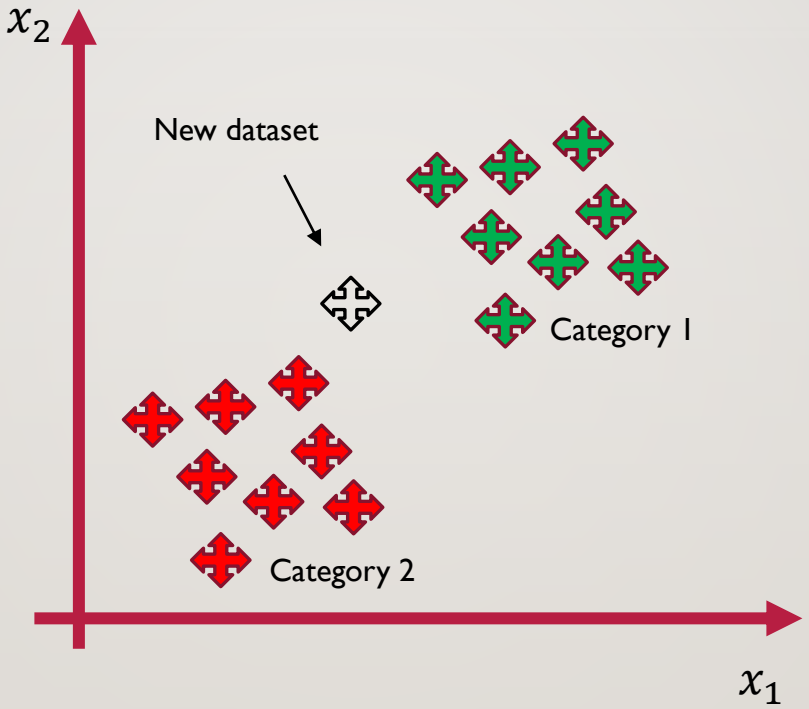
Source — Taxicab geometry Wikipedia

EUCLIDEAN DISTANCE

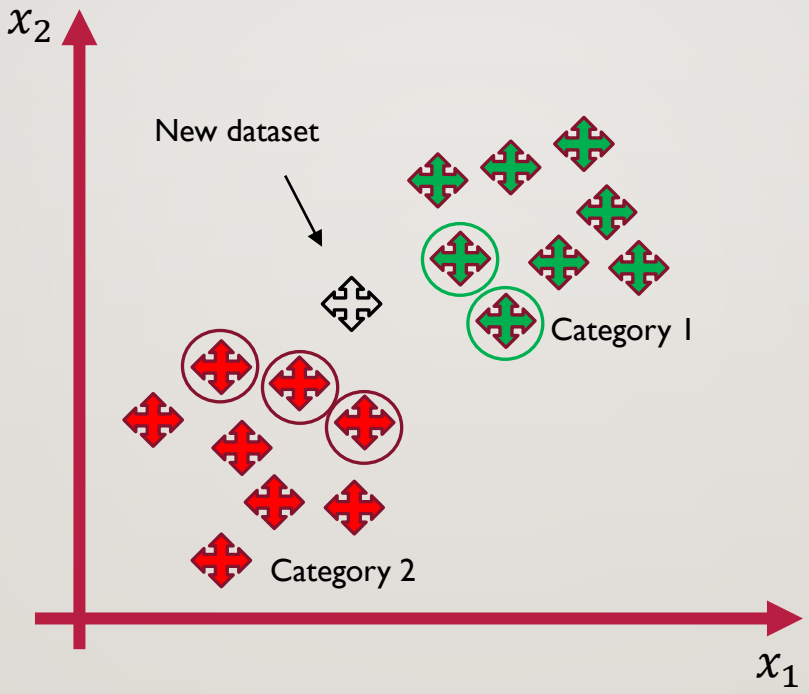
Euclidean distance is calculated as: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



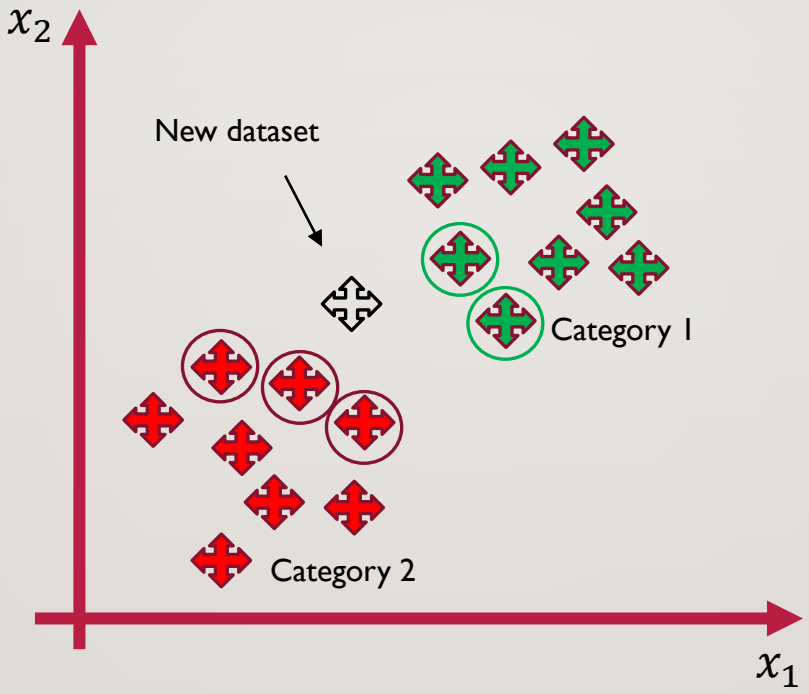
STEP 1: CHOOSE THE NUMBER K OF NEIGHBOURS: K=5



STEP 1: TAKE K=5 NEAREST NEIGHBOURS OF THE NEW DATA POINT ACCORDING TO THE EUCLIDEAN DISTANCE

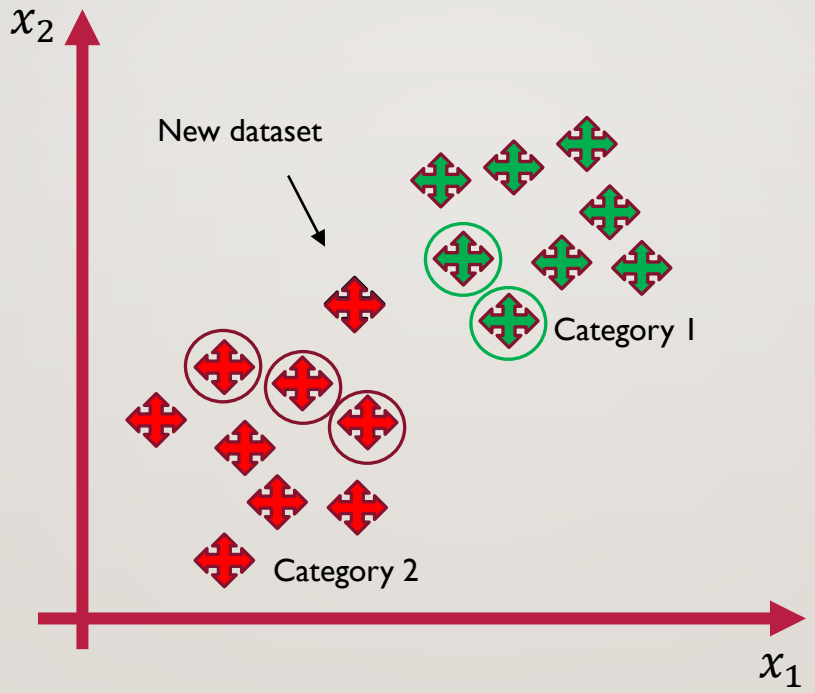


STEP 2: AMONG THESE NEIGHBORS COUNT THE NUMBER OF NEIGHBOURS IN EACH CATEGORY



Category 1: 2 Neighbors
Category 2: 3 neighbors

STEP 3: ASSIGN THE NEW DATASET TO THE CATEGORY WHERE YOU COUNTED THE MOST NEIGHBORS



Category 1: 2 Neighbors
Category 2: 3 neighbors

MODEL IS READY

HOW TO FIND THE IDEAL K ?

- Using odd numbers, fit a KNN classifier for each number.
- Create predictions.
- Further evaluate the performance using the predictions produced in step 2.
- Compare results across each model and decide on the one with the least error.

ASSUMPTIONS

- KNN assumes that the data is in a **feature space**.
- The **KNN** algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. "Birds of a feather flock together."
- Each of the training data consists of a set of vectors and class label associated with each vector. In the simplest case , it will be either + or – (for positive or negative classes). But KNN , can work equally well with arbitrary number of classes.
- We are also given a single number "k" .

ADVANTAGES

- The algorithm is **versatile**- It can be used for classification, regression, and search.
- **No Training Period**- KNN modeling does not include training period as the data itself is a model which will be the reference for future prediction and because of this it is very time efficient in term of improvising for a random modeling on the available data.
- **Easy Implementation**- KNN is very easy to implement as the only thing to be calculated is the distance between different points on the basis of data of different features and this distance can easily be calculated using distance formula such as- Euclidian or Manhattan
- As there is no training period thus new data can be added at any time since it wont affect the model.
- There's no need to build a model, tune several parameters, or make additional assumptions.

DISADVANTAGES

- **Does not work well with large dataset** as calculating distances between each data instance would be very costly.
- The algorithm gets sig**nificantly slower** as the number of examples and/or predictors/independent variables increase.
- **Does not work well with high dimensionality** as this will complicate the distance calculating process to calculate distance for each dimension.
- **Sensitive to noisy and missing data**
- **Feature Scaling-** Data in all the dimension should be scaled (normalized and standardized) properly .

APPLICATIONS

- Recommender Systems
- Text mining
- Agriculture
- Finance
- Medical
- Facial recognition

THANK YOU