Week4: K-Nearest Neighbors (KNN)

Shayan Dadman, PhD candidate
UiT, Narvik

What is KNN?

- KNN is a supervised machine learning algorithm that relies on labeled data.
- It can be used for both classification and regression problems.
- It is mostly considered due to its ease of interpretation and low calculation time.

What is KNN?

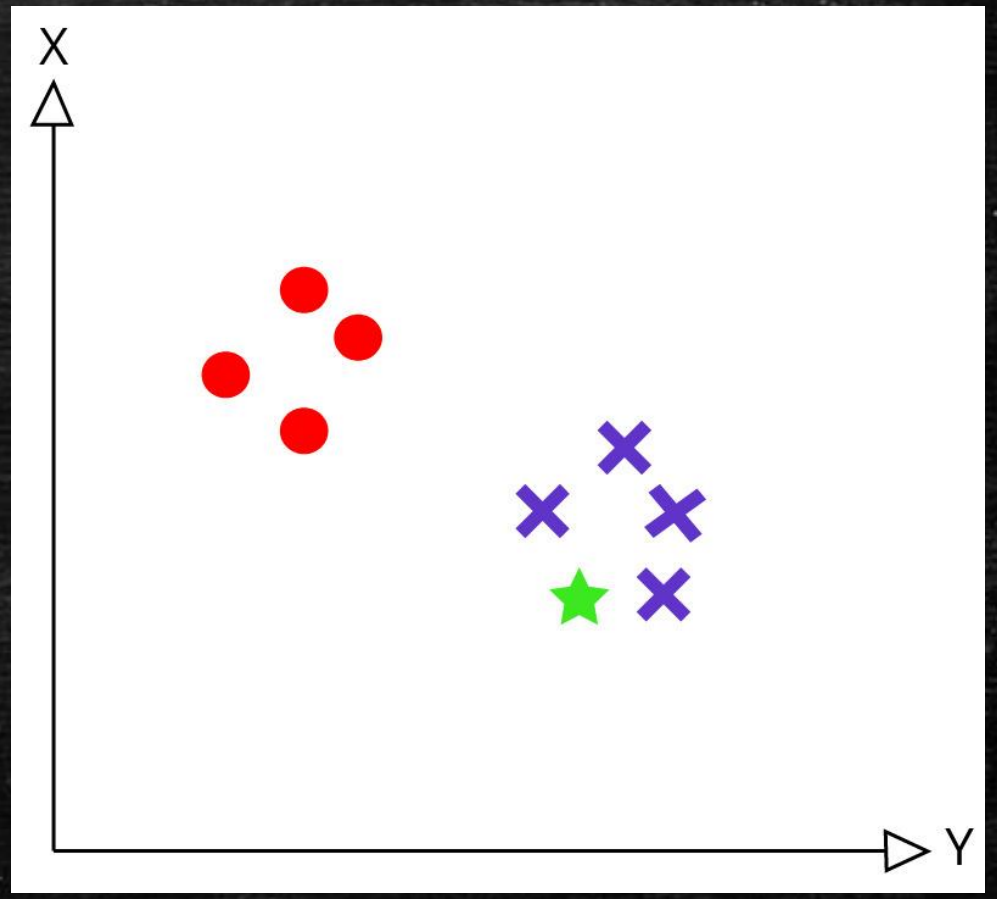
- The KNN algorithm assumes that similar values are placed close to each other.
- *K* in here is defined as the number of nearest neighbors.
- The KNN exhibits the idea of similarity by calculating the distance between the data points.
- Indeed, KNN considers *K* number of nearest neighbors to predict the class or continous value for a new data point.

What is KNN?

- The KNN algorithm has these characteristics:
  - It is instance-based learning, which uses the entire training examples to predict output for unseen data.
  - It uses a lazy learning method, in which the generalization of the training examples is postponed to a time when prediction is demanded on the new example.
  - It is non-parametric as it has no predefined form of the mapping function.

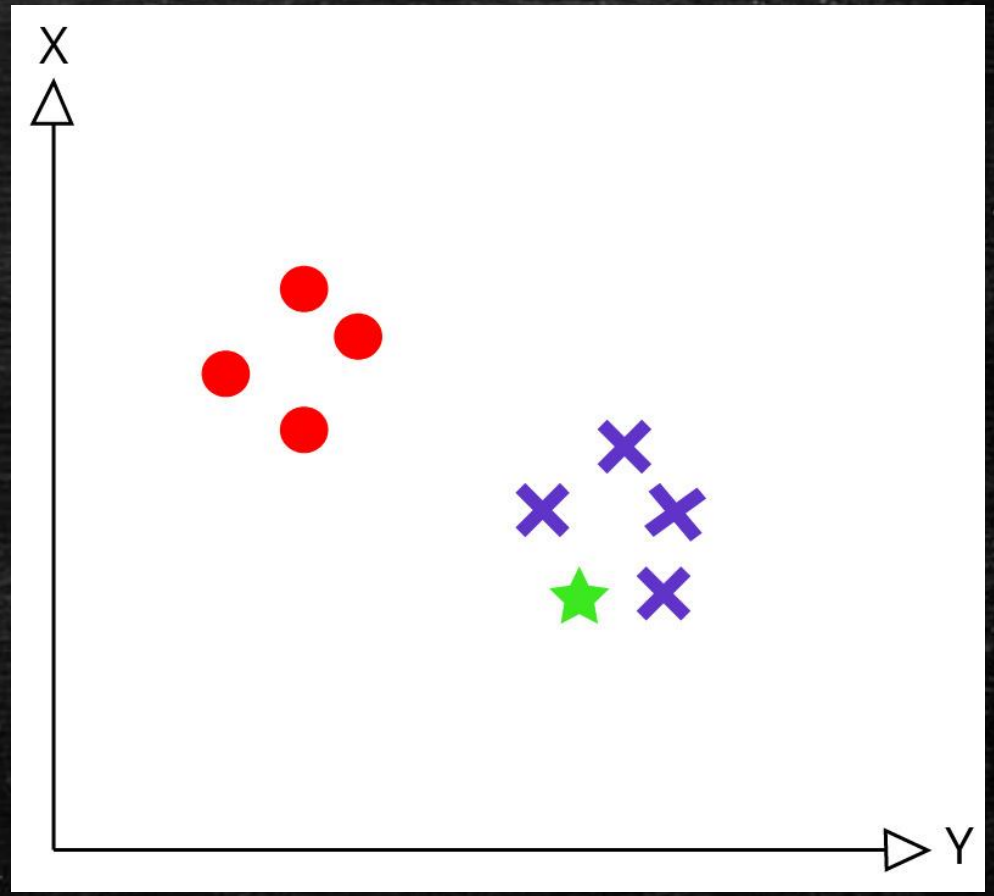
How does the algorithm work?

- In the following figure, we have a plot of the data points from our two dimesional feature space dataset.
- As we can see, we have a total of 8 data points, consist of 4 red and 4 purple.
- Red data points belong to **class1** and purple data points belong to **class2**.
- Green data point represents the new point, which a class is to be predicted.



How does the algorithm work?

- Clearly, it belongs to **class2** (purple points).
- Why? because its nearest neighbors are those data points that have minimum distance.



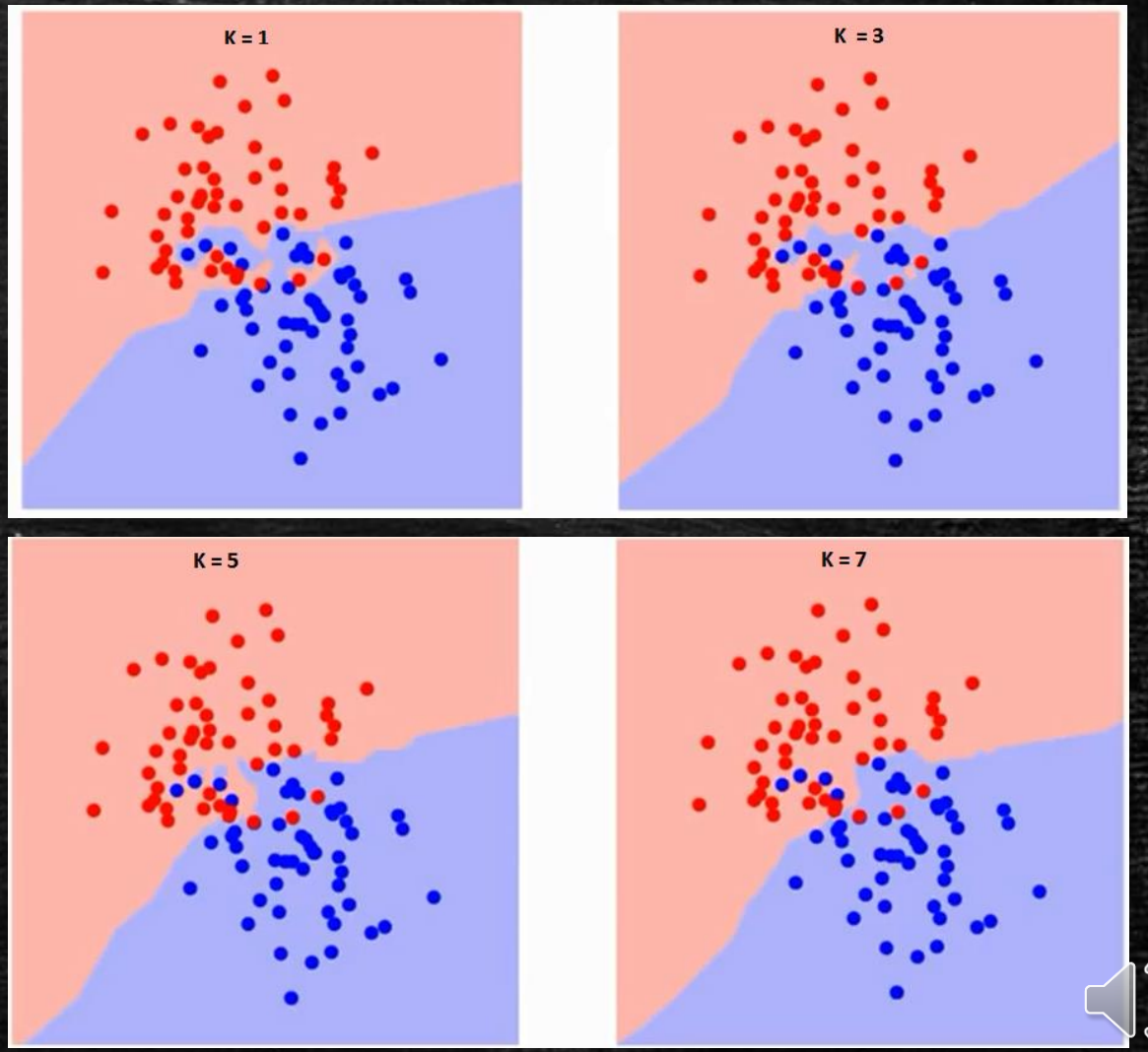
How does the algorithm work?

- They are two essential factors when using the KNN algorithm:
  - Distance metric
  - *K* value

- Euclidean distance is the most common distance metric. Cosine, inverse document frequency, hamming distance, and manhattan distance can also be used based on the application.

How does the algorithm work?

- The *K* value is the number of data points that we consider when calculating the minimum distance.
- In the following figures, you can see different *K* values used to separate two classes.
- As it can be seen, by increasing the *K* values, the boundary becomes smoother.
- In fact, increasing the *K* to infinity results in all blue or all red depending on the total majority.



KNN pseudo code

1. Load the data
2. Initialize the value of *K*
3. For each example in the data:
    1. Calculate the distance between test data and each row of training data.
    2. Add the distance and the index of the example to an ordered collection
4. Sort the calculated distances in ascending order based on distance values
5. Get top *K* entries from the sorted array
6. Get the labels of the selected *K* entries
7. Return the predicted class

Questions

1. Email me at Shayan.Dadman@uit.no
2. My office D3430