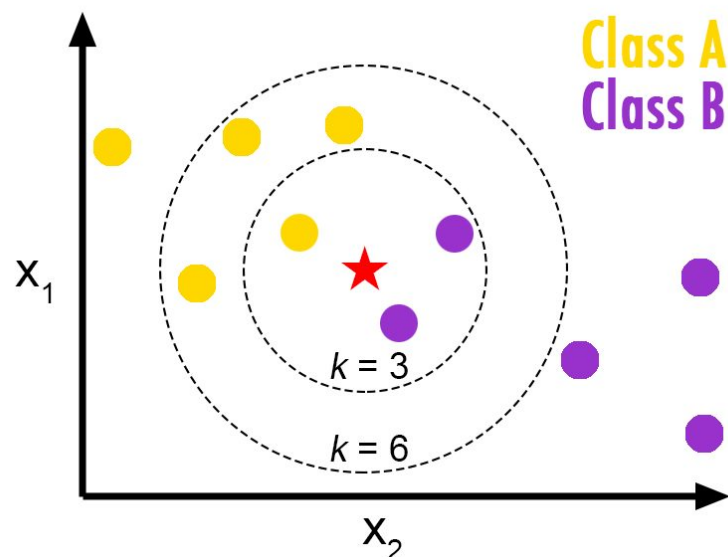


KNN

KNN performs classification or regression tasks for new data by calculating the distance between the new example and all the existing examples in the dataset. But how?

Here's the secret: The algorithm stores the entire dataset and classifies each new data point based on the existing data points that are similar to it. KNN makes predictions based on the training or known data only.

After the user defines a distance function, like the ones we mentioned earlier, KNN calculates the distance between data points in order to find the closest data points from our training data for any new data point. The existing data points closest to the new data point using the defined distance will become the k-neighbors. For a classification task, KNN will use the most frequent of all values from the k-neighbors to predict the new data label. For a regression task, the algorithm will use the average of all values from the k-neighbors to predict the new data value.



Why do we need a KNN algorithm?

(K-NN) algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation. It does not require any assumptions about the underlying data distribution. It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks. It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms.

The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors. This approach allows the algorithm to adapt to different patterns and make predictions based on the local structure of the data.