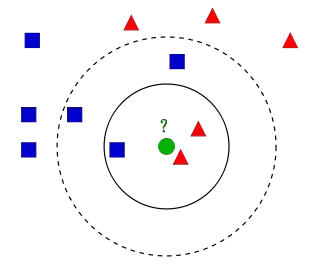
**Machine Learning Algorithms**

**K-Nearest Neighbours (KNN Classifier)**

Step1, Get the K points that are closest to your question point (min Euclidean distance).

Step 2, Make a vote from the closest neighbours.

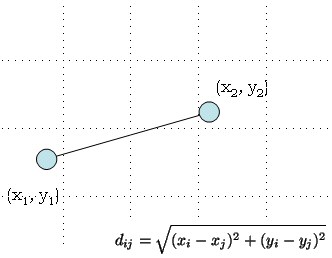
This is a KNN for classification example:



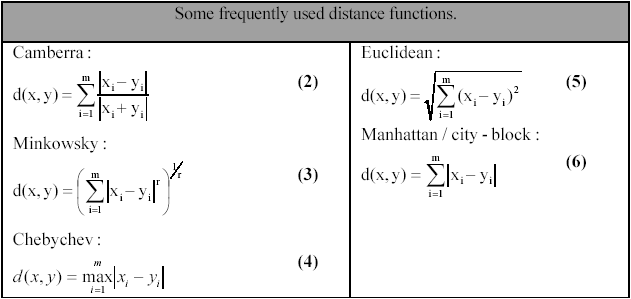
* **If k = 3**(solid line circle) it is assigned to the **red** class because there are **2 triangles and only 1 square** inside the inner circle.
* **If k = 5** (dashed line circle) it is assigned to the **blue** class (**3 squares vs. 2 triangles** inside the outer circle).

We assign a label to the new data based on the labels of the training observations which are most similar (or near) to it.

When this algorithm is faced by a test sample, it plots that sample in the same n-dimensional space as the training data and then searches for its **k** nearest neighbors based on any of these[**distance measures**](https://www2.ia-engineers.org/iciae/index.php/iciae/iciae2015/paper/viewFile/572/423) from the training samples. Yes, **it goes through the complete training data every time it needs to predict a test sample’s classification label!**

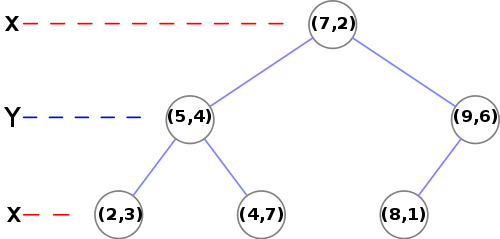


**Distance Measures**



**n\_neighbors**(int): number of nearest neighbors to be brought under consideration, denoted by **k**. By default, its value is 5. Higher value can lead to overfitting.

**algorithm** : the searching algorithm you plan to deploy — auto (default, looks through the training data and selects one algorithm on its own), kd\_tree (uses **KDTree**), ball\_tree (uses **BallTree**) and brute (uses [***brute force search***](https://en.wikipedia.org/wiki/Brute-force_search)).



**DRAWBACKS**

* Trainingtime for each sample
* Training data – numeric for distance calculation
* Poor performance on **imbalanced data**

k-NN algorithm can be used for imputing missing value of both categorical and continuous variables.

**Q. Can we use KNN for regression problems?**