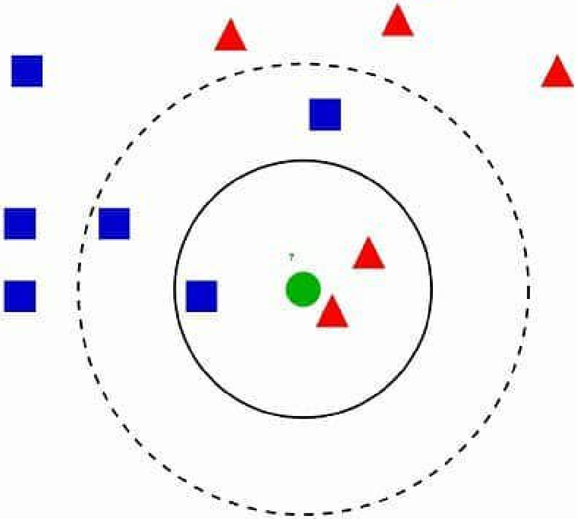
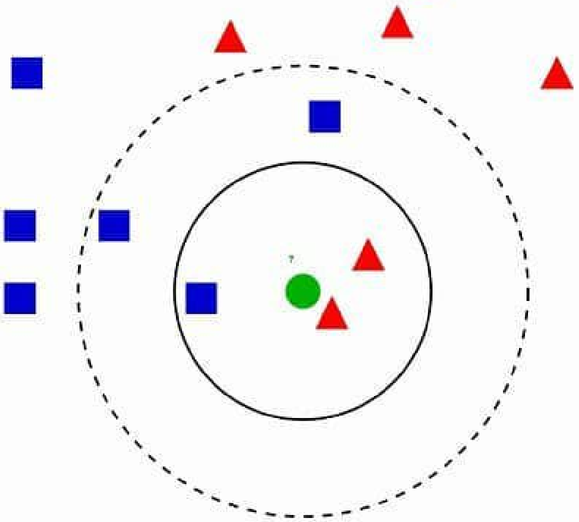
K Nearest Neighbors



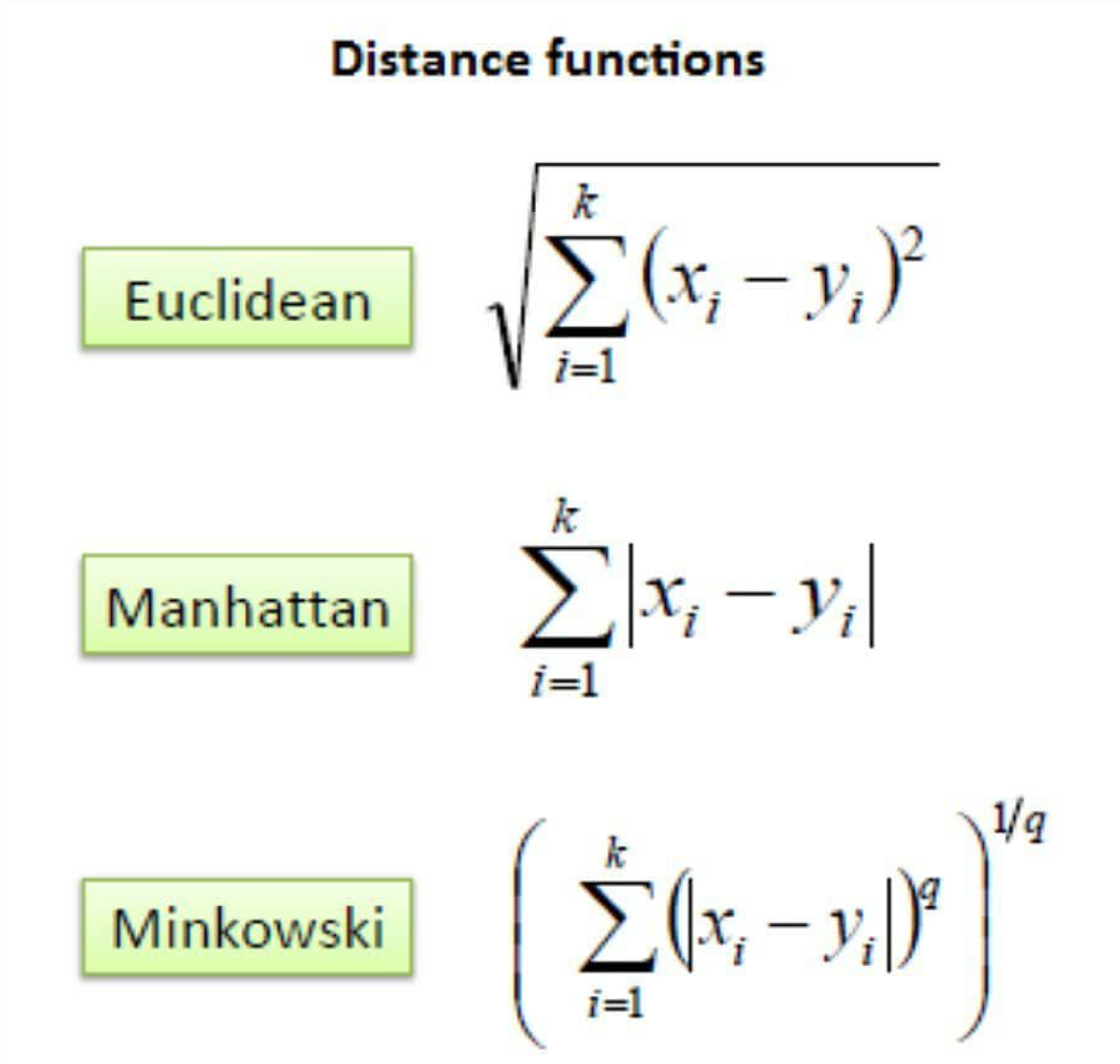
- The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement and can be used to solve both classification and regression problems.
- Let's take an example and learn this algorithm.
- You can see we have blue squares as one class and a red triangle as another class. And now we need to find the green circle belongs to which class?
- Here comes the K in KNN it is used to find the K nearest neighbors for that green circle and take the majority class label and assign it to the green circle.

K Nearest Neighbors



- Let's take K = 3 and you can see that we have 2 red triangles and 1 blue square as the nearest neighbors to green circle and we take the majority and assign green circle as a red triangle.
- When we take K = 5 then you can see it belongs to the blue square.
- So how to find the nearest neighbors?
- KNN works based on a similarity measure (e.g., distance functions)
- There are multiple distance functions like Euclidean, Minkowski, Manhattan etc..

K Nearest Neighbors



Distance functions

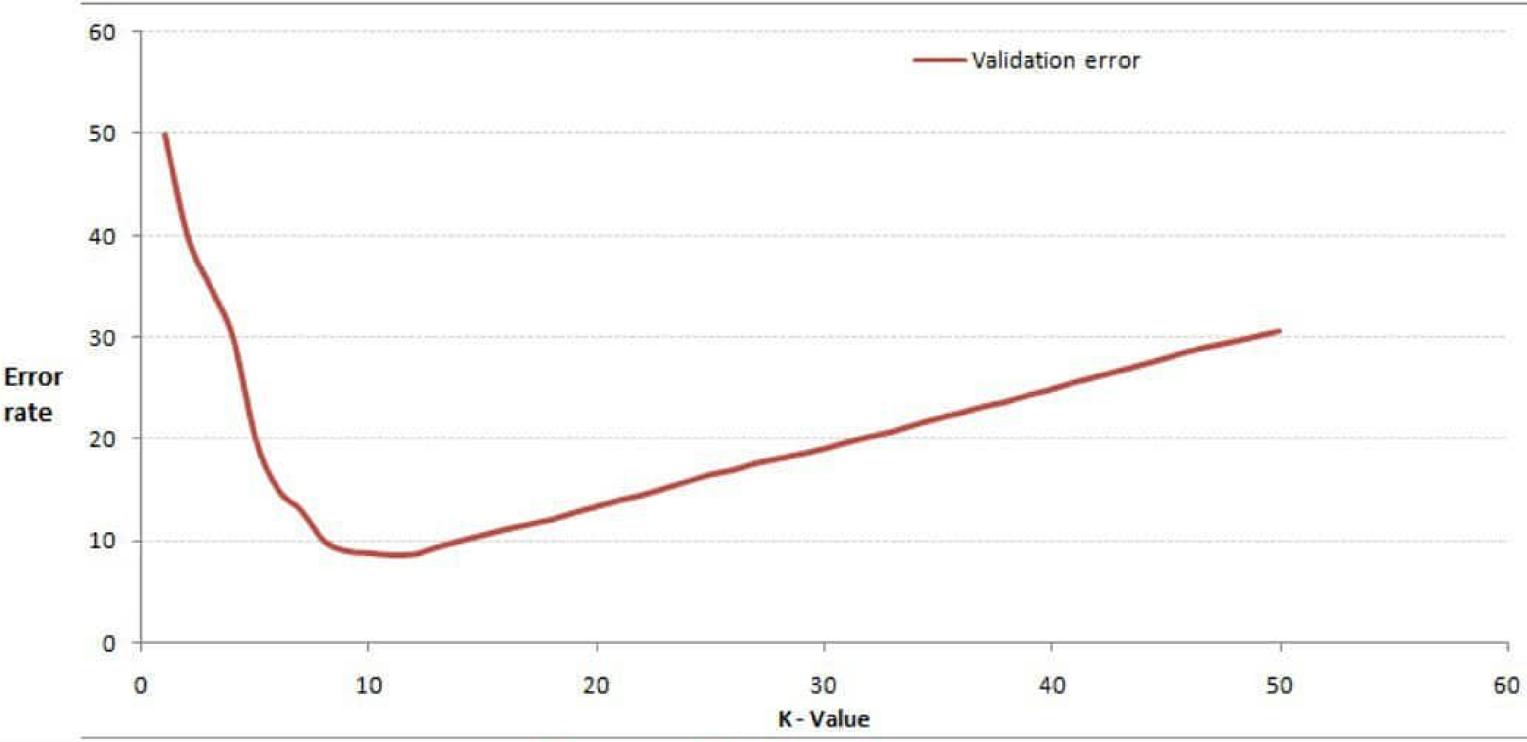
Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}\left|x_i - y_i\right|$$

Minkowski $$\left(\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^q\right)^{1/q}$$

K Nearest Neighbors

- So how to choose the right K?
- We can take the help of domain expert on the problem we are solving to get the best K.
- Or we can use Cross-validation to find the best K. we try with different K values and check how the validation error rate is varying. we choose the elbow point in the graph as best K. See below graph for visual experience.



@learn.machinelearning

K Nearest Neighbors

- Algorithm pseudo code.
- Load the data
- Initialize K to your chosen number of neighbors
- For each example in the data
    - Calculate the distance between the query point and all the test data.
    - Sort the calculated distances in ascending order based on distance values
    - Get top k rows from the sorted array
    - Get the most frequent class of these rows
    - Return the predicted class

In KNN there is no training Phase. we just take the test data and predict the class using the train data.

K Nearest Neighbors

Pros of KNN

- K-NN algorithm is very simple to understand and equally easy to implement.
- K-NN is a non-parametric algorithm which means there are no assumptions to be met to implement K-NN.
- K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data.
- K-NN can also be used for multiclass classification
- one of the biggest advantages of K-NN is that K-NN can be used both for classification and regression problems. classification.
- Given it’s an instance-based learning; k-NN is a memory-based approach. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use.

K Nearest Neighbors

Cons of KNN

- K-NN might be very easy to implement but as the dataset grows efficiency or speed of algorithm declines very fast.
- KNN works well with a small number of input variables but as the numbers of variables grow K-NN algorithm struggles to predict the output of new data point.(we also called it a Curse of Dimensionality)
- We need to have normalized data.
- k-NN doesn’t perform well on imbalanced data.
- K-NN algorithm is very sensitive to outliers as it simply chose the neighbors based on distance criteria.
- K-NN inherently has no capability of dealing with missing value problem.
- One of the biggest issues with K-NN is to choose the optimal number of neighbors to be considered while classifying the new data entry.