**How to calculate K Nearest Neighbor (KNN)?**

Suppose we have the height, weight and T-shirt size of some customers and we need to predict the T-shirt size of a new customer given only height and weight information we have. Data including height, weight and T-shirt size information is shown below -

| **Height (in cms)** | **Weight (in kgs)** | **T Shirt Size** |
| --- | --- | --- |
| 158 | 58 | M |
| 158 | 59 | M |
| 158 | 63 | M |
| 160 | 59 | M |
| 160 | 60 | M |
| 163 | 60 | M |
| 163 | 61 | M |
| 160 | 64 | L |
| 163 | 64 | L |
| 165 | 61 | L |
| 165 | 62 | L |
| 165 | 65 | L |
| 168 | 62 | L |
| 168 | 63 | L |
| 168 | 66 | L |
| 170 | 63 | L |
| 170 | 64 | L |
| 170 | 68 | L |

**Step 1 : Calculate Similarity based on distance function**

There are many distance functions but **Euclidean** is the most commonly used measure. It is mainly used when data is continuous.

**Here's the formula:** √(X₂-X₁)²+(Y₂-Y₁)²

The idea to use distance measure is to find the distance (similarity) between new sample and training cases and then find the k-closest customers to new customer in terms of height and weight. New customer named 'Monica' has height 161cm and weight 61kg. Euclidean distance between first observation and new observation (monica) is as follows -

d1 = √(161 - 158)² + (61 - 58)²

= √9 + 9

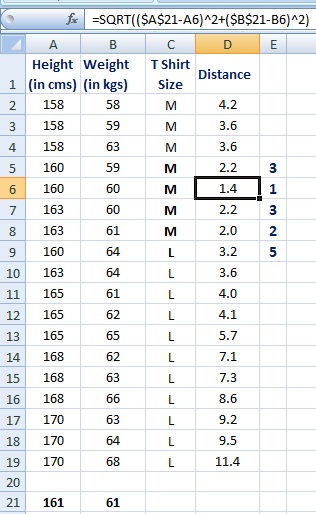
= √18

= 4.2

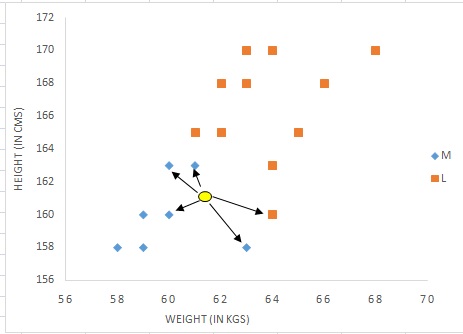
Similarly, we will calculate the distance of all the training cases with new cases and calculate the rank in terms of distance. The smallest distance value will be ranked 1 and considered as nearest neighbor.

**Step 2 : Find K-Nearest Neighbors**

Let k be 5. Then the algorithm searches for the 5 customers closest to Monica, i.e. most similar to Monica in terms of attributes, and see what categories those 5 customers were in. If 4 of them had ‘Medium T shirt sizes’ and 1 had 'Large T shirt size' then your best guess for Monica is ‘Medium T shirt. See the calculation shown in the snapshot below -



In the graph below, binary dependent variable (T-shirt size) is displayed in blue and orange color. 'Medium T-shirt size' is in blue color and 'Large T-shirt size' in orange color. New customer information is exhibited in a yellow circle. Four blue highlighted data points and one orange highlighted data point are close to the yellow circle. so the prediction for the new case is blue highlighted data point which is Medium T-shirt size.



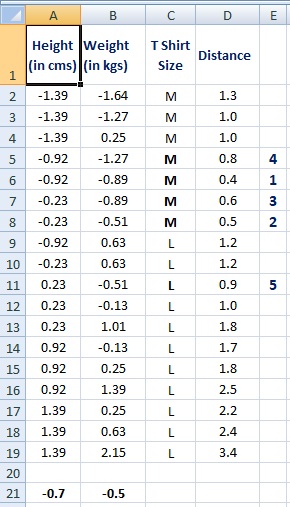
**Assumptions of KNN**

**1. Standardization**

When independent variables in training data are measured in different units, it is important to standardize variables before calculating distance. For example, if one variable is based on height in cms, and the other is based on weight in kgs then height will influence more on the distance calculation. In order to make them comparable we need to standardize them which can be done by any of the following methods :



After standardization, 5th closest value got changed as height was dominating earlier before standardization. Hence, it is important to standardize predictors before running the K-nearest neighbor algorithm.



**2. Outlier**

Low k-value is sensitive to outliers and a higher K-value is more resilient to outliers as it considers more voters to decide prediction.

**Why KNN is non-parametric?**

Non-parametric means not making any assumptions on the underlying data distribution. Non-parametric methods do not have fixed numbers of parameters in the model. Similarly in KNN, model parameters actually grow with the training data set - you can imagine each training case as a "parameter" in the model.

**Difference between KNN and K-mean**

Many people get confused between these two statistical techniques- K-mean and K-nearest neighbor. See the difference between them below -

K-mean is an unsupervised learning technique (no dependent variable) whereas KNN is a supervised learning algorithm (dependent variable exists)

K-mean is a clustering technique which tries to split data points into K-clusters such that the points in each cluster tend to be near each other whereas K-nearest neighbor tries to determine the classification of a point, combines the classification of the K nearest points

**How to find the best K value?**

Cross-validation is a smart way to find out the optimal K value. It estimates the validation error rate by holding out a subset of the training set from the model building process. Cross-validation (let's say 10 fold validation) involves randomly dividing the training set into 10 groups, or folds, of approximately equal size. 90% data is used to train the model and remaining 10% to validate it. The misclassification rate is then computed on the 10% validation data. This procedure repeats 10 times. Different groups of observations are treated as a validation set each of the 10 times. It results to 10 estimates of the validation error which are then averaged out.