K-Nearest Neighbour (KNN)

Supervised Learning technique

Assume the similarity between the new case/data and available cases and put the

new case into the category that is most similar to the available categories.

Stores all the available data and classifies a new data point based on the similarity - means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

Nearest Neighbor Decision Boundary

Assuming a Euclidean distance metric, the decision boundary between any two

training examples a and b is a straight line.

If a query point is located on the decision boundary, this means its equidistant from both training example a and b.

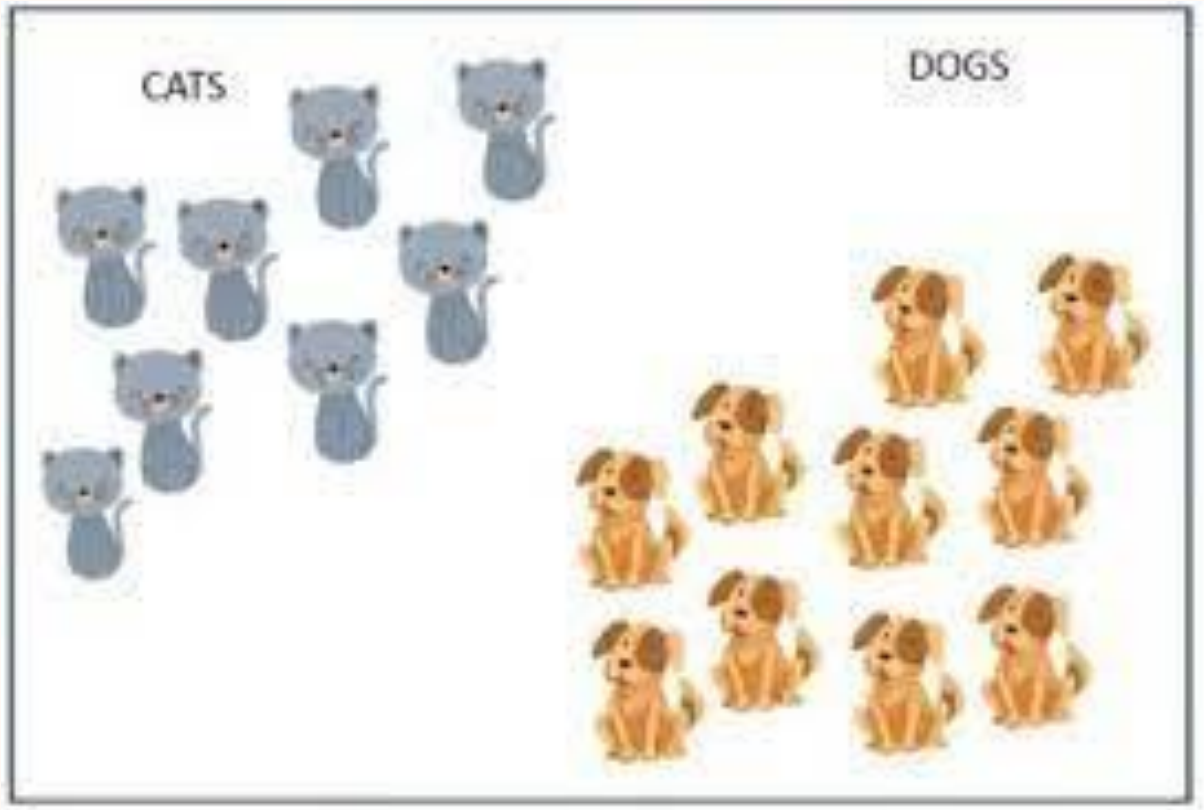
While the decision boundary between a pair of points is a straight line, the decision boundary of the NN model on a global level, considering the whole training set, is a set of connected, convex polyhedra.

All points within a polyhedron are closest to the training example inside, and all points outside the polyhedron are closer to a different training example..

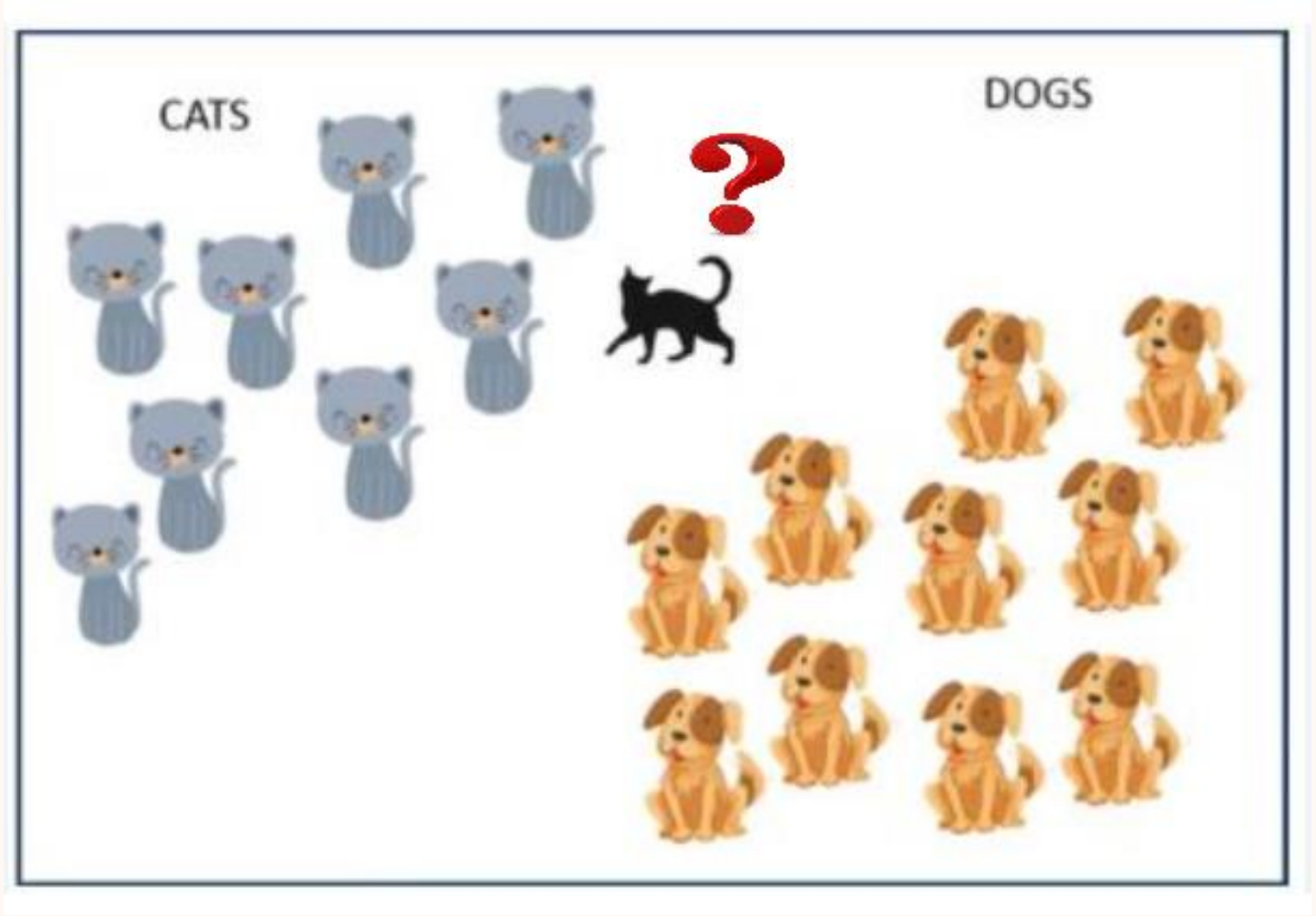
Example

CATS	DOGS
	
Sharp Claws, uses to climb	Dull Claws
Smaller length of ears	Bigger length of ears
Meows and purrs	Barks
Doesn't love to play around	Loves to run around

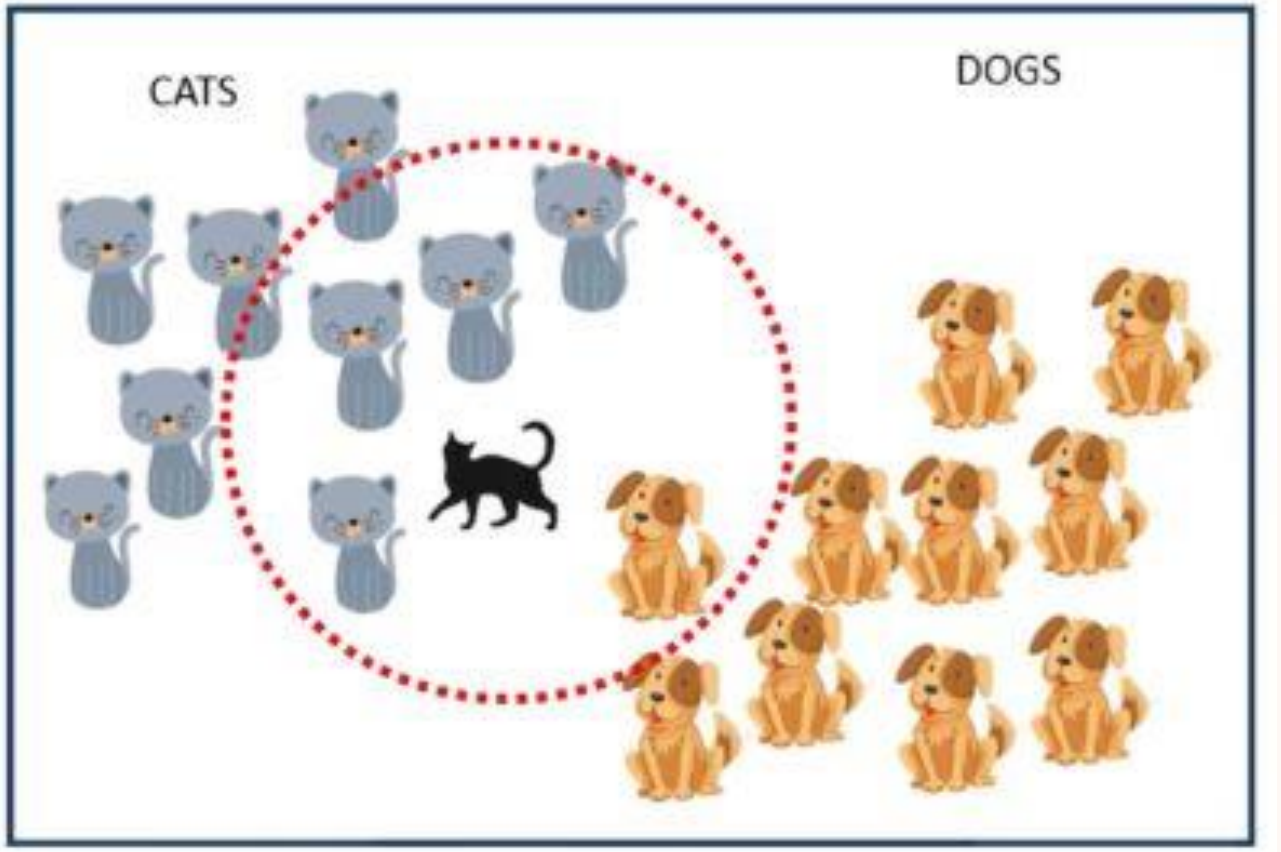
Example



Example



Example



KNN

KNN Algorithm

Step-1: Select the number K of the neighbors

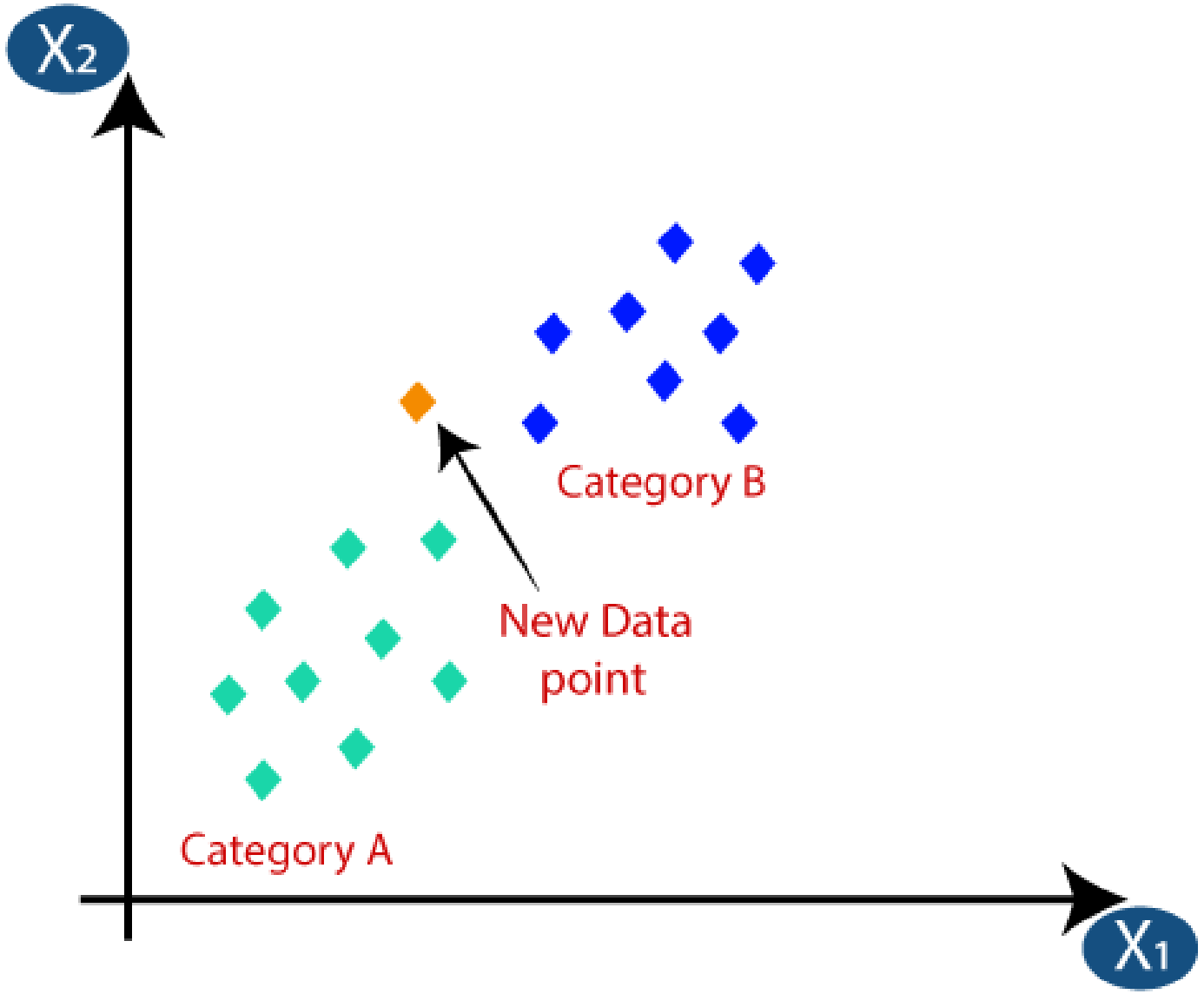
Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

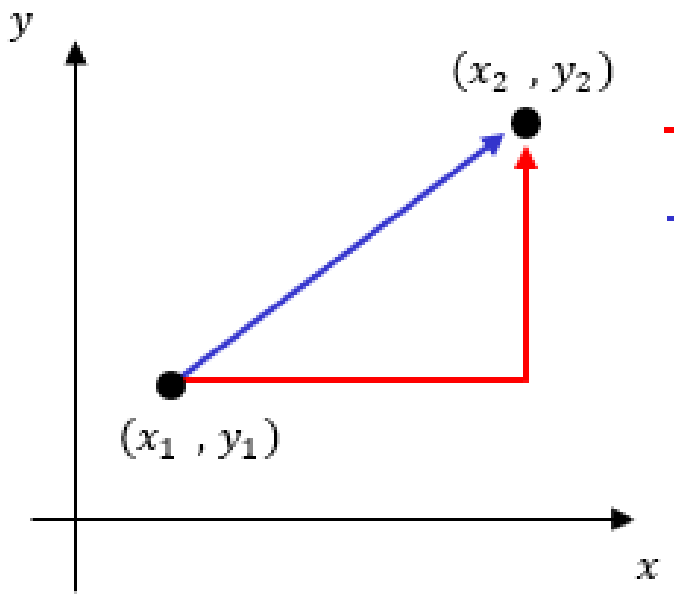
Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Example



calculate
the **Euclidean distance/Manhattan distance**



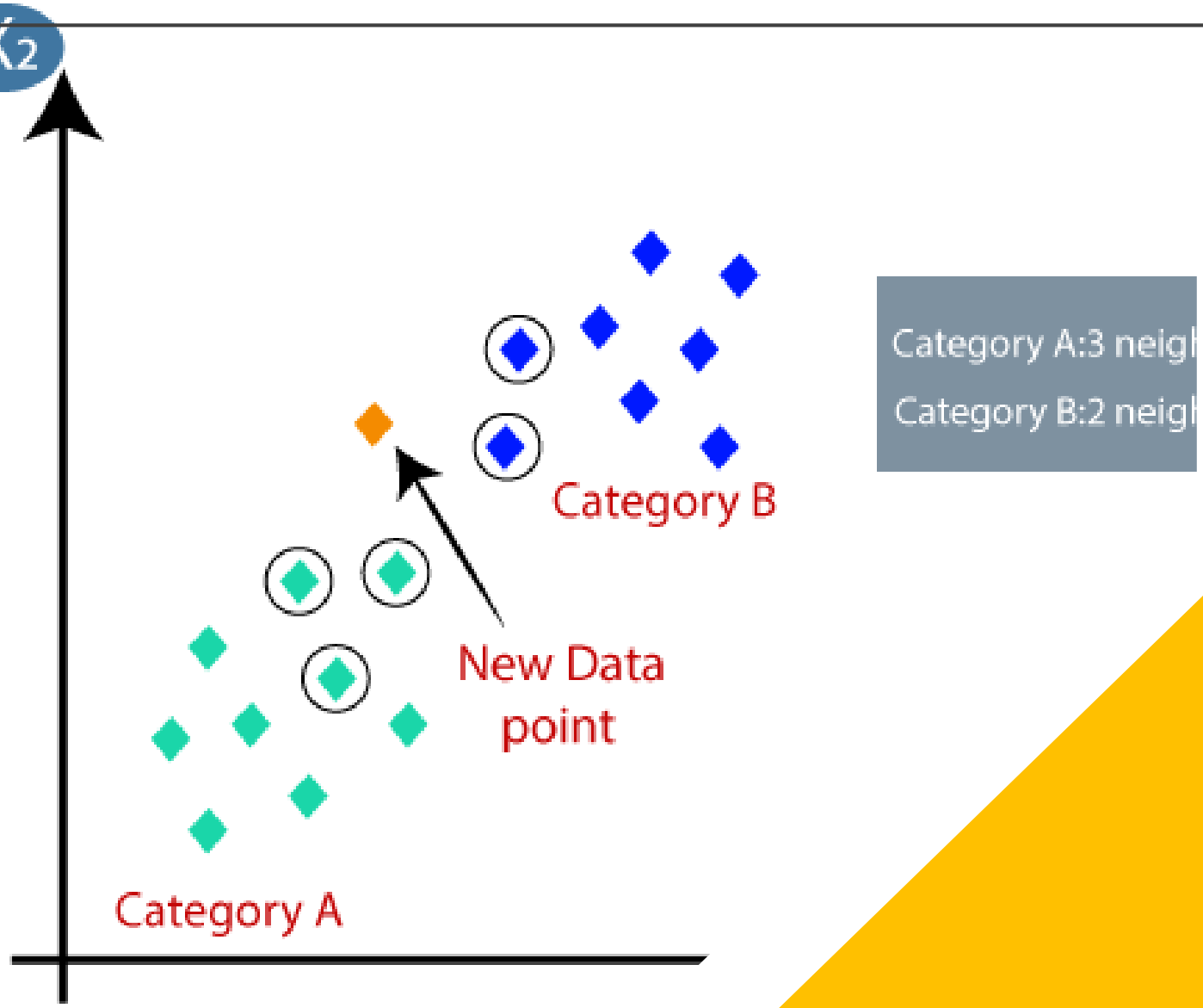
Manhattan Distance L^1

Euclidean Distance L^2

$$L^1 = |x_2 - x_1| + |y_2 - y_1|$$

$$L^2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Among these k neighbors, count the number of the data points in each category



Example

Suppose we have 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.

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

Lets choose K=5

Calculate Similarity based on distance function

- many distance functions exists: but Euclidean is the most commonly used measure - mainly used when data is continuous.
- Manhattan distance is also very common for continuous variables

- The idea to use distance measure is to find the distance (similarity) between new sample and training cases and then finds the k-closest customers to new customer in terms of height and weight.

Euclidean distance

Given two points A and B in d dimensional space such that $A = [a_1, a_2, \cdots, a_d]$ and $B = [b_1, b_2, \cdots, b_d]$, the Euclidean distance between A and B is defined as:

$$||A - B|| = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}$$

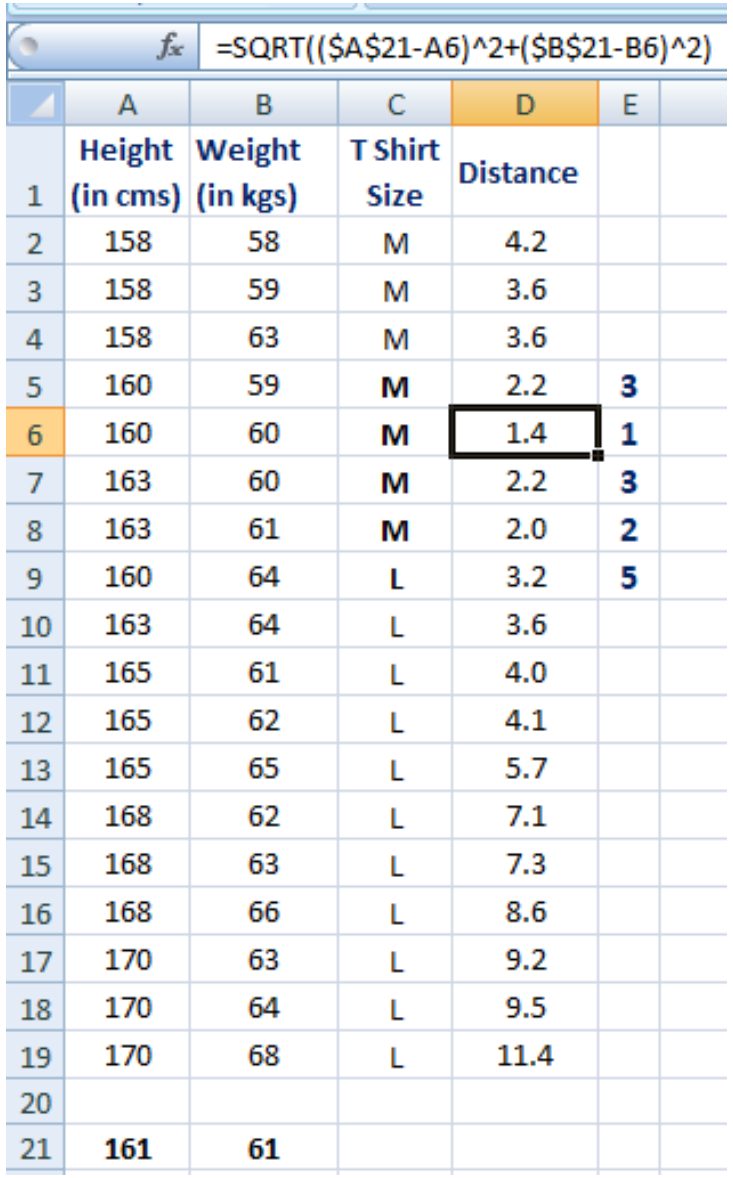
Manhattan distance

Given two random points A and B in d dimensional space such that $A = [a_1, a_2, \cdots, a_d]$ and $B = [b_1, b_2, \cdots, b_d]$, the Manhattan distance between A and B is defined as:

$$|A - B| = \sum_{i=1}^{d} |a_i - b_i|$$

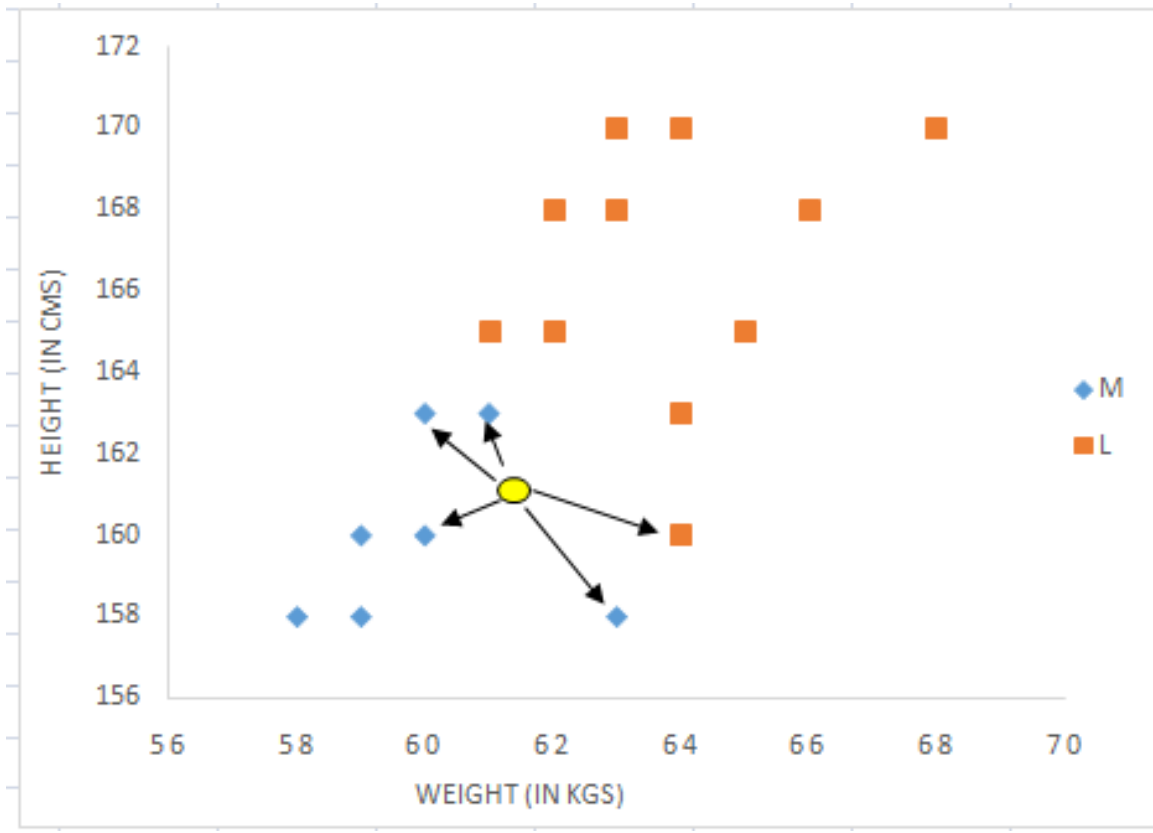
Calculate Similarity based on distance function

- **New customer named 'Monica' has height 161cm**
- **and weight 61kg.**
  - Euclidean distance between first observation and new observation is as follows -
- =SQRT((161-158)^2+(61-58)^2)
  - Similarly, we will calculate distance of all the training cases with new case and calculates the rank in terms of distance.
  - The smallest distance value will be ranked 1 and
- considered as nearest neighbor.



	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	158	58	M	4.2	
3	158	59	M	3.6	
4	158	63	M	3.6	
5	160	59	M	2.2	3
6	160	60	M	1.4	1
7	163	60	M	2.2	3
8	163	61	M	2.0	2
9	160	64	L	3.2	5
10	163	64	L	3.6	
11	165	61	L	4.0	
12	165	62	L	4.1	
13	165	65	L	5.7	
14	168	62	L	7.1	
15	168	63	L	7.3	
16	168	66	L	8.6	
17	170	63	L	9.2	
18	170	64	L	9.5	
19	170	68	L	11.4	
20					
21	161	61			

- In the graph, 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 yellow circle.
- Four blue highlighted data points and one orange highlighted data point are close to yellow circle.
- so the prediction for the new case is blue highlighted data point which is Medium T- shirt size.



How to choose the value of k?

There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5

Sqrt(n), where n is the number of data points

Odd value of k to avoid two classes of data

When do we use KNN?

- Data is labeled
- Data is noise free
- Dataset is small