**Importance of Distance Metrics in Machine Learning Modelling**

The distance metric helps algorithms to recognize similarities between the contents.

A distance function provides distance between the elements of a set. If the distance is zero then elements are equivalent else they are different from each other.

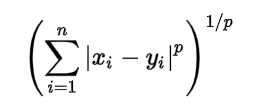
**Minkowski Distance (magnitude-based)**

A metric (or distance) and a norm are two different things. You can use a norm to define a metric, but not necessarily the other way around. The Minkowski metric is the metric induced by the LpLp norm, that is, the metric in which the distance between two vectors is the norm of their difference.

Minkowski distance is a metric in Normed vector space. A Normed vector space is a vector space on which a norm is defined (“in a space where distances can be represented as a vector that has a length.”). Suppose X is a vector space then a norm on X is a real valued function ||x||which satisfies below conditions -

1. **Zero Vector-**Zero vector will have zero length.
2. **Scalar Factor-** The direction of vector doesn’t change when you multiply it with a positive number though its length will be changed.
3. **Triangle Inequality-** If distance is a norm then the calculated distance between two points will always be a straight line.

The distance can be calculated using below formula -

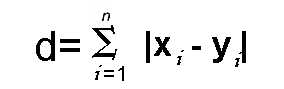


**Manhattan Distance:**

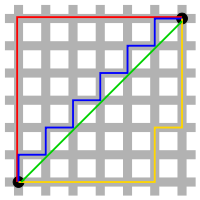
We use Manhattan Distance if we need to calculate the distance between two data points in a grid like path.

setting **p’s** value as **1**

Distance **d**will be calculated using an **absolute sum of difference**between its cartesian co-ordinates as below :

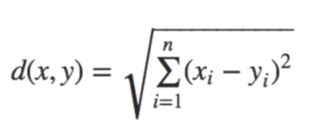


If you try to visualize the distance calculation, it will look something like as below :



**Euclidean Distance:**

It is calculated using Minkowski Distance formula by setting **p’s** value to **2**. This will update the distance **‘d’**formula as below :



Euclidean distance formula can be used to calculate the distance between two data points in a plane.

**Chebyshev distance:**

It is the extreme case of Minkowski distance. When we use infinity as the value of the parameter p, we end up with a metric that defines distance as the maximal absolute difference between coordinates:

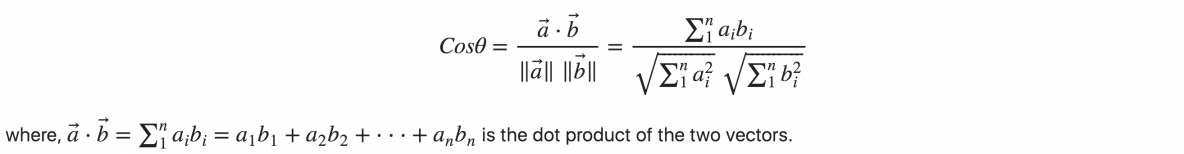
DChebyshev(x,y):=maxi(|xi−yi|)

Let’s start by proving that a map is a vector space. If we take a map, we see that distances between cities are normed vector space because we can draw a vector that connects two cities on the map. We can combine multiple vectors to create a route that connects more than two cities. When we can use a map of a city, we can give direction by telling people that they should walk/drive two city blocks North, then turn left and travel another three city blocks. In total they will travel five city blocks, that is the Manhattan distance between the starting point and their destination. When we draw another straight line that connects the starting point and the destination, we end up with a triangle. In this case, the distance between the points can be calculated using the Pythagorean theorem. In a warehouse, the distance between locations can be represented as Chebyshev distance if an overhead crane is used because the crane moves on both axes at the same time with the same speed.

**Cosine similarity:**

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. In this context, the two vectors I am talking about are arrays containing the word counts of two documents. When plotted on a multi-dimensional space, **where each dimension corresponds to a word in the document**, the cosine similarity captures the orientation (the angle) of the documents and not the magnitude. If you want the magnitude, compute the Euclidean distance instead.



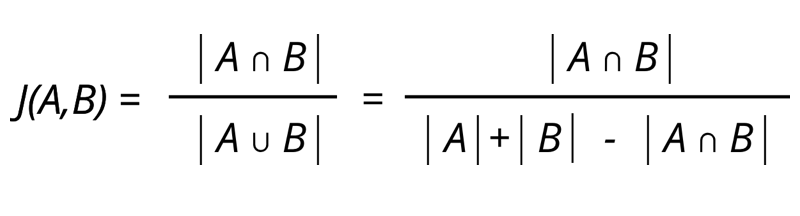
**Jaccard similarity:**

The Jaccard similarity index (sometimes called the Jaccard similarity coefficient) compares members for two sets to see which members are shared and which are distinct. It’s a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two populations. Although it’s easy to interpret, it is extremely sensitive to small samples sizes and may give erroneous results, especially with very small samples or data sets with missing observations.



[Source](https://en.wikipedia.org/wiki/Jaccard_index)

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used in understanding the similarities between sample sets. The measurement emphasizes similarity between finite sample sets, and is formally defined as the size of the intersection divided by the size of the union of the sample sets. The mathematical representation of the index is written as:



**This percentage tells you how similar the two sets are.**

* Two sets that share all members would be 100% similar. the closer to 100%, the more similarity (e.g. 90% is more similar than 89%).
* If they share no members, they are 0% similar.
* The midway point — 50% — means that the two sets share half of the members.

The Jaccard distance, is a measure of how dissimilar two sets are. It is the complement of the Jaccard index and can be found by subtracting the Jaccard Index from 100%.

In set notation, subtract from 1 for the Jaccard Distance:  
D(X,Y) = 1 – J(X,Y)

Convolutional Neural Networks, which are commonly tasked with image identification applications, apply the Jaccard Index measurements as a way of conceptualizing accuracy of object detection.