

# **Mahalanobis Distance**

## Definition :-

The Mahalanobis distance (MD) is **the distance between two points in multivariate space**. In a regular Euclidean space, variables (e.g. x, y, z) are represented by axes drawn at right angles to each other; The distance between any two points can be measured with a ruler. For uncorrelated variables, the Euclidean distance equals the MD. However, if two or more variables are correlated, the axes are no longer at right angles, and the measurements become impossible with a ruler. In addition, if you have more than three variables, you can't plot them in regular 3D space at all. The MD solves this measurement problem, as it measures distances between points, even correlated points for multiple variables.

## Uses :-

The most common use for the Mahalanobis distance is to find multivariate outliers, which indicates unusual combinations of two or more variables. For example, it's fairly common to find a 6' tall woman weighing 185 lbs, but it's rare to find a 4' tall woman who weighs that much.

## Formal Definition :-

The Mahalanobis distance between two objects is defined (Varmuza & Filzmoser, 2016, p.46) as:

$$d(\text{Mahalanobis}) = [(x_B - x_A)^T * \mathbf{C}^{-1} * (x_B - x_A)]^{0.5}$$

### **Where:**

$x_A$  and  $x_B$  is a pair of objects, and  
 $\mathbf{C}$  is the sample covariance matrix.

Another version of the formula, which uses distances from each observation to the central mean:

$$d_i = [(x_i - \bar{x})^T \mathbf{C}^{-1} (x_i - \bar{x})]^{0.5}$$

Where:

$x_i$  = an object vector  
 $\bar{x}$  = arithmetic mean vector

## Steps to integrate the code :-

- The train data set should be in csv format without headers. This train data file should be included in the working directory of the project.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1639000.3635	1744537.7775	6321.37												
2	1639000.3635	1744536.7775	6321.38												
3	1639001.3635	1744536.7775	6321.29												
4	1639000.3635	1744535.7775	6321.39												
5	1639001.3635	1744535.7775	6321.31												
6	1639002.3635	1744535.7775	6321.23												
7	1639000.3635	1744534.7775	6321.38												
8	1639001.3635	1744534.7775	6321.3												
9	1639002.3635	1744534.7775	6321.22												
10	1639000.3635	1744533.7775	6321.37												
11	1639001.3635	1744533.7775	6321.29												
12	1639002.3635	1744533.7775	6321.21												
13	1639003.3635	1744533.7775	6321.04												
14	1639000.3635	1744532.7775	6321.36												
15	1639001.3635	1744532.7775	6321.28												
16	1639002.3635	1744532.7775	6321.11												
17	1639003.3635	1744532.7775	6320.94												
18	1639004.3635	1744532.7775	6320.78												
19	1639000.3635	1744531.7775	6321.35												
20	1639001.3635	1744531.7775	6321.18												
21	1639002.3635	1744531.7775	6321.02												
22	1639003.3635	1744531.7775	6320.85												
23	1639004.3635	1744531.7775	6320.68												
24	1639005.3635	1744531.7775	6320.64												
25	1639000.3635	1744530.7775	6321.29												
26	1639001.3635	1744530.7775	6321.13												
27	1639002.3635	1744530.7775	6320.96												
28	1639003.3635	1744530.7775	6320.79												
29	1639004.3635	1744530.7775	6320.71												
30	1639005.3635	1744530.7775	6320.67												
31	1639000.3635	1744529.7775	6321.23												
32	1639001.3635	1744529.7775	6321.07												

- To measure the mahalanobis distance, create an object of the MahalanobisDistance Class passing the file name to the constructor.
- The MahalanobisDistance Class provides two functionalities :-
  - getDissimilarityMeasure(double point1[],double point2[])
  - getDistance(double point[])
 The vector of points need to be passed to the required functionality.

Sample example for 3D points:-

```
import java.util.Scanner;

public class Project {

    public static void main(String[] args) {

        int choice;
        MahalanobisDistance mahalanobisDistance = new
        MahalanobisDistance(args[0]);
        System.out.println("File = "+ args[0]);
        System.out.println("Enter 1 for Distance and 2 for
        DissimilarityMeasure");

        Scanner scan = new Scanner(System.in);
        choice =scan.nextInt();
        if(choice==1)
        {
            System.out.println("Enter point x");
            double x[]= new double[3];
```

```

        for(int i=0;i<3;i++)
        {
            x[i]= scan.nextDouble();
        }
        System.out.println("Distance =
"+mahalanobisDistance.getDistance(x));
    }
    else
    {
        double x[]= new double[3];
        double y[]= new double[3];
        System.out.println("Enter point x");
        for(int i=0;i<3;i++)
        {
            x[i]= scan.nextDouble();
        }
        System.out.println("Enter point y");

        for(int i=0;i<3;i++)
        {
            y[i]= scan.nextDouble();
        }
        System.out.println("DissimilarityMeasure =
"+mahalanobisDistance.getDissimilarityMeasure(x, y));

    }
    scan.close();

}

}

```