Min Max Normalization of data in data mining

Min Max normalization of Data Mining?

Min Max is a data normalization technique like **Z score**, **decimal scaling**, and normalization with **standard deviation**. It helps to normalize the data. It will scale the data between 0 and 1. This normalization helps us to understand the data easily.

For example, if I say you to tell me the difference between 200 and 1000 then it's a little bit confusing as compared to when I ask you to tell me the difference between 0.2 and 1.

How to normalize the data through the min-max normalization technique?

marks
8
10
15
20

Min:

The minimum value of the given attribute. Here Min is **8**

Max:

The maximum value of the given attribute. Here Max is **20**

V: V is the respective value of the attribute. For example here V1=8, V2=10, V3=15, and V4=20

newMax:

1

newMin:

0

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A$$

For Marks as 8:

$$MinMax = \frac{(V - Min\ marks)}{Max\ omarks - Min\ marks}(newMax - newMin)^{\square} + newMin$$

$$MinMax = \frac{(8 - 8)}{20 - 8} * (1 - 0)^{\square} + 0$$

$$MinMax = \frac{(0)}{12} * 1$$

$$MinMax = 0$$

For Marks as 10:

$$MinMax = \frac{(10 - 8)}{20 - 8} * (1 - 0)^{\square} + 0$$

$$MinMax = \frac{(2)}{12} * 1$$

$$MinMax = 0.16$$

For Marks as 15:

$$MinMax = \frac{(15 - 8)}{20 - 8} * (1 - 0)^{\square} + 0$$

$$MinMax = \frac{(7)}{12} * 1$$

$$MinMax = \mathbf{0.58}$$

For Marks as 20:

$$MinMax = \frac{(20 - 8)}{20 - 8} * (1 - 0)^{\square} + 0$$

$$MinMax = \frac{(12)}{12} * 1$$

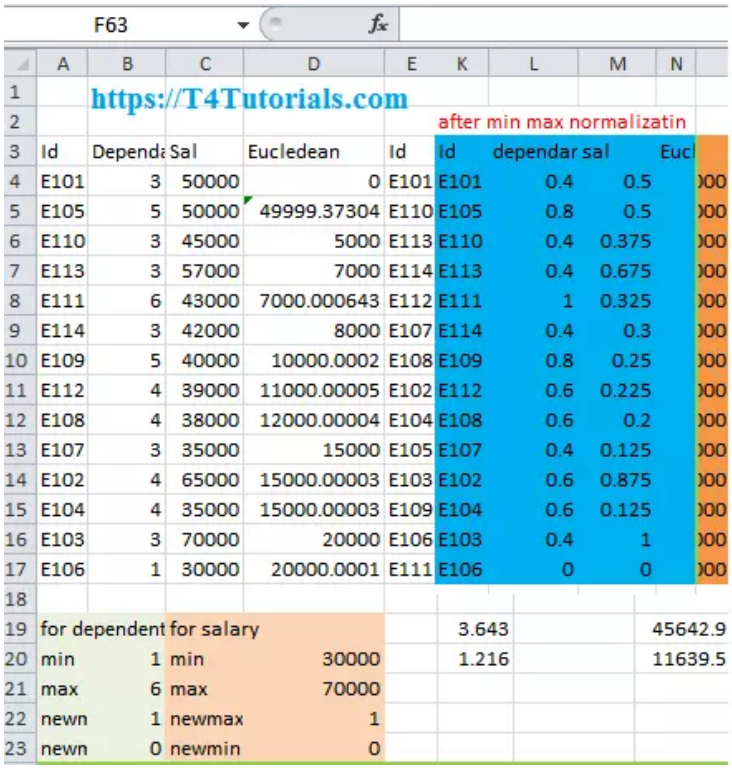
$$MinMax = 1$$

Figure: min-max normalization scaling

marks	marks after Min-Max normalization

8	0
10	0.16
15	0.58
20	1

Download Excel File Calculations



F63

	A	B	C	D	E	K	L	M	N
1		https://T4Tutorials.com							
2						after min max normalizatin			
3	Id	Depend	Sal	Eucledean	Id	Id	dependar	sal	Eucl
4	E101	3	50000	0	E101	E101	0.4	0.5	)00
5	E105	5	50000	49999.37304	E110	E105	0.8	0.5	)00
6	E110	3	45000	5000	E113	E110	0.4	0.375	)00
7	E113	3	57000	7000	E114	E113	0.4	0.675	)00
8	E111	6	43000	7000.000643	E112	E111	1	0.325	)00
9	E114	3	42000	8000	E107	E114	0.4	0.3	)00
10	E109	5	40000	10000.0002	E108	E109	0.8	0.25	)00
11	E112	4	39000	11000.00005	E102	E112	0.6	0.225	)00
12	E108	4	38000	12000.00004	E104	E108	0.6	0.2	)00
13	E107	3	35000	15000	E105	E107	0.4	0.125	)00
14	E102	4	65000	15000.00003	E103	E102	0.6	0.875	)00
15	E104	4	35000	15000.00003	E109	E104	0.6	0.125	)00
16	E103	3	70000	20000	E106	E103	0.4	1	)00
17	E106	1	30000	20000.0001	E111	E106	0	0	)00
18									
19	for dependent		for salary			3.643		45642.9	
20	min	1	min	30000		1.216		11639.5	
21	max	6	max	70000					
22	newn	1	newmax	1					
23	newn	0	newmin	0					

Comparison of Min-Max Normalization and Z-Score Normalization

Let’s see the comparison of Min-Max Normalization and Z-Score Normalization

Min-max normalization	**Z-score normalization**
Not very well efficient in handling the outliers	Handles the outliers in a good way.
Min-max Guarantees that all the features will have the exact	Helpful in the normalization of the data but not with the

same scale.	*exact* same scale.

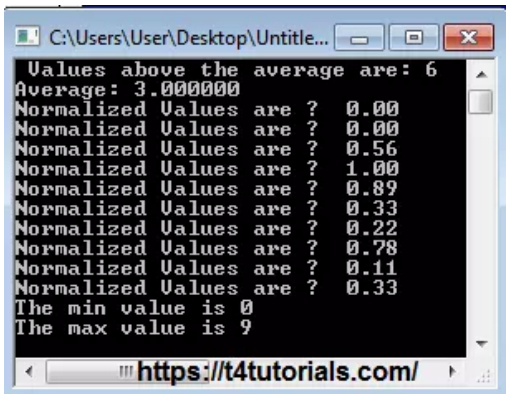
Implementation of Min-Max normalization in C++

- Calculate and show the maximum value from the array.
- Calculate and show the minimum value from the array.
- Calculate and show the average value from the array, and the number of values that are larger than the average.
- Calculate and show the normalized values of the original array values.

```
#include <stdio.h>
int Maximum_Value( double MinMaxArray[], int num_elements)
// This function will calculate and show the maximum value from the array
{
   int i, max=-32000;
   for (i=0; i<num_elements; i++)
   {
     if (MinMaxArray[i]>max)
     {
        max=MinMaxArray[i];
     }
   }
   return(max);
}
int MinValue( double MinMaxArray[], int num_elements)
// This function will calculate and show the minimum value from the array
{
    int i, min=0;
    for (i=0; i<num_elements; i++)
    {
       if (i==0)
           min=MinMaxArray[i];
    else
        if (MinMaxArray[i]<min)
            min=MinMaxArray[i];
    }
    return(min);
}
    double averageX(double MinMaxArray[], int num_elements)
    //This function will calculate and show the average value from the array, and the number
    // larger then the average
    {
        float average;
        int i, sum=0, valuesAboveAverage=0;
            for (i=0; i<num_elements; i++)
            {
                sum+=MinMaxArray[i];
                average=sum/num_elements;
                if (MinMaxArray[i]>average)
                    valuesAboveAverage++;
            }
            printf(" Values above the average are: %d\n", valuesAboveAverage);
            printf("Average: %f\n", average);
            return(average);
    }
    void norm_1D(int min, int max, int average, double MinMaxArray[], int num_elements)
    // This function will calculate and show the normalized values of the original
    // array values
    {
        int i;
        double normalizedVal[10];
        for (i=1; i<num_elements; i++)
        {
                normalizedVal[i]=(MinMaxArray[i]-min) / (max-min);
        }
        for (i=0; i<10; i++)
        printf("Normalized Values are ? %5.2f\n", normalizedVal[i]);
    }
int main(void)
{
   double  MinMaxArray[10] = {1, 0, 5, 9, 8, 3, 2, 7, 1, 3};
```

```
    int max, min;
    double average;
    max = Maximum_Value(MinMaxArray, 10);
    min = MinValue(MinMaxArray, 10);
    average = averageX(MinMaxArray, 10);
    norm_1D(min, max, average, MinMaxArray, 10);
    printf("The min value is %d\n", min);
    printf("The max value is %d\n", max);
     return(0);
}
```

Output



Example **of** Min-max scaling in data mining**:**

Min-max normalization detail is available in the previous tutorial.

Here, There is just another example of the practice.

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

S.No	Mid Term Marks	CGPA
1	84	4
2	63	3.2
3	77	2.6
4	78	2.1
5	90	3.2
6	75	3.7
7	49	2.1
8	79	2.9
9	77	1.9
10	52	2.5
11	74	3.3
12	90	3.9

mid term marks	
min	49
max	90
v	84
new_maxa	1
new_mina	0
√	0.853659

CGPA	
min	1.9
max	4
v	4
new_maxa	1
new_mina	0
√	1

min-max normalization

S.No	Mid Term Marks	CGPA
1	0.853658537	1
2	0.341463415	0.6190476
3	0.682926829	0.3333333
4	0.707317073	0.0952381
5	1	0.6190476
6	0.634146341	0.8571429
7	0	0.0952381
8	0.731707317	0.4761905
9	0.682926829	0
10	0.073170732	0.2857143
11	0.609756098	0.6666667
12	1	0.952381

The similarity after min-max normalization

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.6383	0.6882	0.9165	0.4081	0.2619	1.2439	0.5378	1.0145	1.058	0.413	0.1539
2	0.6383	0	0.4452	0.6389	0.6585	0.3773	0.6253	0.4156	0.707	0.4279	0.2725	0.7381
3	0.6882	0.4452	0	0.2393	0.4268	0.5261	0.7232	0.151	0.3333	0.6116	0.3413	0.6955
4	0.9165	0.6389	0.2393	0	0.6	0.7654	0.7073	0.3817	0.0983	0.6621	0.5797	0.9057
5	0.4081	0.6585	0.4268	0.6	0	0.4365	1.1289	0.304	0.6955	0.9849	0.3931	0.3333
6	0.2619	0.3773	0.5261	0.7654	0.4365	0	0.9913	0.3932	0.8585	0.8008	0.192	0.378
7	1.2439	0.6253	0.7232	0.7073	1.1289	0.9913	0	0.8249	0.6895	0.204	0.8357	1.3171
8	0.5378	0.4156	0.151	0.3817	0.304	0.3932	0.8249	0	0.4787	0.6855	0.2262	0.5466
9	1.0145	0.707	0.3333	0.0983	0.6955	0.8585	0.6895	0.4787	0	0.6734	0.6707	1.0038
10	1.058	0.4279	0.6116	0.6621	0.9849	0.8008	0.204	0.6855	0.6734	0	0.6581	1.1417
11	0.413	0.2725	0.3413	0.5797	0.3931	0.192	0.8357	0.2262	0.6707	0.6581	0	0.4837
12	0.1539	0.7381	0.6955	0.9057	0.3333	0.378	1.3171	0.5466	1.0038	1.1417	0.4837	0

Min Max Normalization in Python and Matlab

Min-Max normalization is explained very briefly in the **next tutorial**.

Related Posts:

1. **how to normalize the data with min max normalization...**
2. **Min Max Normalization Python and Matlab - Data Mining**
3. **Frequent pattern Mining, Closed frequent itemset,...**
4. **Standard Deviation normalization of data in data mining**
5. **Data Normalization before Data Mining**
6. **Z-Score Normalization (Data Mining)**
7. **Normalization with decimal scaling in data mining - Examples**
8. **Data Stream Mining - Data Mining**
9. **What is data mining? What is not data mining?**
10. **Horizontal Fragmentation, min term predicates,...**