

PRACTICAL- 2(A)
NORMALIZATION

AIM:

Write a c program to implement min-max Normalization.

Data transformation:

In data transformation process data are transformed from one format to another format, that is more appropriate for data mining.

Some data transformation strategies:

1. Smoothing:

It is process of removing noise from the data.

2. Aggregation:

It is a process where summary or aggregation operations are applied to data.

3. Generalization:

In generalization low level data are replaced with high level data by using concept hierarchies climbing.

4. Normalization:

Normalization scales attribute data so as to fall within a small specified range, such as 0.0 to 1.0.

Data Normalization:

Normalization is a data transformation technique, means an attribute data is scaled to fall within a specific range like [0,1), [-1, +1], etc.,

There are three types of normalization.

MIN MAX NORMALIZATION:

In Min-Max normalization, for every feature the minimum value of that feature gets Transformed into '0' or '-1' and maximum value transformed into '1' and every other value get transformed into a decimal between min and max values

Steps:

Let age attribute values are

10 20 5 30 50 55

We should consider new min & new max values as new

min=0 & new max=1

The formula we used is

$$V_{\text{minA}} / \text{MaxA} - \text{MinA} = v1 - \text{new minA} / \text{NewMaxA} - \text{New MinA}$$

here V be the actual value & V1 be the
Normalized value. For v=10

$$5/50 = v - 0 / 1 - 0$$

$$v1 = 0.1$$

For v = 20

$$15/50 = v - 0 / 1 - 0$$

$$v1 = 0.3$$

For v=5

$$0/50 = v - 0 / 1 - 0$$

```

v1=0
For v=30
25/S0=v'-0/1-0
v1=0.5
For v=50
45/S0=v'-0/1-0
V 1 =0.9
For v=55
50/ 50= v1-o/1-o
v1=1

```

Normalizing the values of the age attribute are given in the following table.

| Age | Min-Max Normalization |
|-----|-----------------------|
| 10 | 0.1 |
| 20 | 0.3 |
| 5 | 0 |
| 30 | 0.5 |
| 50 | 0.9 |
| 55 | 1 |

Program:

```

#include<stdio.h>
void main()
{
float min,max,newmin,newmax,y,v;
printf("Performing the min and max normalization\n");
printf("Enter min:");
scanf("%f",&min);
printf("Enter max:");
scanf("%f",&max);
printf("Enter New Min:");
scanf("%f",&newmin);
printf("Enter New Max:");
scanf("%f",&newmax);
printf("Enter value of v:");
scanf("%f",&v);
y=((v-min)/(max-min))*(newmax-newmin)+newmin
printf("Value of y:%f",y);
}

```

OUTPUT:

OUT PUT:

```
C:\TURBOC3\BIN>TC
Performing the min and max normalization
Enter min:2
Enter max:4
Enter New Min:8
Enter New Max:10
Enter value of v:12
Value of y:18.000000Performing the min and max normalization
Enter min:
```

PRACTICAL- 2(B)

Z-SCORE NORMALIZATION

AIM :A c-program to implement the z-score Normalization technique.

Z-Score Normalization

In z-score normalization transform the data by converting the values to a common scale with an average of zero normalized to V' by computing

$$V' = \frac{V - \bar{A}}{\sigma A}$$

Where \bar{A} and σA are the mean and standard deviation of attribute A

$$\sigma A = \sqrt{\frac{\sum (Every\ individual\ value\ of\ A - mean\ of\ values)^2}{(No.\ of\ values\ in\ A - 1)}}$$

Explanation:

1. Consider age attribute values(A)

10 20 15 30 50 55

2. Find out the mean

\bar{A} = mean of the attribute A

(10+20+15+30+50+55)/6=30

3. σA =standard deviation of attribute A

$$\sqrt{\frac{\sum (A - \bar{A})^2}{N - 1}}$$

N= No of values attribute

$$\sum (A - \bar{A})^2 = 1950$$

$$\begin{aligned} &= \sqrt{\frac{1950}{5}} \\ &= 18.7 \end{aligned}$$

The Normalized value is v'

The actual value is v

for $v=10 \Rightarrow v' = (10 - 30) / 18.7$

for $v=10 \Rightarrow v' = 10 - 30 / 18.7$

$= -1.06$

for $v=20 \Rightarrow v' = 20 - 30 / 18.7$

$= -0.534$

for $v=15 \Rightarrow v' = 15 - 30 / 18.7$

$= -0.802$

for $v=50 \Rightarrow v' = 50 - 30 / 18.7$

$= 1.06$

for $v=30 \Rightarrow v' = 30 - 30 / 18.7$

$= 0$

for $v=55 \Rightarrow v' = 55 - 30 / 18.7$

$= 1.336$

After Normalization using z-score normalization the attribute
Avalues are:

| Age | z-score Normalization |
|-----|-----------------------|
| 10 | -1.06 |
| 20 | -0.534 |
| 15 | -0.802 |
| 30 | 0 |
| 50 | 1.06 |
| 55 | 1.336 |

PROGRAM:

```
#include<stdio.h>
```

```
#include<math.h>
```

```
int main()
```

```
{
```

```
int i,n;
```

```
float v,v1,sig,avg,sum=0,a[20];
```

```
printf("Enter number of elements to be entered\n");
```

```
scanf("%d",&n);
```

```
printf("Enter the elements\n");
```

```
for(i=0;i<n;i++) scanf("%f",&a[i]);
```

```

printf("enter v value\n");
scanf("%f",&v);
for(i=0;i<n;i++)
sum=sum+a[i];
avg=sum/n;
printf("sum is =%f\n",sum);
for(i=0;i<n;i++)
{
sum=sum+pow(avg-a[i],2);
}
sig=sum/n;
printf("variance is %f\n",sqrt(sig));
v1=(v-avg)/sqrt(sig);
printf("v1 value=%f\n",v1);
}

```

OUTPUT:

OUTPUT:

```

C:\TURBOC3\BIN>TC
Enter number of elements to be entered
4
Enter the elements
23
12
38
27
enter v value
20
sum is =100.000000
variance is 10.559356
v1 value=-0.473514
Enter number of elements to be entered
-

```