Name: Pratham Yadav
PRN: 2020BTEIT0000(
Sub: DM LAB

Experiment NO. 2

Aim: To perform normalization of data (min-max) and (z-score)

Theory: Data normalization is a basic element of data mining. It means transforming the data, namely converting the source data into another format that allows processing data effectively. The main purpose of data normalization is to minimize or even exclude duplicate data. This is a very essential and important issue because it is increasingly problematic to keep data in relational databases, which stores identical data in more than one place.

The use of data mining normalization has a number of advantages:

- Data mining algorithms get more effective and efficient
- Data is converted into the format that everyone can get their heads around
- Data can be extracted from database faster.
- It is possible to analyze the data in specific manner.

Formula:

1) Min-Max Normalization

$$D' = \left(\frac{D - \min F}{\max F - \min F}\right) \times (new\_maxF - new\_minF) + new\_minF$$

Where v' - new value

v - Respected value of the attribute

maxF - maximum value from given dataset

minF - minimum value from given dataset

new_minF : minimum value by user

new_maxF : maximum value by user

B) Z-Score Normalization.

$$v' = \frac{v - \bar{F}}{\sigma_F}$$

where v - actual data value

F - mean value of data

σ_F - standard deviation of data

Algorithm:

1) Take dataset and read data from csv file.

2) If we choose min-max normalization then find min-max value from given dataset and if it is z-score normalization then calculate mean and standard deviation.

3) For min-max normalization conside new_min, new_max by considering suitable range

4). Calculate normalization value for all data in the given dataset with using respective normalized formula.

Example : A) Min-Max normalization

Data [Marks] min = 20 max = 75

20 new_min = 0 new_max = 1

75

84

55

63

For marks 20:

$$V'_{20} = \frac{20-20}{75-20} \times (1-0) + 0 = 0$$

$$V'_{75} = \frac{75-20}{75-20} \times (1-0) + 0 = 1 \; \text{~~0·2545~~}$$

$$V'_{84} = \frac{\cdot 34-20}{75-20} \times (1-0) + 0 = 0\cdot2545$$

$$V'_{55} = \frac{55-20}{75-20} \times (1-0) + 0 = 0\cdot6363$$

$$V'_{63} = \frac{63-20}{75-20} \times (1-0) + 0 = 0\cdot7818$$

Normalized Table

Marks	Normalized marks
20	0
75	1
34	0.25
55	0.64
63	0.78

B) Z-score Normalization

Marks : [9, 12, 15, 20]

$$\text{Mean} = \frac{9+12+15+20}{4} = 14 \qquad \mu = 14$$

Standard Deviation.

$$\sigma = \sqrt{\frac{(9-14)^2 + (12-14)^2 + (15-14)^2 + (20-14)^2}{4}}$$

$$= \sqrt{\frac{25+4+1+36}{4}} = \sqrt{16.5} = 4.062$$

For marks 9

$$Z_9 = \frac{9-14}{4.062} = -1.231$$

For mark 12:

$$Z_{12} = \frac{12-14}{4.062} = -0.4923$$

$$Z_{15} = \frac{15-14}{4.062} = 0.2462$$

$$Z_{20} = \frac{20-14}{4.062} = 1.4771$$

Marks	Z-score Normalized Marks
9	-1.23
12	-0.49
25	0.25
20	1.48

Conclusions:

The transformed values obtained by calculation and by knime and weka are identical

successfully performed transformation of given data using min-max and z-score normalization

Knime Result :

Min-Max

Input Marks	Knime min-max Normalized
20	0
25	1
34	0,255
55	0.636
63	0.783

z-score

I/P Mark	z-score Normalized
9	-1.066
12	-0.426
15	0.213
20	1.279

Conclusion:

Normalization is an important processor task in data preprocessing. It is used to ensure consistency ~~d~~ in data records. In order to bring all attribute on the same scale min-max normalization is used. Standardizing score on some scale by dividing a scores deviation by standard deviation for that z-score is used It scales a date to particular small scale which helps to allow processing data effeciency.

Program:

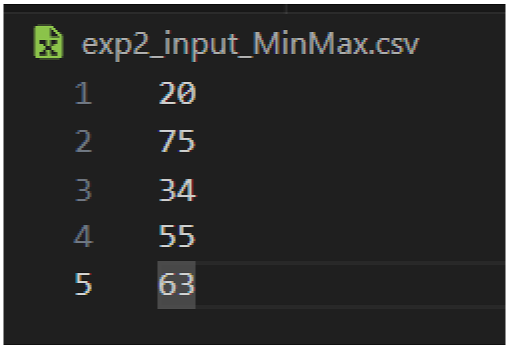
```
#include <bits/stdc++.h>
#include <fstream>
using namespace std;

void calculateMinMax(ifstream &inputFile, double &minValue, double &maxValue)
{
    double currentValue;
    inputFile >> currentValue;
    while (inputFile)
    {
        if (currentValue > maxValue)
            maxValue = currentValue;
        if (currentValue < minValue)
            minValue = currentValue;
        inputFile >> currentValue;
    }
}

void performMinMaxNormalization(ifstream &inputFile, ofstream &outputFile, double oldMin,
double oldMax, double newMin, double newMax)
{
    outputFile << "Original Data,"
               << "Normalized Data"
               << "\n";

    double currentValue;
    inputFile >> currentValue;
    while (inputFile)
    {
        double previousValue = currentValue;
        currentValue = (((currentValue - oldMin) / (oldMax - oldMin)) * (newMax - newMin)) +
newMin;
        outputFile << previousValue << "," << currentValue << "\n";
        inputFile >> currentValue;
    }
}

void performZScoreNormalization(ifstream &inputFile, ofstream &outputFile)
```

```
{
    double sum = 0.0, count = 0.0, squareSum = 0.0, mean, standardDeviation;
    double currentValue;

    // Calculate mean
    while (inputFile)
    {
        sum += currentValue;
        count++;
        inputFile >> currentValue;
    }
    mean = sum / count;

    // Calculate standard deviation
    inputFile.clear();
    inputFile.seekg(0, ios::beg);
    while (inputFile)
    {
        squareSum += (currentValue - mean) * (currentValue - mean);
        inputFile >> currentValue;
    }
    inputFile.clear();
    inputFile.seekg(0, ios::beg);

    standardDeviation = sqrt(squareSum / count);

    // Perform z-score normalization
    outputFile << "Original Data,"
               << "Normalized Data"
               << "\n";

    while (inputFile)
    {
        double prev = currentValue;
        currentValue = (currentValue - mean) / standardDeviation;
        outputFile << prev << "," << currentValue << endl;
        inputFile >> currentValue;
    }
}
```

```
int main()
{
    double currentValue, minValue, maxValue, newMinValue, newMaxValue;
    double sum, count, squareSum, mean, standardDeviation;

    ifstream inputFileMinMax("exp2_input_MinMax.csv");
    ifstream inputFileMinMax_2("exp2_input_MinMax.csv");
    ifstream inputFileZScore("exp2_input_Zscore.csv");

    int option;
    cout << "\nEnter an option: \n1. Min-Max Normalization \n2. Z-Score Normalization\nOption:
";
    cin >> option;

    ofstream outputFileMinMax("exp2_output_MinMax.csv", ios::app);
    ofstream outputFileZScore("exp2_output_ZScore.csv", ios::app);

    switch (option)
    {
    case 1: // Min-Max Normalization
        if (!inputFileMinMax)
        {
            cout << "Error opening file, please try again.";
            exit(0);
        }
        calculateMinMax(inputFileMinMax, minValue, maxValue);
        cout << "Enter new minimum value: ";
        cin >> newMinValue;
        cout << "\nEnter new maximum value: ";
        cin >> newMaxValue;
        performMinMaxNormalization(inputFileMinMax_2, outputFileMinMax, minValue,
maxValue, newMinValue, newMaxValue);
        break;

    case 2: // Z-Score Normalization
        if (!inputFileZScore)
        {
            cout << "Error opening file, please try again.";
            exit(0);
        }
        performZScoreNormalization(inputFileZScore, outputFileZScore);
        break;

    default:
        cout << "Invalid option";
    }

    return 0;
}
```

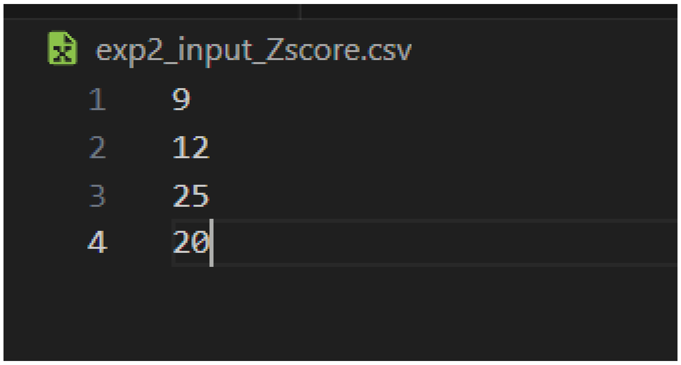
I/P

Min-Max



```
20
75
34
55
63
```

Z-Score

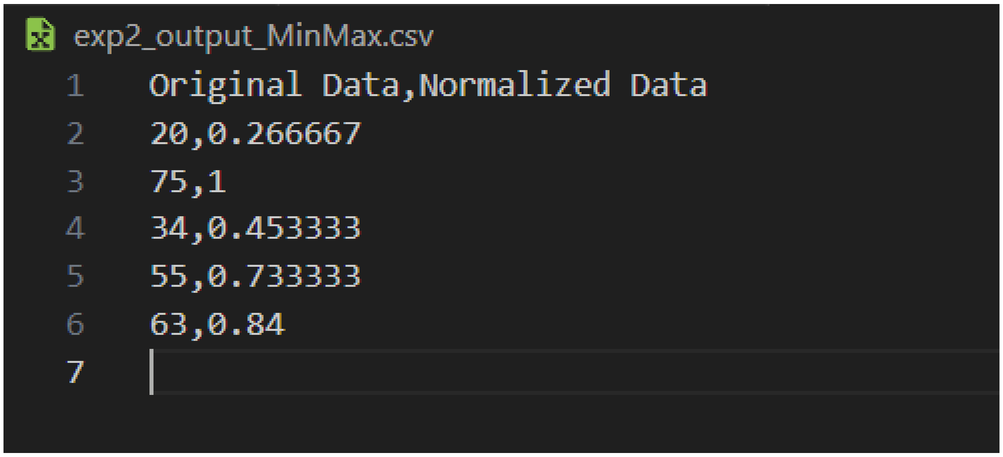


```
9
12
25
20
```

,

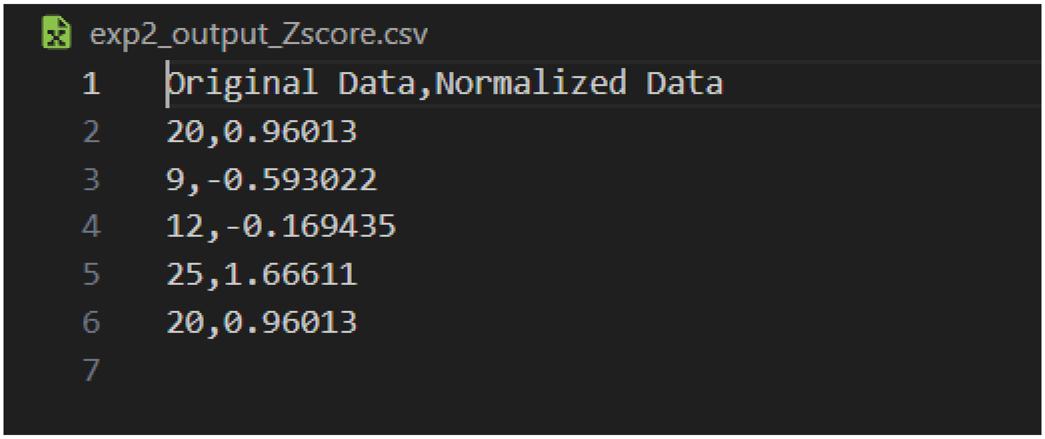
O/P

MIN_MAX



```
Original Data,Normalized Data
20,0.266667
75,1
34,0.453333
55,0.733333
63,0.84

```

Z-Score



```
Original Data,Normalized Data
20,0.96013
9,-0.593022
12,-0.169435
25,1.66611
20,0.96013

```