**Experiment no. 2**

**Name: Sonali Dattatray Kaingade**

**PRN: 21620002**

**Title:** To perform normalization of data (Min-max and z-score).

**1. Min max normalization**

**Code:**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

double min\_max\_scaling(double x, double x\_min, double x\_max, double x\_newmin, double x\_newmax) {

    return ((x - x\_min) / (x\_max - x\_min)) \* (x\_newmax - x\_newmin) + x\_newmin;

}

int main() {

    ifstream input\_file("input.txt");

    ofstream output\_file("output\_minmax.txt");

    vector<double> data;

    double value;

    while (input\_file >> value) {

        data.push\_back(value);

    }

    double x\_min = \*min\_element(data.begin(), data.end());

    double x\_max = \*max\_element(data.begin(), data.end());

    double x\_newmin = 1.0; // New minimum value for scaled data

    double x\_newmax = 10.0; // New maximum value for scaled data

    for (const double &x : data) {

        double normalized\_value = min\_max\_scaling(x, x\_min, x\_max, x\_newmin, x\_newmax);

        output\_file << normalized\_value << endl;

    }

    input\_file.close();

    output\_file.close();

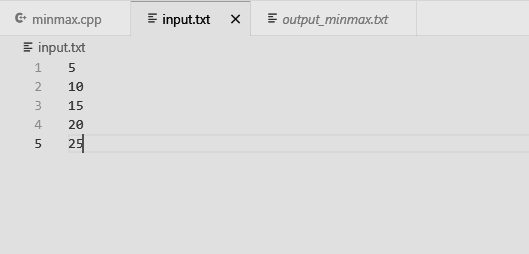
    cout << "output is generated in output\_minmax file" << endl;

    return 0;

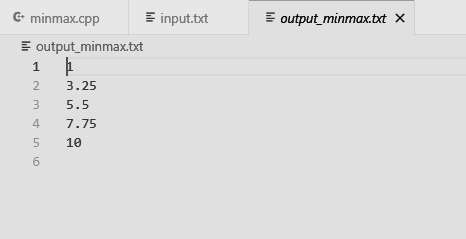
}

**Output:**

**input.txt**



**Output\_minmax.txt**



**2. Z-score normalization**

**Code:**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <cmath>

using namespace std;

double z\_score(double x, double mean, double std\_dev) {

    return (x - mean) / std\_dev;

}

int main() {

    ifstream input\_file("input.txt");

    ofstream output\_file("output\_zscore.txt");

    vector<double> data;

    double value;

    while (input\_file >> value) {

        data.push\_back(value);

    }

    double sum = 0.0;

    for (const double &x : data) {

        sum += x;

    }

    double mean = sum / data.size();

    double squared\_diff\_sum = 0.0;

    for (const double &x : data) {

        squared\_diff\_sum += pow(x - mean, 2);

    }

    double std\_dev = sqrt(squared\_diff\_sum / data.size());

    for (const double &x : data) {

        double normalized\_value = z\_score(x, mean, std\_dev);

        output\_file << normalized\_value << endl;

    }

    input\_file.close();

    output\_file.close();

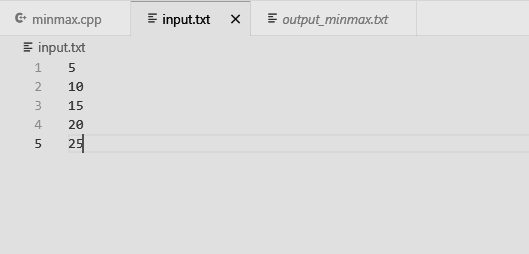
    cout << "output is generated in output\_zscore file" << endl;

    return 0;

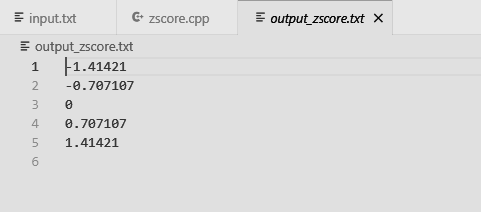
}

**Output:**

**input.txt:**

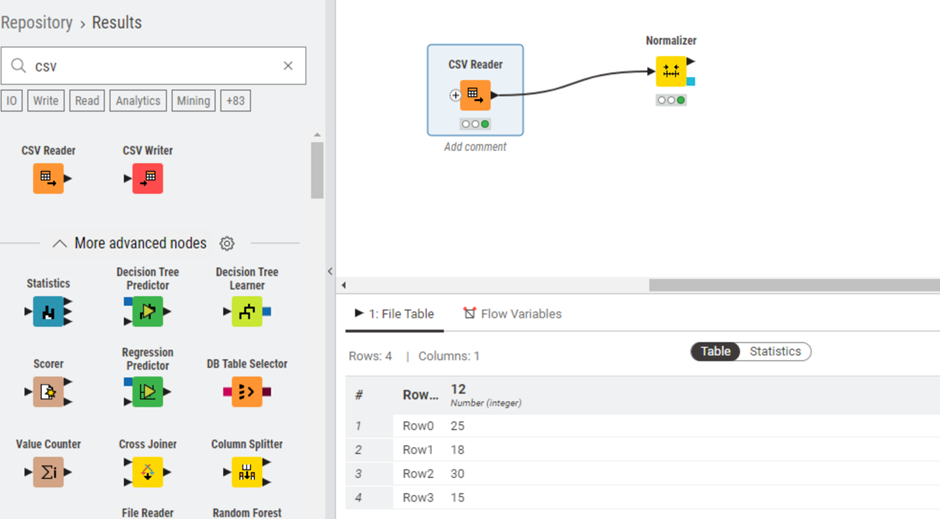
****

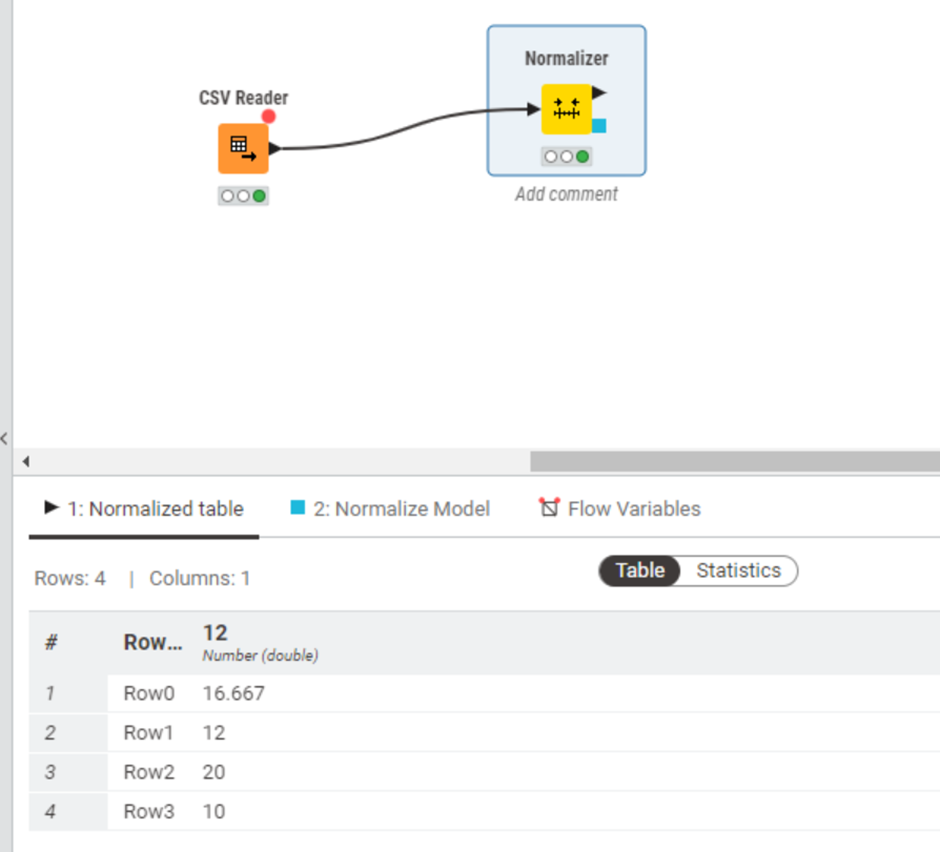
**Output\_zscore.txt:**

****

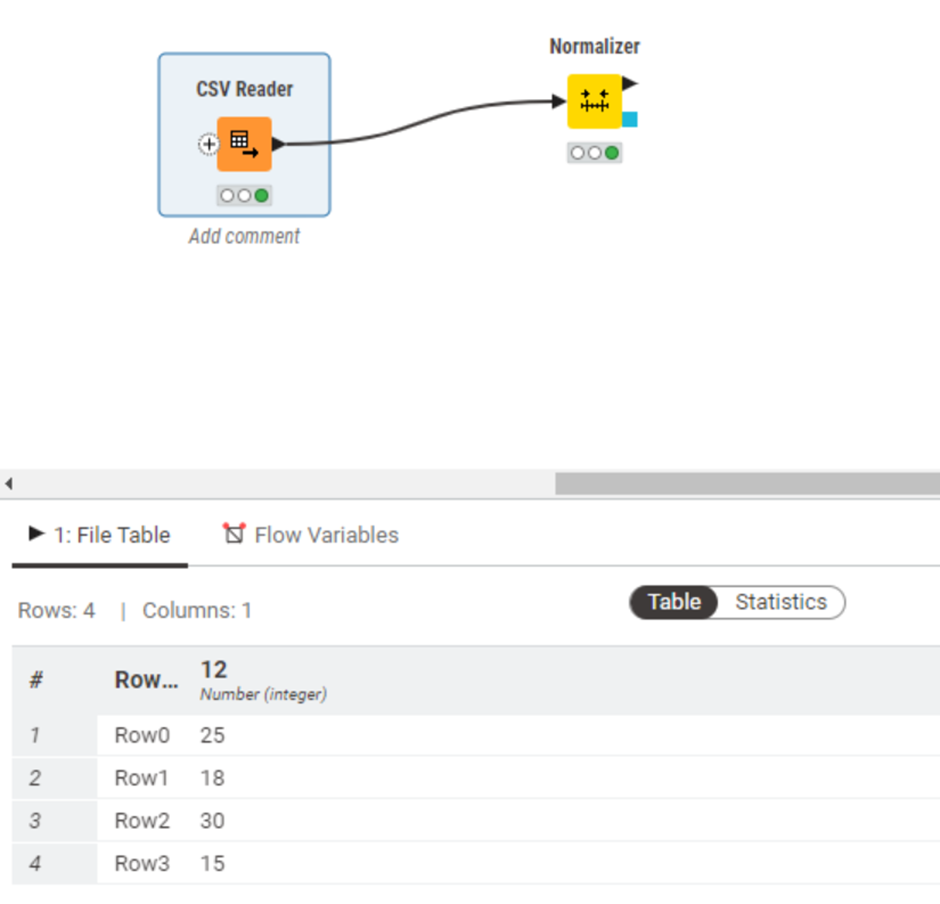
**Knime:**

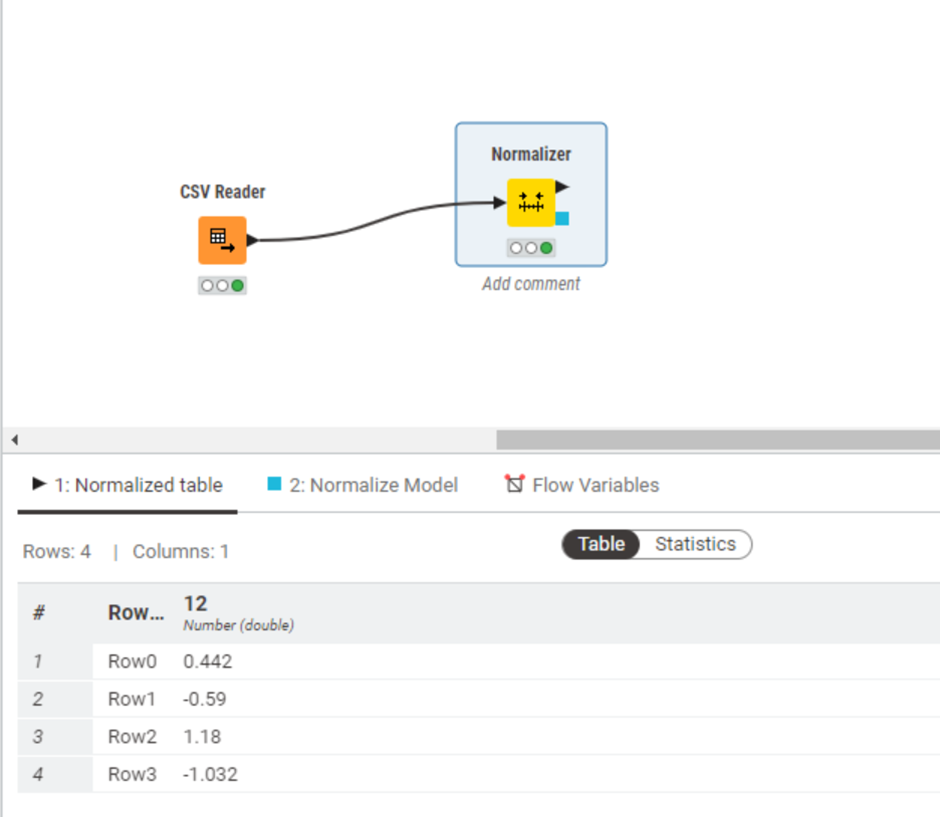
**Min-Max Normalization:**

****

****

**Z - Score Normalization:**

****

****