**Experiment no. 10**

**Name: Sonali Dattatray Kaingade**

**PRN: 21620002**

**Title:**  Distance and cluster

- Assume some points (multidimensional)

Compute centre of cluster assuming all points belonging to one cluster.

Find distance of all points with obtained cluster centre using suitable distance function

Display results in the form of upper or lower triangular matrix

**code:**

#include <bits/stdc++.h>

using namespace std;

double finddist(pair<double,double>p1, pair<double,double>p2){

double x1 = p1.first;

double x2 = p2.first;

double y1 = p1.second;

double y2 = p2.second;

return sqrt((x1-x2)\*(x1-x2) + (y1-y2)\*(y1-y2));

}

int main(){

string line;

ifstream input("cluster\_input.csv");

vector<pair<double, double>>points;

int j = 0;

while(getline(input,line)){

stringstream str(line);

if(j==0){

j++;

continue;

}

string name,x,y;

getline(str,name,',');

getline(str,x,',');

getline(str,y);

points.push\_back({stoi(x), stoi(y)});

}

input.close();

double x\_sum, y\_sum, x\_mean, y\_mean;

int n = points.size();

for(auto it : points){

x\_sum += it.first;

y\_sum += it.second;

}

x\_mean = x\_sum/n;

y\_mean = y\_sum/n;

//calculate distance of each point from every point

ofstream output("output.csv");

output<<","<<"p1," <<"p2,"<<"p3,"<<"p4,"<<"p5"<<endl;

for(int i = 0; i<points.size();i++){

output<<"p"<<i+1<<",";

for(int j = 0; j<=i;j++){

if(i==j){

output<<"0,";

}

else{

double ans = finddist(points[i],points[j]);

output<<ans<<",";

}

}

output<<endl;

}

output<<"centroid,";

for(int i = 0; i<points.size(); i++){

double ans = finddist(points[i],{x\_mean,y\_mean});

output<<ans<<",";

}

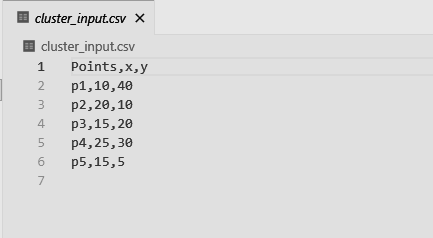
cout << endl;

output.close();

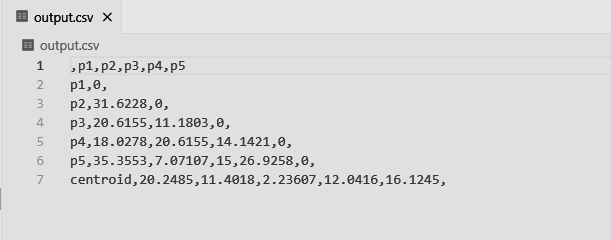
}

**Output:**

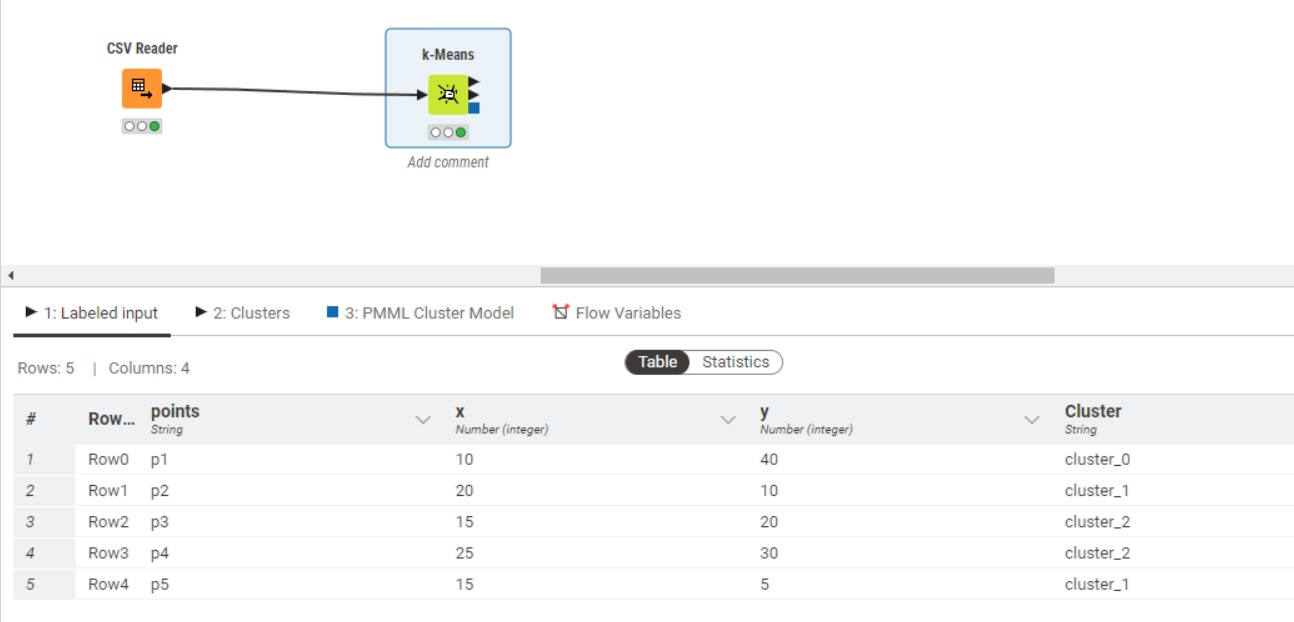
**Input.csv:**



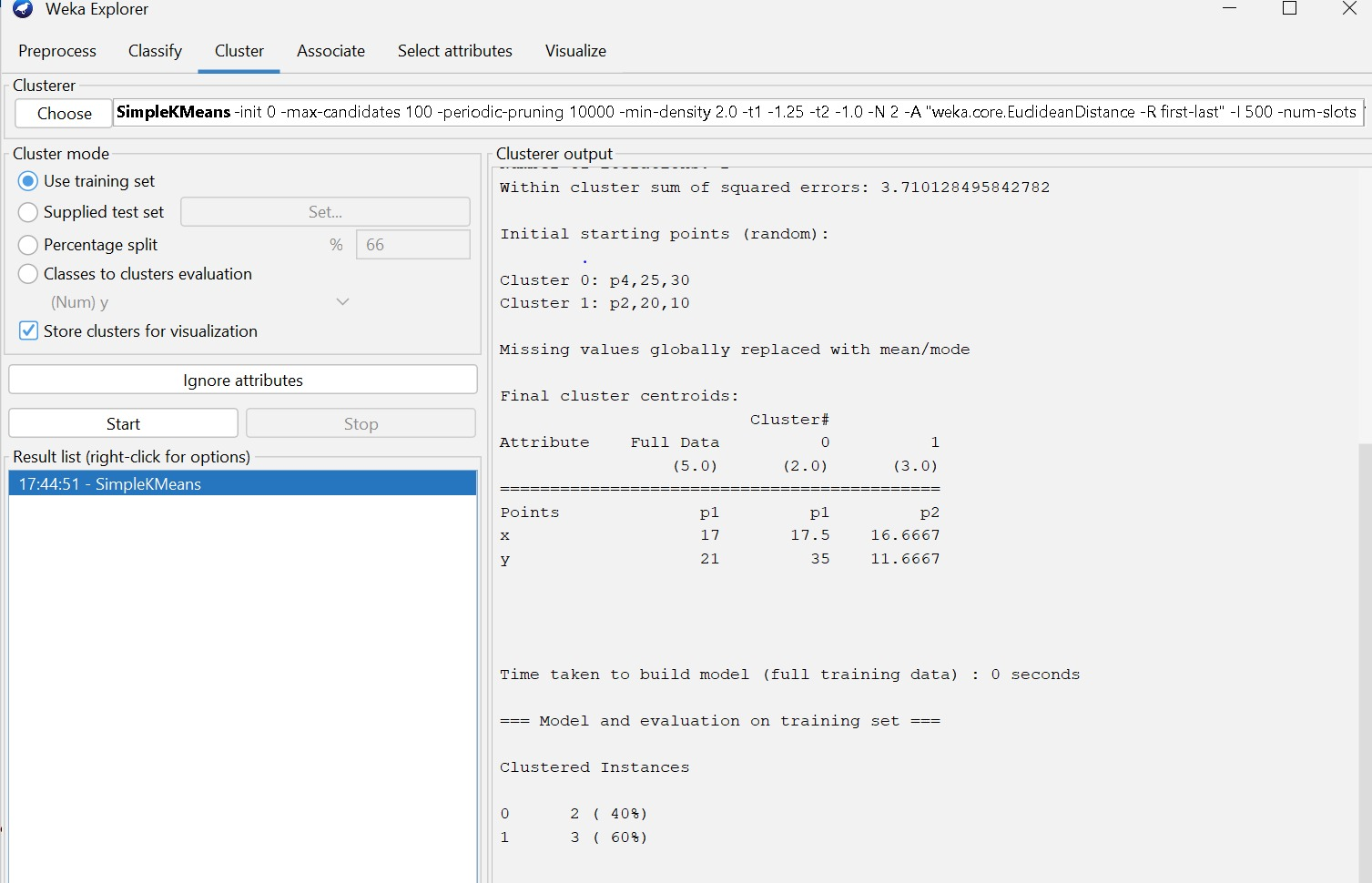
**Output.csv:**



**Knime:**

****

**Weka:**

****