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:

cluster_input.csv

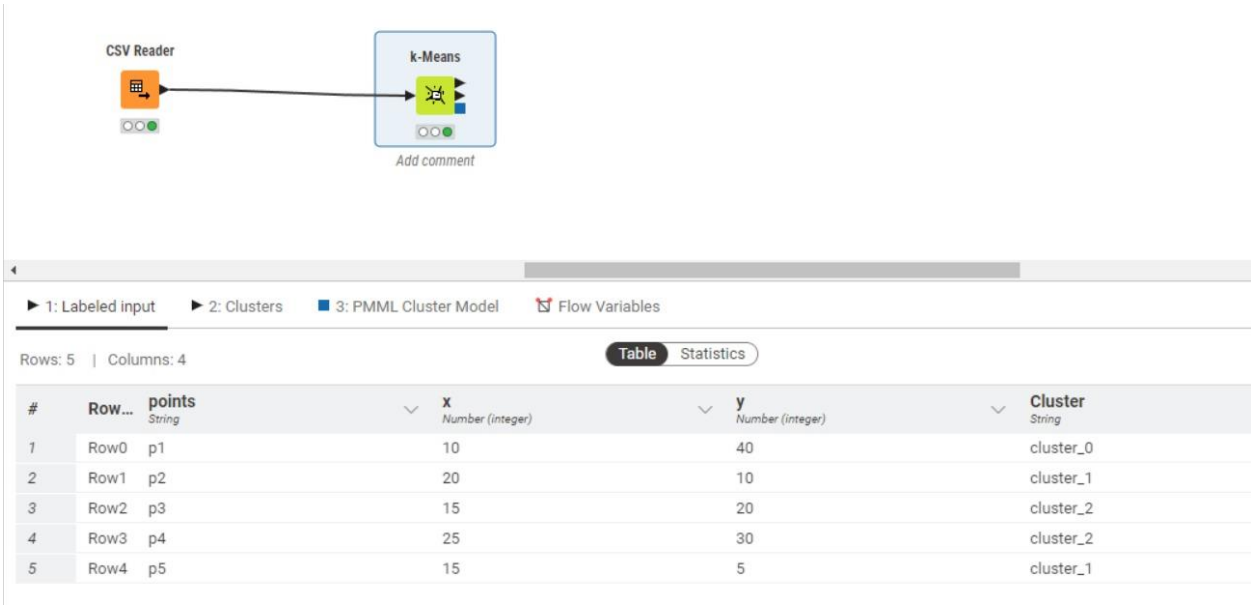
```
Points,x,y
p1,10,40
p2,20,10
p3,15,20
p4,25,30
p5,15,5
```

Output.csv:

output.csv

```
,p1,p2,p3,p4,p5
p1,0,
p2,31.6228,0,
p3,20.6155,11.1803,0,
p4,18.0278,20.6155,14.1421,0,
p5,35.3553,7.07107,15,26.9258,0,
centroid,20.2485,11.4018,2.23607,12.0416,16.1245,
```

Knime:



#	Row...	points String	x Number (integer)	y Number (integer)	Cluster String
1	Row0	p1	10	40	cluster_0
2	Row1	p2	20	10	cluster_1
3	Row2	p3	15	20	cluster_2
4	Row3	p4	25	30	cluster_2
5	Row4	p5	15	5	cluster_1

Weka: