Sample: C# - Data Mining

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Mining
{
    public partial class Calc : Form
    {
        public Calc()
        {
            InitializeComponent();
        }


        Bitmap btm = new Bitmap(420, 350);
        Graphics gr=null;
        float[,] points = new float[Program.second.mark_X.Length, 2];
        List<TextBox> centr = new List<TextBox>();



        private void Calc_Load(object sender, EventArgs e)
        {
                gr = Graphics.FromImage(btm);
            Pen pn1 = new Pen(Color.Black, 1);
            pn1.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;


            gr.DrawLine(pn1, 10, 340, 10, 10);
            gr.DrawLine(pn1, 10, 340, 340, 340);
            pictureBox1.Image = (Image)btm;


            for (int i = 0; i < Program.second.mark_X.Length; i++)
            {
                gr.FillRectangle(Brushes.Blue, 10 + (float)(Program.second.mark_X[i] *
330) / 4, 340 - (float)(Program.second.mark_Y[i] * 340) / 3, 5, 5);
                points[i,0] =10 + (float)(Program.second.mark_X[i] * 330) / 4;
                points[i,1] = 340 - (float)(Program.second.mark_Y[i] * 340) / 3;
            }


        }

        List<Point> Cen_p = new List<Point>();


        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

            for (int i = 0; i < centr.Count; i++) Controls.Remove(centr[i]);
```

```
            centr.Clear();
            gr.Clear(SystemColors.Control);
            for (int i = 0; i < comboBox1.SelectedIndex+2; i++)
            {
                centr.Add(new Mytxt { Location = new Point(485, 112 + i + (i+1) * 30),
Text = "0", Width=40 });
                centr[centr.Count - 1].Parent = this;
                centr.Add(new Mytxt { Location = new Point(568, 112 + i + (i + 1) * 30),
Text = "0", Width = 40 });
                centr[centr.Count - 1].Parent = this;
            }

            button1.Visible = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            gr.Clear(SystemColors.Control);



            Pen pn1 = new Pen(Color.Black, 1);
            pn1.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;
            gr.DrawLine(pn1, 10, 340, 10, 10);
            gr.DrawLine(pn1, 10, 340, 340, 340);
            Cen_p.Clear();

            for (int i = 0; i < centr.Count; i=i+2)
            {
                Cen_p.Add(new Point(Int32.Parse(centr[i].Text),340-
Int32.Parse(centr[i+1].Text)));
                //gr.FillEllipse(Brushes.Black, Cen_p[Cen_p.Count - 1].X+6,
Cen_p[Cen_p.Count - 1].Y-6, 12, 12);
            }
             double[] clus_dis = new double[Cen_p.Count];


            double[] dis = new Double[Cen_p.Count];
            double min_dis = 0;
            int idex=0;


            for (int i = 0; i < points.GetLength(0); i++)
            {
                min_dis = Double.MaxValue;
                idex=0;
                for (int j = 0; j < dis.Length; j++)
                {
                    dis[j] = Math.Sqrt((Cen_p[j].X - points[i, 0]) * (Cen_p[j].X -
points[i, 0]) + (Cen_p[j].Y - points[i, 1]) * (Cen_p[j].Y - points[i, 1]));
                    if (min_dis > dis[j]) { idex = j; min_dis = dis[j]; }
                }

                clus_dis[idex] += dis[idex];


                switch (idex)
                {
```

```
                    case 0: gr.FillRectangle(Brushes.Red, points[i, 0], points[i, 1], 5,
5); break;
                    case 1: gr.FillRectangle(Brushes.Green, points[i, 0], points[i, 1],
5, 5); break;
                    case 2: gr.FillRectangle(Brushes.Blue, points[i, 0], points[i, 1], 5,
5); break;
                    case 3: gr.FillRectangle(Brushes.Yellow, points[i, 0], points[i, 1],
5, 5); break;

                }
            }

            for (int i = 0; i < Cen_p.Count; i ++)
            {
                gr.FillEllipse(Brushes.Black, Cen_p[i].X + 6, Cen_p[i].Y - 6, 12, 12);
            }

            pictureBox1.Image = (Image)btm;
            Color[] color = { Color.Red, Color.Green, Color.Blue, Color.Yellow };
            string info = "Clusters info(SSE):\n";
            for (int i = 0; i < clus_dis.Length; i++)
            {
                info += (i + 1).ToString() + ": " + color[i].Name;
                info += String.Format("{0,15:###,###.##}\n", clus_dis[i]);
            }
            MessageBox.Show(info);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Random rnd = new Random();
            int count = rnd.Next(2, 5);

            centr.Clear();

            for (int i = 0; i < count; i++)
            {
                centr.Add(new Mytxt{Text=rnd.Next(30,320).ToString()});
                centr.Add(new Mytxt { Text = rnd.Next(30, 320).ToString() });
            }

            button1_Click(null, null);
        }
    }



    class Mytxt:TextBox
    {
        public Mytxt()
        {
            KeyPress += Mytxt_KeyPress;
        }

        void Mytxt_KeyPress(object sender, KeyPressEventArgs e)
        {
           int digit=0;
           if(e.KeyChar=='\b') return;
```

```
            if (!Int32.TryParse(Text + e.KeyChar, out digit) || digit > 340) e.Handled =
true;
        }
    }

}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Windows.Forms;


namespace Mining
{
    using Excel = Microsoft.Office.Interop.Excel;

    public partial class Form1 : Form
    {
        DataGridView table = new DataGridView();
        public double[] mark_X;
        public double[] mark_Y;
        public Form1()
        {
            InitializeComponent();
            table.RowsAdded += table_RowsAdded;
            table.RowHeadersWidthSizeMode =
DataGridViewRowHeadersWidthSizeMode.AutoSizeToAllHeaders;
            table.AllowUserToAddRows = false;
        }

        void table_RowsAdded(object sender, DataGridViewRowsAddedEventArgs e)
        {
            table.Rows[e.RowIndex].HeaderCell.Value = (e.RowIndex + 1).ToString();
        }

        public static List<List<string>> attributes = new List<List<string>>();

        private void Form1_Load(object sender, EventArgs e)
        {

        }


        public double SimMeasure(List<int> mas, int value)
        {
            double measure = 1.0;
            int mid = (mas.Max() + mas.Min()) / 2;
            double delta = (1 / (double)(mas.Max() - (double)(mas.Max() + mas.Min()) /
2));

            for (int i = 0; i <= mid; i++)
            {
                if (mid+i+1 == value || mid - i == value) return measure;
                else
                {
                    measure -= delta;
                }
            }
```

```
            return measure;
        }

        public double SimMeasure(List<DateTime> mas, DateTime value)
        {
            double measure = 1.0;
            long maxDate =mas.Max().Ticks/1000000000000;
            long minDate = mas.Min().Ticks / 1000000000000;



            long mid =  (maxDate+minDate)/ 2;
            double delta = 1 / (double)(maxDate - mid);

            for (int i = 0; i <= mid; i++)
            {
                if (mid + i + 1 == value.Ticks / 1000000000000 || mid - i == value.Ticks
/ 1000000000000) return measure;
                else
                {
                    measure -= delta;
                }
            }

            return measure;
        }

        public double SimMeasure(List<string> mas, string value)
        {
            double measure = 1.0;
            char chMax = mas.Max()[0];
            char chMin = mas.Min()[0];

            char mid = (char)((int)(chMax + chMin) / 2);
            double delta = 1 / (double)(chMax - chMin-10);

            for (int i = 0; i <= mid; i++)
            {
                if (mid + i + 1 == value[0] || mid - i == value[0]) return measure;
                else
                {
                    measure -= delta;
                }
            }

            return measure;
        }

        public void button1_Click(object sender, EventArgs e)
        {

            for (int i = 0; i < Program.first.ls.Length; i++)
            {
                attributes.Add(Program.first.ls[i].attribute);
            }
```

```
            DateTime date;
            for (int i = 0; i < attributes.Count; i++)
            {
                table.Columns.Add(new DataGridViewTextBoxColumn { HeaderText =
attributes[i][0] });
                table.Columns[table.Columns.Count - 1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.AllCells;

            }

            for (int i = 1; i < attributes[0].Count; i++)
            {
                table.Rows.Add();
                for (int j = 0; j < attributes.Count; j++)
                {
                    if (DateTime.TryParse(attributes[j][i], out date)) attributes[j][i] =
date.ToShortDateString();
                    table.Rows[i - 1].Cells[j].Value = attributes[j][i];
                }
            }


            table.Location = new Point(5, 25);
            table.Size = new Size(20, ClientSize.Height - 10);
            table.MaximumSize = new Size(ClientSize.Width - 10, ClientSize.Height - 10);


            for (int i = 0; i < table.Columns.Count; i++)
            {
                table.Size = new Size(table.Size.Width +
table.Columns[i].HeaderCell.Size.Width, table.Size.Height);
            }

            table.Parent = this;
            int res = 0;
            DateTime res_d;

            for (int i = 0; i < attributes.Count; i++) attributes[i].RemoveAt(0);

            mark_X = new double[attributes[0].Count];
            mark_Y = new double[attributes[0].Count];
            ////////////////////////////////////////
            for (int i = 0; i < attributes.Count; i++)
            {
                if (Int32.TryParse(attributes[i][1], out res))
                    for (int j = 0; j < attributes[i].Count; j++)
                    {
                        mark_X[j] += SimMeasure(attributes[i].ConvertAll<int>((n) => {
int m = 0; Int32.TryParse(n, out m); return m; }), Convert.ToInt32(attributes[i][j]));
                    }

                else
                    if (DateTime.TryParse(attributes[i][1], out res_d))
                        for (int j = 0; j < attributes[i].Count; j++)
                        {
                            mark_X[j] +=
SimMeasure(attributes[i].ConvertAll<DateTime>((n) => { DateTime m; DateTime.TryParse(n,
out m); return m; }), DateTime.Parse(attributes[i][j]));
```

```
                    }
                else
                {
                    if (attributes[i][0].Contains('-')) continue;
                    for (int j = 0; j < attributes[i].Count; j++)
                    {
                        mark_Y[j] += SimMeasure(attributes[i], attributes[i][j]);
                    }
                }

            }
            MessageBox.Show(String.Format("The number of selected objects:  {0}\n The
number of selected attributes:  {1}:", table.Rows.Count, table.Columns.Count));
        }

        private void showGraphToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Calc cl = new Calc();
            cl.Show();
        }



    }



    public class DataM
    {


       public  static Excel.Application obj = new Excel.Application();
       public   Thread td;
       static Excel.Workbook wrb;
       static Excel.Worksheet ObjWorkSheet = null;
       static Excel.Worksheet act;
       static string path = System.IO.Directory.GetCurrentDirectory()+@"\data.xlsx";
       static int Num=0;
       int num = 0;


       public List<string> attribute = new List<string>();

        string cell_name = "";
        int start = 0;
        int rec = 0;
        static object lk="1";


        static DataM()
        {
            if(!System.IO.File.Exists(path)) {MessageBox.Show("data.xlsx must exist in
current directory!");Environment.Exit(0);}
            wrb = obj.Workbooks.Open(path, 0, false, 5, "", "", false,
Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "", true, false, 0, true, false,
false);
            ObjWorkSheet = (Excel.Worksheet)wrb.Sheets[1];
```

```
            act = wrb.ActiveSheet;
        }


        public DataM(string name,int start,int rec)
        {
            num = Num++;
            cell_name = name;
            this.start = start;
            this.rec = rec;

            td = new Thread(Run);
            td.Start();

        }

        void Run()
        {
            List<string> ls = new List<string>();
            string m;
            for (int i = 0; i <= rec+start-1; i++)
            {
                lock(lk)
                {
                    m = ObjWorkSheet.Range[cell_name+(i+1).ToString()].Value.ToString();
                    if(m!=null)  attribute.Add(m);
                    if(i==0) i = start-1;
                }
                Thread.Sleep(10);
            }
        }

    }

}
```