گزارش پروژه درس داده‌کاوی

نام و نام خانوادگی

نگار فتحی

شماره دانشجویی

۹۷۷۲۳۱۳۷

هدف گزارش: تحلیل تراکم (در کدام طول و عرض جغرافیایی تراکم تاکسی‌ها بیشتر است؟)

گام ۱: نمونه‌گیری از مجموعه‌داده:

یک نمونه‌ی تصادفی به اندازه‌ی ۶۲۷۷ رکورد، از مجموعه‌داده گرفته شده است و sample.txt نامگذاری شده است.

این نمونه‌ی تصادفی حاصل اجرای برنامه‌ی C# زیر می‌باشد.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Sampling
{
   public static class Class
   {
      private static readonly Random rng = new Random();
      public static void Shuffle<T>(this IList<T> list)
      {
         int n = list.Count;
         while (n > 1)
         {
            n--;
            int k = rng.Next(n + 1);
            T value = list[k];
            list[k] = list[n];
            list[n] = value;
         }
      }
   }
}

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace Sampling
{
   class Program
```

```
    {
        static void Main(string[] args)
        {
            var dest = @"C:\Users\hamed\Desktop\sample.txt";
            var folderPath = @"C:\Users\hamed\Desktop\T-drive Taxi
Trajectories\release\taxi_log_2008_by_id";
            var allFiles = Directory.GetFiles(folderPath);
            var allData = new List<string>();
            foreach (var file in allFiles.Take(500))
            {
                var lines = File.ReadAllLines(file);
                var length = lines.Length;
                int percent = Convert.ToInt32(0.01 * length);
                lines.Shuffle();
                var newList = lines.Take(percent).ToList();
                allData.AddRange(newList);
                Console.WriteLine(Path.GetFileNameWithoutExtension(file));
            }
            allData.Shuffle();
            File.WriteAllLines(dest, allData);
        }
    }
}
```

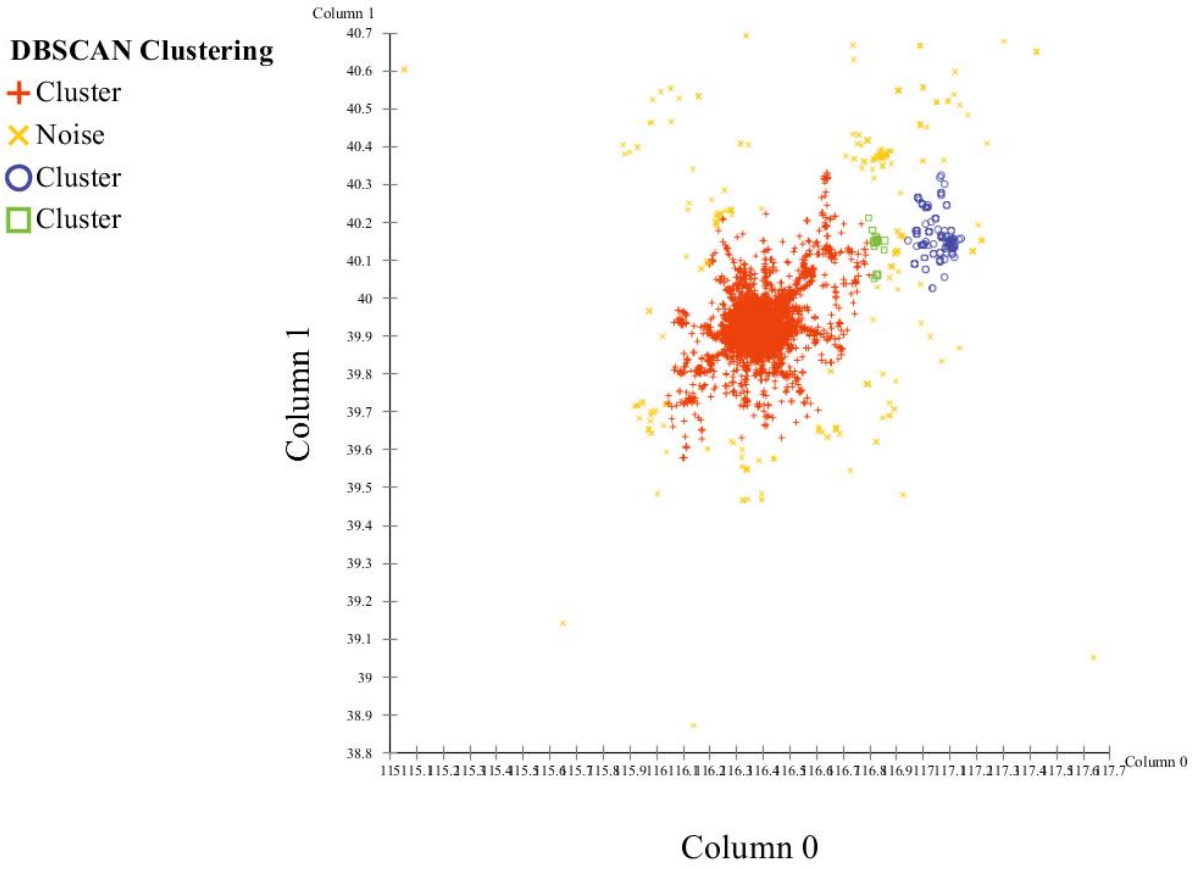
گام ۲: اجرای sample.txt در ELKI

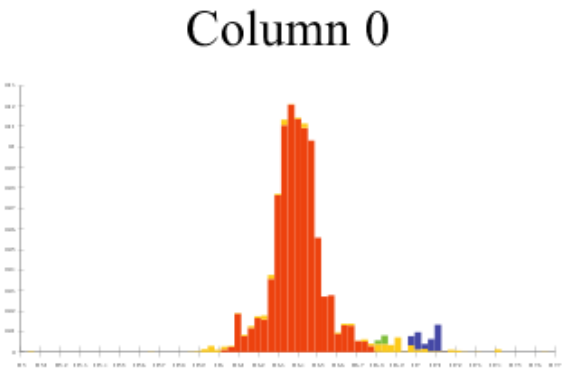
```
KDDCLIApplication -dbc.in "C:\\Users\\hamed\\Desktop\\sample.txt" -
parser.labelIndices 0 -dbc.filter transform.ProjectionFilter -projection
NumericalFeatureSelection -projectionfilter.selectedattributes 2,3 -time -
algorithm clustering.DBSCAN -algorithm.distancefunction
geo.LatLngDistanceFunction -dbscan.epsilon 5000.0 -dbscan.minpts 50 -
resulthandler ResultWriter,AutomaticVisualization -out
"C:\\Users\\hamed\\Desktop\\out" -out.silentoverwrite
```

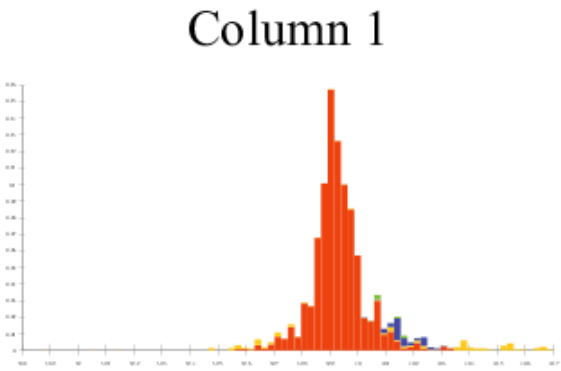
دستورات	توضیحات
-dbc.in "C:\\Users\\hamed\\Desktop\\sample.txt"	مسیر فایل ورودی

-parser.labelIndices 0 -dbc.filter transform.ProjectionFilter -projection NumericalFeatureSelection -projectionfilter.selectedattributes 2,3	از آن‌جایی که رکوردهای مجموعه‌داده دارای ۴ بعد بودند ولی ما نیازمند تنها ۲ بعد longitude و latitude بودیم این دستورات اعمال شدند.
-time	
-algorithm clustering.DBSCAN	الگوریتم خوشه‌بندی DBSCAN نیازمند تعیین تعداد خوشه‌ها نبوده (البته نیازمند تعیین دو پارامتر epsilon و minpts است) و خوشه‌هایی با اشکال دلخواه تولید می‌کند و یکی از گزینه‌های خوب برای خوشه‌بندی داده‌های مکانی_زمانی است.
-algorithm.distancefunction geo.LatLngDistanceFunction	با توجه به مجموعه‌داده (حاوی داده‌های مکانی_زمانی)، این تابع فاصله مناسب‌تر از سایرین به نظر می‌آمد.
-dbscan.epsilon 5000.0	پس از دادن مقادیر مختلف به epsilon و مشاهده نتایج، این مقدار برای epsilon مناسب‌تر از سایرین به نظر می‌آمد.
-dbscan.minpts 50	پس از دادن مقادیر مختلف به minpts و مشاهده نتایج، این مقدار برای minpts مناسب‌تر از سایرین به نظر می‌آمد.
-resulthandler ResultWriter,AutomaticVisualization	دو مدل برای نمایش نتیجه
-out "C:\\Users\\hamed\\Desktop\\out"	مسیر فایل نتیجه
-out.silentoverwrite	

گام ۳: خروجی اجرای sample.txt در ELK







گام ۴: نتیجه

تراکم تاکسی‌ها در مختصات جغرافیایی زیر بیشتر از سایرین است.

longitude = ۱۱۶ تا ۱۱۶,۸

latitude = ۳۹,۶ تا ۴۰,۳