# پروژه ارشد هوش مصنوعی

عنوان: خوشه­بندی داده­ها روی نقشه – فاز 2

دانشجو: صدرا صمدی طاهرگورابی

شماره دانشحویی: 990197451

استاد: مجید ایرانپور مبارکه

دانشگاه: پیام نور واحد نجف­آباد

پاییز 1401

# مقدمه

در فاز اول، داده­ها را به پنجره­های یک در یک تقسیم، الگوریتم خوشه­بندی K-Means را روی آنها اجرا و در نقشه نمایش دادیم.

در فاز دوم میخواهیم الگوریتم خوشه­بندی را روی خود داده­ها طوری پیاده کنیم که 1- مرکز خوشه­ها در خیابان اصلی قرار بگیرد و 2- فاصله هر نقطه از مرکز کمتر از یک کیلومتر باشد.

این پروژه شامل دو بخش Backend و Frontend است.

# توضیحات بخش Backend

در بخش Backend که توسط زبان برنامه­نویسی پایتون نوشته شده است مراحل اصلی یعنی پردازش داده­ها و اجرای الگوریتم خوشه­بندی انجام می­شود که به شرح زیر است:

1. آرگومان­های برنامه که طریقه استفاده از آن­ها در ادامه توضیح داده خواهد شد را از خط فرمان میخوانیم.
2. داده­های موجود در فایل ورودی که شامل مختصات هر نمونه است را به صورت یک جدول بارگذاری می­کنیم (از آنجایی که تعداد نمونه­ها و اطلاعات مربوط به نقشه آن­ها زیاد و پردازش آن­ها کمی زمان­بر است، داده­های مربوط به شهر نجف­آباد در فایل جدا قرار داده شده­اند تا برای آزمایش الگوریتم­ها از آن استفاده شود، همچنین امکان انتخاب کسری از داده­ها به صورت تصادفی وجود دارد که هر دو در آرگومان­های برنامه مشخص می­شوند).
3. گراف نقشه که از پلتفرم OpenStreetMap دانلود شده و شامل مختصات تمام خیابان­های اصلی برای ناحیه­ای که نمونه­ها در آن قرار دارند است را بارگذاری می­کنیم.
4. مختصات نزدیکترین یال (خیابان اصلی) و فاصله آن تا هر نمونه را جهت استفاده در الگوریتم به صورت جداگانه پیدا می­کنیم.
5. داده­ها را توسط الگوریتم انتخاب شده در آرگومان برنامه خوشه­بندی و مختصات مرکز را پیدا می­کنیم (شش الگوریتم مختلف پیاده­سازی شده که به هرکدام جداگانه خواهیم پرداخت).
6. جهت ارزیابی، فاصله مرکز هر خوشه تا نمونه­های آن و نزدیکترین یال را محاسبه میکنیم.
7. مقادیر محاسبه شده را در خروجی چاپ میکنیم.
8. نتیجه نهایی شامل نمونه­ها و اطلاعات خوشه را در یک فایل خروجی ذخیره می­کنیم.

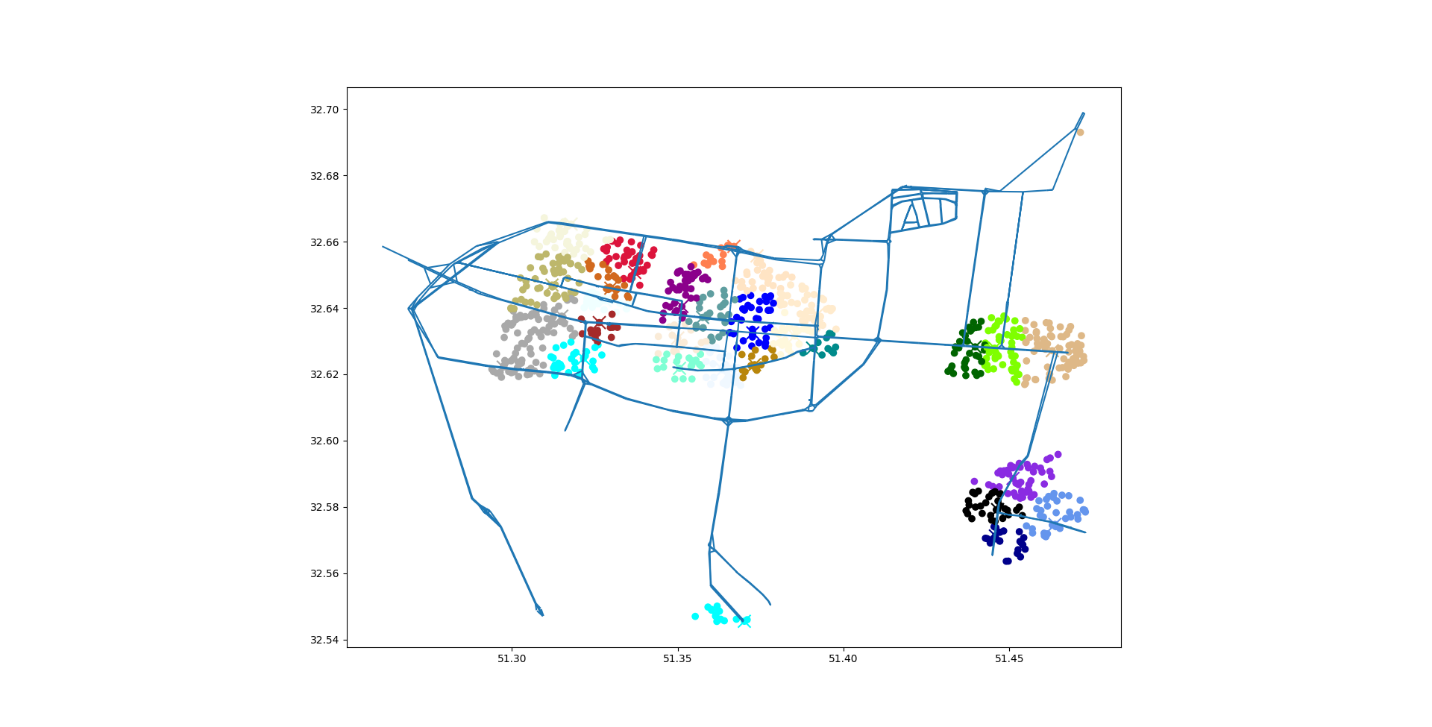
## توضیحات الگوریتم

برای حل مسأله موجود، الگوریتم­های مختلف خوشه­بندی بررسی شده­اند که هر یک مزایا و معایب خود را دارند، از میان آن­ها شش الگوریتم در پروژه پیاده­سازی و نتیجه آن­ها ارزیابی شده است. جدول زیر شامل نتایج اجرای هر الگوریتم روی داده­های شهر نجف­آباد است:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| الگوریتم | | مقدار پارامتر اصلی | تعداد خوشه یافت شده | زمان اجرای الگوریتم (ms) | میانگین فاصله هر مرکز تا نمونه­های آن (m) | میانگین فاصله هر مرکز تا نزدیکترین یال آن (m) |
| 1 | Affinity Propagation | 1000 | 29 | 780 | 627.514 | 7.070 |
| 2 | K-Means | 30 | 30 | 1090 | 483.964 | 334.384 |
| 3 | Mean-Shift | 0.05 | 28 | 3230 | 547.260 | 295.327 |
| 4 | Agglomerative Clustering | 1 | 41 | 190 | - | - |
| 5 | DBSCAN | 0.5 | 6 | 100 | - | - |
| 6 | BIRCH | 0.05 | 24 | 830 | 574.315 | 284.769 |

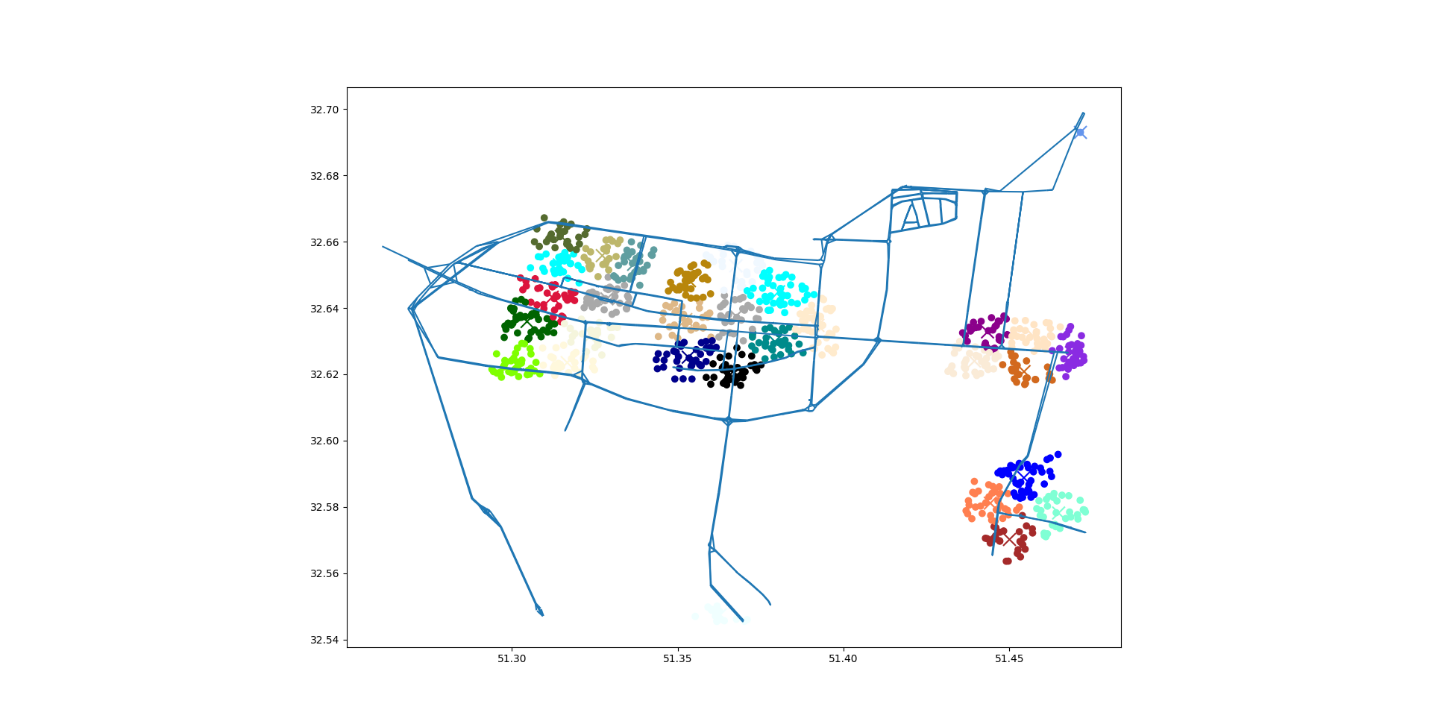
### 1. الگوریتم Affinity Propagation

این الگوریتم به عنوان الگوریتم اصلی برنامه و بهترین الگوریتم موجود برای حل مسأله موردنظر انتخاب شده است. تفاوت اساسی آن با سایر الگوریتم­ها وجود پارامتری است که امکان وزن­دهی به نمونه­ها جهت انتخاب برای مرکز خوشه را در اختیار ما قرار می­دهد و همانطور که در جدول بالا مشخص است کمترین فاصله مرکز تا یال توسط همین الگوریتم به دست آمده است. در این الگوریتم نمونه­ها با انتشار پیام به یکدیگر و ملاک قرار دادن یک نمونه به عنوان مرکز به همگرایی رسیده و خوشه­بندی می­شوند. ورودی این الگوریتم ماتریس فاصله نمونه­ها است که توسط فرمول Haversine از روی مختصات جغرافیایی آن­ها به واحد کیلومتر ساخته شده. برای وزن­دهی به نمونه­ها جهت انتخاب مرکز نیز از فاصله نزدیکترین یال که در مراحل قبل محاسبه شده و با یک ضریبی که در آرگومان برنامه مشخص شده است استفاده می­کنیم. اندازه خوشه ها با ضریب اشاره شده رابطه مستقیم دارد، یعنی با افزایش آن مساحت خوشه و نمونه­های موجود در آن افزایش می­یابد و بالعکس. با مقداردهی مناسب به این ضریب می­توان نتیجه مطلوب مسأله را به­دست آورد. تصویر زیر نشانگر خوشه­ها و مراکز یافت شده توسط الگوریتم مذکور روی نقشه است:



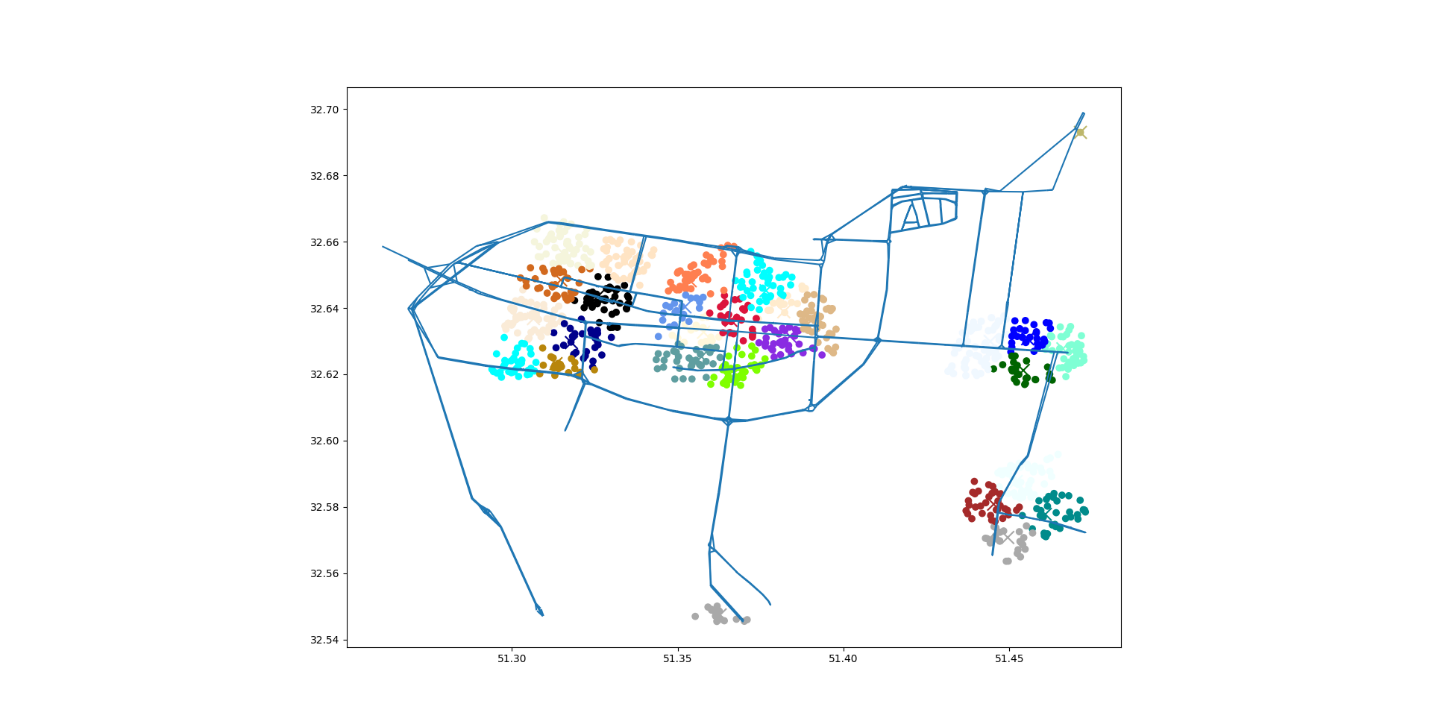
### 2. الگوریتم K-Means

این الگوریتم یکی از پرکاربردترین الگوریتم­های مبتنی بر مرکز در مبحث خوشه­بندی است که در فاز یک پروژه نیز مورد استفاده قرار گرفته بود اما به دلیل اینکه پارامتری جهت انتخاب مرکز خوشه ندارد انتخاب مناسبی برای مسأله ما نیست. ورودی این الگوریتم مختصات جغرافیایی نرمال شده داده­ها و پارامتر اصلی آن که توسط آرگومان برنامه مشخص می­شود تعداد خوشه­ها است.



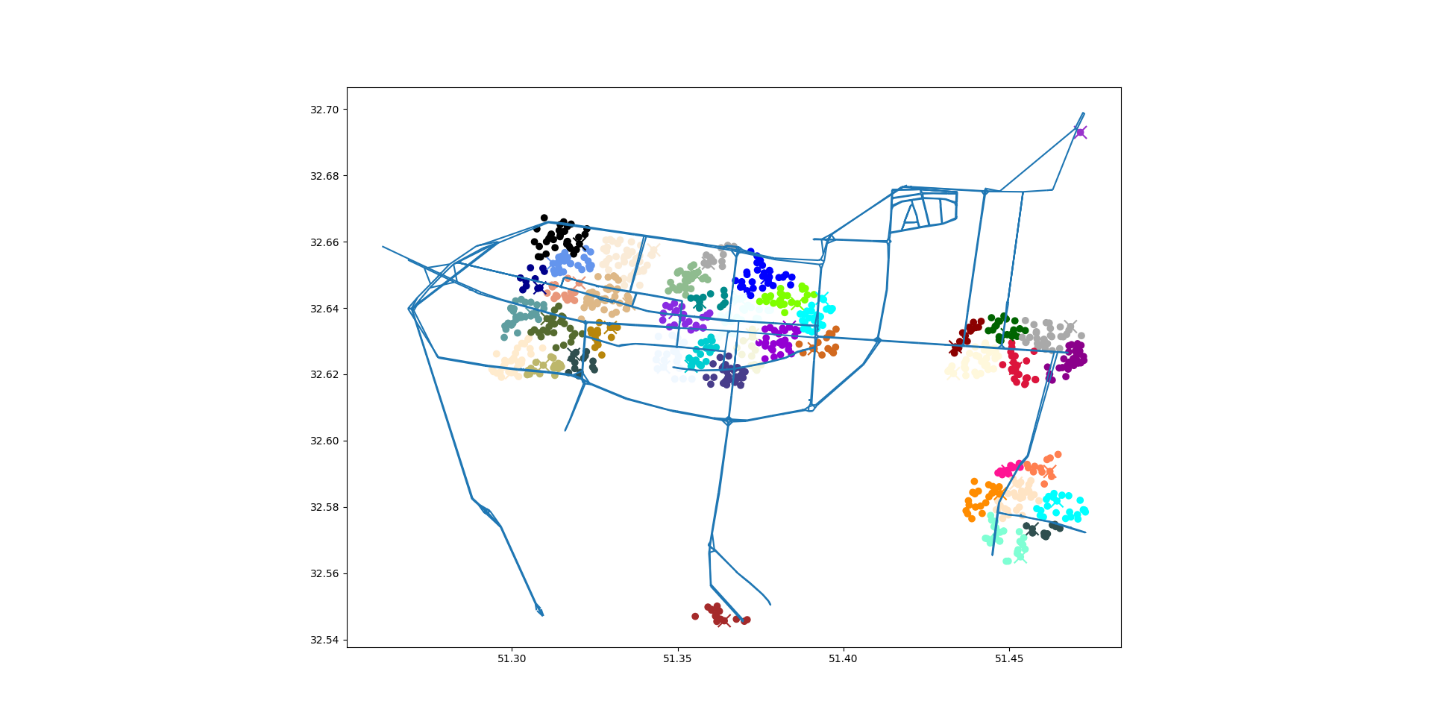
### 3. الگوریتم Mean-Shift

این الگوریتم که مبتنی بر مرکز است در هر تکرار سعی می­کند نمونه­ها را به سمتی که تراکم بیشتری دارد میل دهد تا خوشه­ها یافت شوند. به دلیل عدم وجود پارامتری برای انتخاب مرکز این الگوریتم هم برای ما قابل استفاده نیست. ورودی این الگوریتم مختصات جغرافیایی نرمال شده داده­ها و پارامتر اصلی که توسط آرگومان برنامه مشخص می­شود پهنای­باند یا شعاعی است که داده­ها مراکز متراکم را جستجو می­کنند.



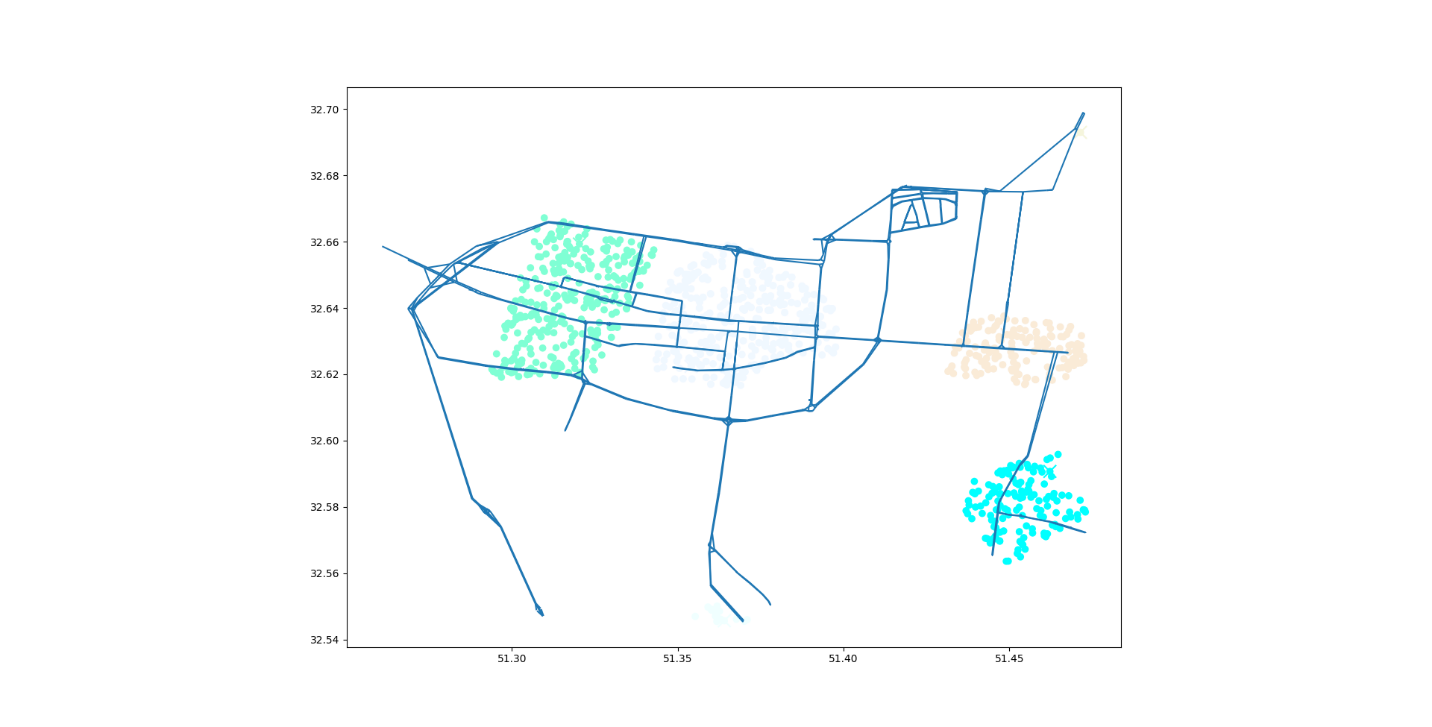
### 4. الگوریتم Agglomerative Clustering

این الگوریتم مبتنی بر سلسله­مراتب با رویکرد پایین به بالا است که با ساختن یک درخت از نمونه­ها و ادغام گره­های نزدیک به­هم می­تواند در سطوح مختلف عمل خوشه­بندی را انجام دهد. بزرگترین عیب این الگوریتم نداشتن مرکز خوشه است پس برای مسأله ما قابل استفاده نیست. ورودی الگوریتم، ماتریس فاصله نمونه­ها است که توسط فرمول Haversine از روی مختصات جغرافیایی آن­ها به واحد کیلومتر ساخته شده و پارامتر اصلی که توسط آرگومان برنامه مشخص می­شود حداکثر فاصله مجاز خوشه­ها جهت ادغام است.



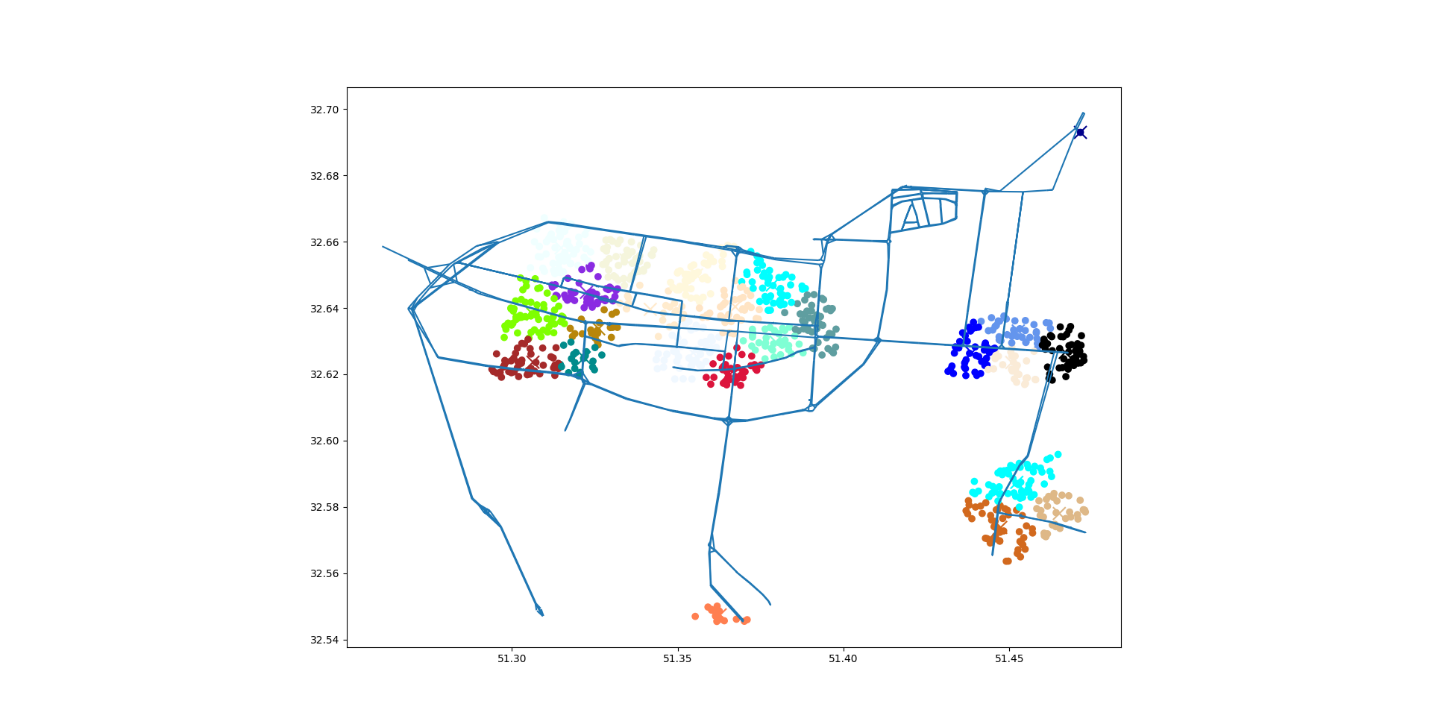
### 5. الگوریتم DBSCAN

این الگوریتم مبتنی بر چگالی است و برای یافتن خوشه­ها براساس تراکم داده ها خوب عمل می­کند ولی برای مسأله ما اصلاً مناسب نیست چون داده­های ما در بعضی مناطق بسیار متراکم هستند و امکان جداسازی آن­ها توسط این الگوریتم وجود ندارد و همچنین خوشه­های یافت شده در این الگوریتم مرکز ندارند. ورودی الگوریتم، ماتریس فاصله نمونه­ها است که توسط فرمول Haversine از روی مختصات جغرافیایی آن­ها به واحد کیلومتر ساخته شده و پارامتر اصلی که توسط آرگومان برنامه مشخص می­شود حداقل فاصله بین خوشه­ها است.



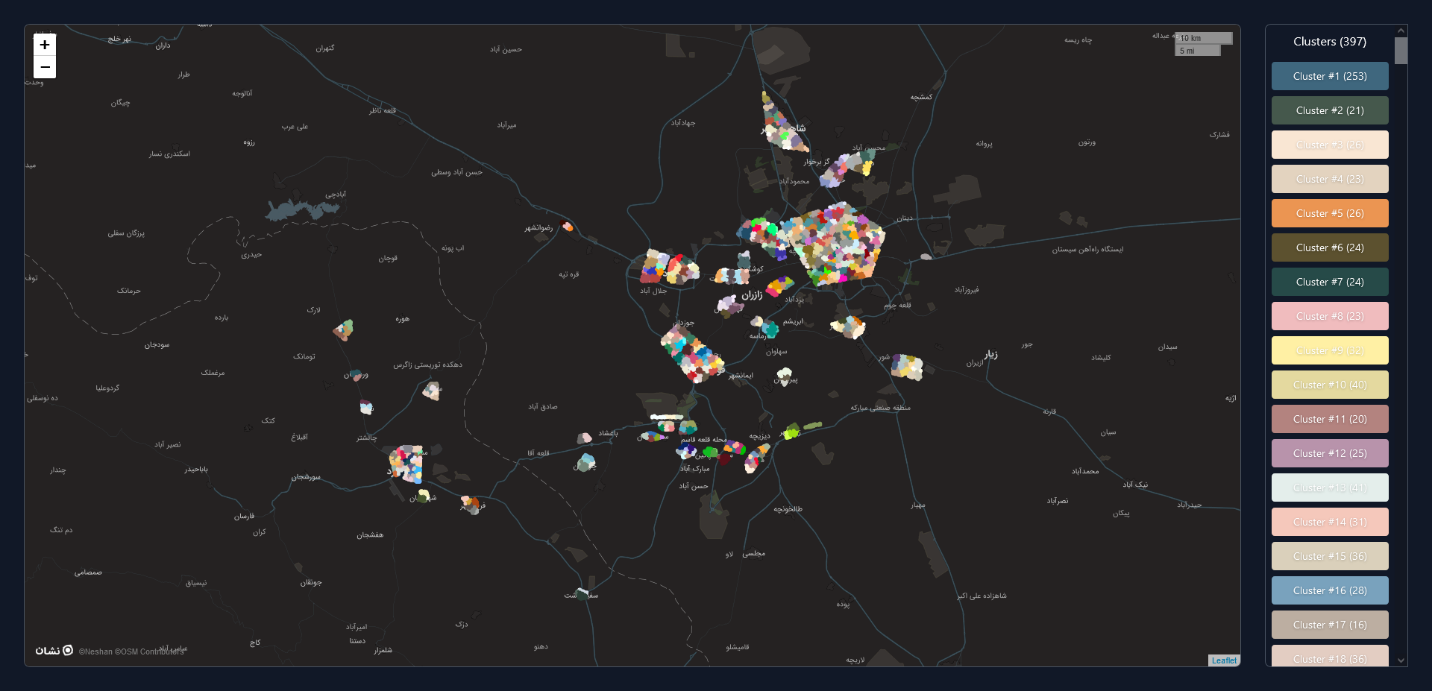
### 6. الگوریتم BIRCH

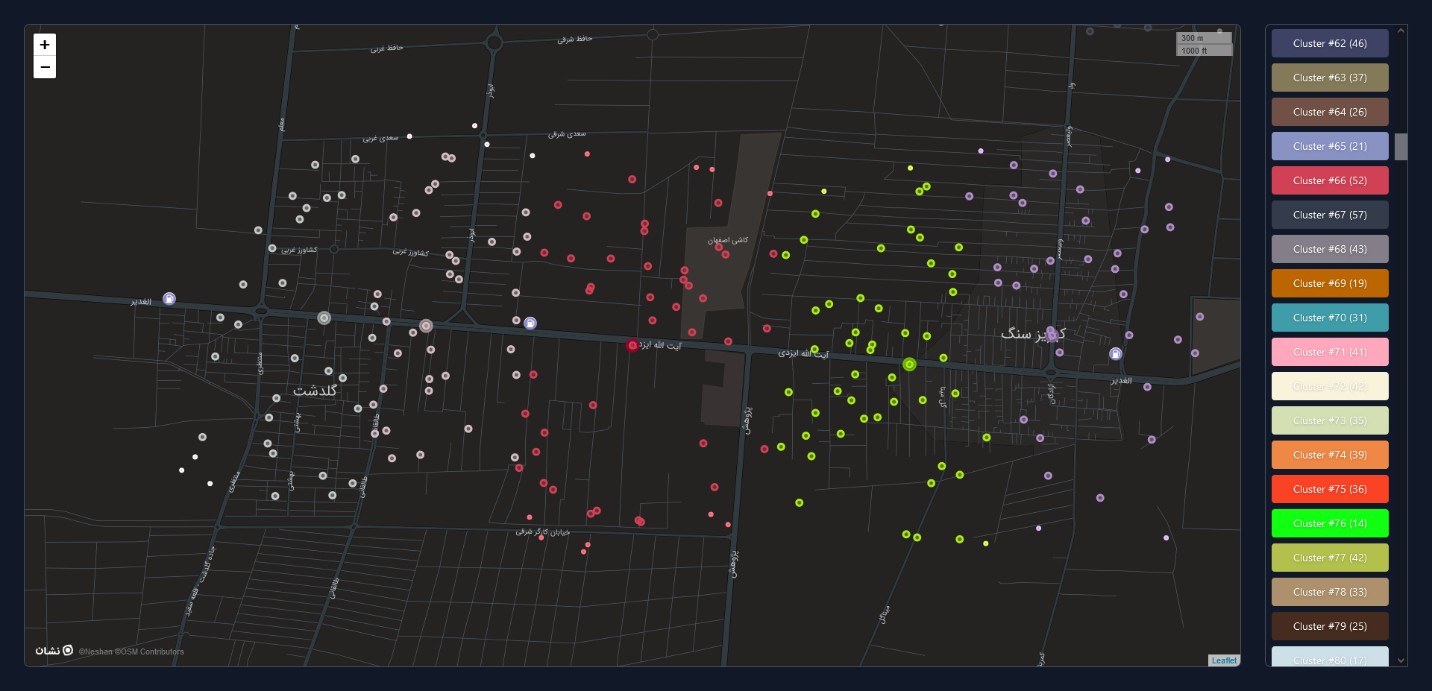
این الگوریتم هم مبتنی بر سلسله­مراتب و بسیار شبیه به الگوریتم Agglomerative Clustering است. تفاوت اصلی این الگوریتم بهینه بودن آن برای داده­های حجیم است زیرا میتواند ابتدا آن­ها را به زیرمجموعه­های کوچکتر تقسیم و سپس خوشه­بندی کند، همچنین خوشه­ها در این الگوریتم دارای مرکز هستند. اما همچنان مانند سایر الگوریتم­ها به دلیل نداشتن پارامتر جهت انتخاب مرکز برای مسأله ما قابل استفاده نیست. ورودی این الگوریتم مختصات جغرافیایی نرمال شده داده­ها و پارامتر اصلی که توسط آرگومان برنامه مشخص می­شود حداکثر فاصله مجاز خوشه­ها جهت ادغام است.



توضیحات بخش Frontend

در بخش Frontend که به صورت یک وب اپ و به کمک کتابخانه ReactJS طراحی شده است، اطلاعات موجود در فایل خروجی تولید شده در بخش Backend را با یک رابط کاربری ساده نمایش می­دهیم.





همانطور که ملاحظه می­کنید هر خوشه با رنگ متفاوت و نمونه­های آن نمایش داده می­شود.

با نگه داشتن نشانگر موس روی نمونه­ها شماره خوشه آن نشان داده می­شود.

مراکز خوشه با رنگ تیره­تر و اندازه بزرگتر و داده­هایی که فاصله آن­ها از مرکز بیش از یک کیلومتر باشد با رنگ روشن­تر و اندازه کوچکتر متمایز شده­اند.

در سمت راست نیز تعداد کل خوشه­ها، شماره هر خوشه و تعداد نمونه موجود در آن مشخص شده که با کلیک بر روی هریک به مرکز خوشه منتقل می­شوید.

# راه­اندازی بخش Backend

برای این بخش نیاز است که نسخه 3.8 یا بالاتر نرم­افزار Pyhton و pip روی سیستم شما نصب باشد.

<https://www.python.org/>

کدهای اصلی این قسمت در مسیر /src/app/main.py و فایل­های ورودی و خروجی در مسیر /data قرار داده شده­اند.

جهت اجرا مراحل زیر را به ترتیب انجام دهید:

1. یک خط فرمان (command line, cmd, terminal, powershell, …) را دقیقا در مسیر پوشه /backend اجرا کنید.
2. دستورات زیر را جهت ساخت یک محیط مجازی و نصب پکیج­های استفاده شده در پروژه اجرا کنید:

python -m venv .venv

.\.venv\Scripts\activate

pip install poetry

poetry install

1. منتظر بمانید تا پکیج ها دریافت و نصب شوند (اولین اجرا ممکن است کمی زمان­بر باشد).
2. جهت اجرای برنامه دستور زیر را وارد کنید:

poetry run main

1. آرگومان­های ورودی برنامه در جدول زیر تعریف شده­اند:

|  |  |  |  |
| --- | --- | --- | --- |
| نام آرگومان | مقادیر قابل قبول | مقدار پیش­فرض | توضیحات |
| dataset | {original, njfb} | njfb | از کل داده­های موجود (original) استفاده شود یا فقط از داده های موجود در شهر نجف­آباد (njfb) |
| fraction | (0.0, 1.0] | 1.0 | چه کسری از نمونه­های موجود به صورت تصادفی انتخاب شوند |
| download | - | غیرفعال | آیا نیاز است که فایل گراف نقشه از OpenStreetMap دوباره دانلود شود یا خیر |
| validate | - | غیرفعال | آیا نمونه­هایی که فاصله آن­ها از نزدیکترین خیابان اصلی بیش از یک کیلومتر است حذف شوند یا خیر |
| algorithm | {1, 2, 3, 4, 5, 6} | 1 | شماره الگوریتم مورد نظر |
| param | عدد حقیقی | وابسته به الگوریتم | پارامتر اصلی الگوریتم که بسته به نوع آن مقیاس متفاوتی دارد |

1. برای مشخص کردن آرگومان­ها در دستور اجرای برنامه به مثال زیر توجه کنید:

poetry run main --dataset original --fraction 0.5 --algorithm 3 --param 10.0 --validate

1. اگر همه مراحل بدون مشکل اجرا شوند، یک فایل خروجی در مسیر /data/{dataset}/output.xlsx ساخته می­شود.

# راه­اندازی بخش Frontend

برای این بخش نیاز است که یکی از نسخه­های (ترجیحا به­روز) نرم­افزار Node.JS و npm روی سیستم شما نصب باشد.

<https://nodejs.org/en/>

کدهای اصلی این قسمت در مسیر /src/app.tsx قرار داده شده­اند.

جهت اجرا مراحل زیر را به ترتیب انجام دهید:

1. یک خط فرمان (command line, cmd, terminal, powershell, …) را دقیقا در مسیر پوشه /frontend اجرا کنید.
2. دستور زیر را جهت نصب پکیج های استفاده شده در پروژه اجرا کنید:

npm install

1. منتظر بمانید تا پکیج ها دریافت و نصب شوند (اولین اجرا ممکن است کمی زمان بر باشد).
2. فایل خروجی که در بخش Backend ساخته شد را در مسیر /frontend/public/output.xlsx کپی کنید.
3. دستور زیر را اجرا کنید:

npm run start

1. پس از چند ثانیه وارد آدرس زیر در مرورگر خود شوید:

<http://localhost:3000/>

1. اگر همه مراحل بدون مشکل اجرا شوند، رابط کاربری برنامه در صفحه مرورگر نمایش داده می­شود.
2. درصورت نیاز جهت بروزرسانی فایل خروجی، فایل جدید را جایگزین فایل قبلی کرده و صفحه مرورگر را به­روز (Refresh) کنید.
3. برای بستن برنامه در خط فرمان دکمه های Ctrl و C کیبورد را دوبار به صورت همزمان فشار دهید.