پروژه ارشد هوش مصنوعی

عنوان: خوشه‌بندی داده‌ها روی نقشه – فاز ۲

دانشجو: صدرا صمدی طاهرگورابی

شماره دانشجویی: ۹۹۰۱۹۷۴۵۱

استاد: مجید ایرانپور مبارکه

دانشگاه: پیام نور واحد نجف‌آباد

پاییز ۱۴۰۱

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 ۱- مرکز خوشه‌ها در خیابان اصلی قرار بگیرد و ۲- 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:

۱. آرگومان‌های برنامه که طریقه استفاده از آن‌ها در ادامه توضیح داده خواهد شد را از خط فرمان میخوانیم.
۲. 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).
۳. 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.
۴. مختصات نزدیکترین یال (خیابان اصلی) و فاصله آن تا هر نمونه را جهت استفاده در الگوریتم به صورت جداگانه پیدا می‌کنیم.
۵. 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).

۶. جهت ارزیابی، فاصله مرکز هر خوشه تا نمونه‌های آن و نزدیکترین یال را محاسبه میکنیم.
۷. مقادیر محاسبه شده را در خروجی چاپ میکنیم.
۸. نتیجه نهایی شامل نمونه‌ها و اطلاعات خوشه را در یک فایل خروجی ذخیره می‌کنیم.

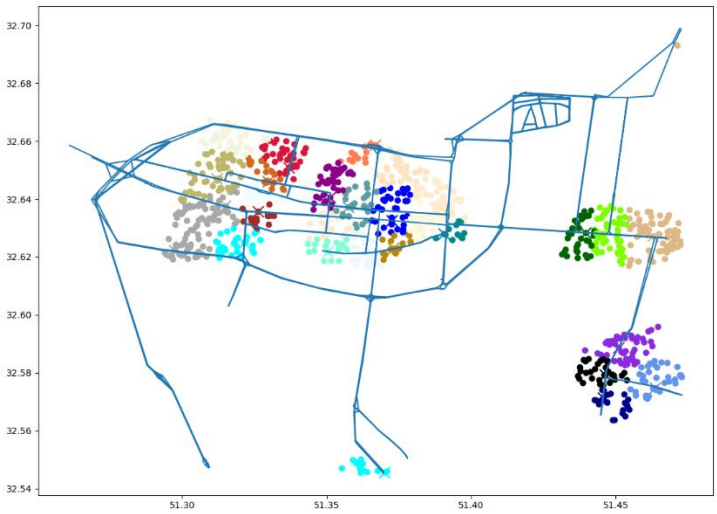
توضیحات الگوریتم

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)
۱	Affinity Propagation	۱۰۰۰	۲۹	۷۸۰	۶۲۷.۵۱۴	۷.۰۷۰
۲	K-Means	۳۰	۳۰	۱۰۹۰	۴۸۳.۹۶۴	۳۳۴.۳۸۴
۳	Mean-Shift	۰.۰۵	۲۸	۳۲۳۰	۵۴۷.۲۶۰	۲۹۵.۳۲۷
۴	Agglomerative Clustering	۱	۴۱	۱۹۰	-	-
۵	DBSCAN	۰.۵	۶	۱۰۰	-	-
۶	BIRCH	۰.۰۵	۲۴	۸۳۰	۵۷۴.۳۱۵	۲۸۴.۷۶۹

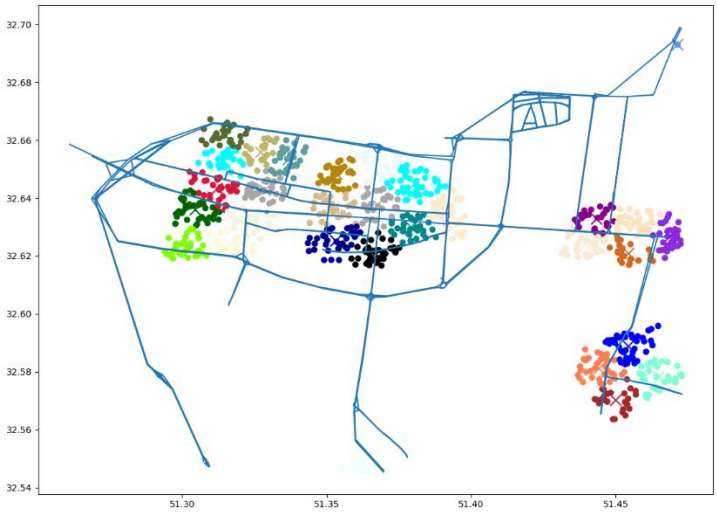
۱. الگوریتم Affinity Propagation

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:



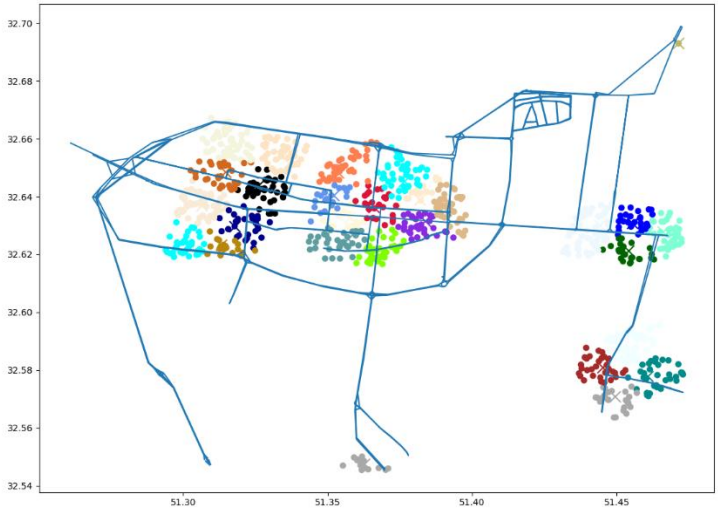
۲. الگوریتم K-Means

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.



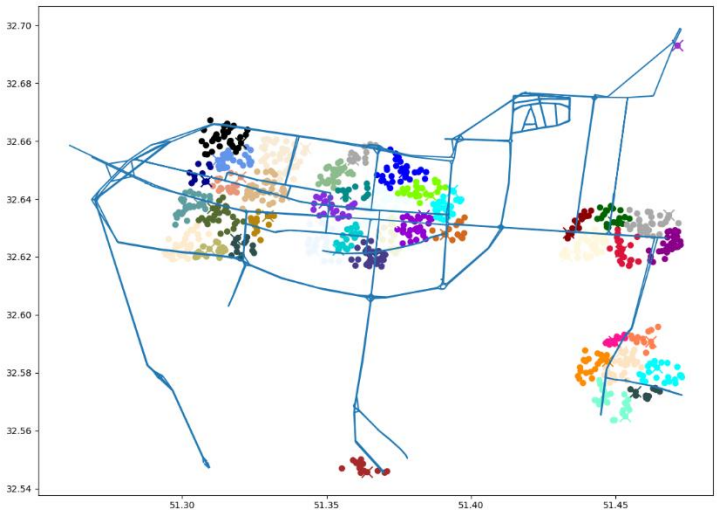
۳. الگوریتم Mean-Shift

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.



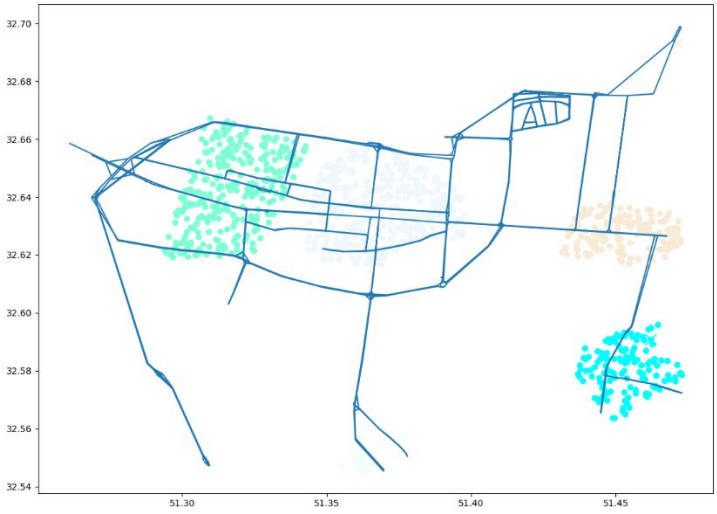
۴. الگوریتم Agglomerative Clustering

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.



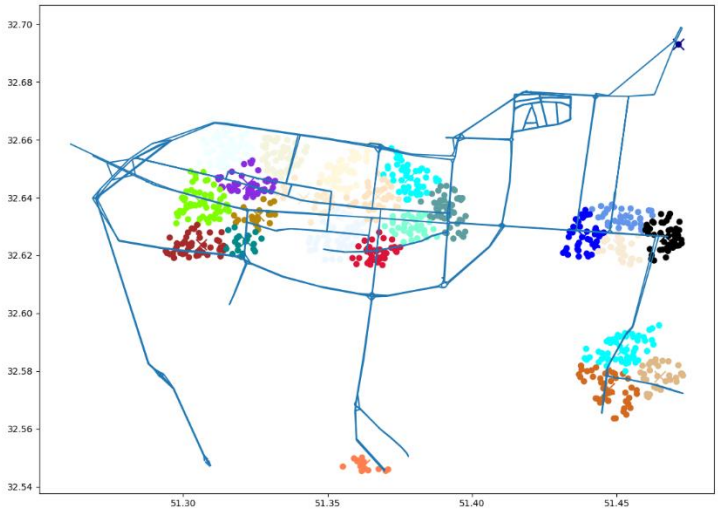
۵. الگوریتم DBSCAN

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.



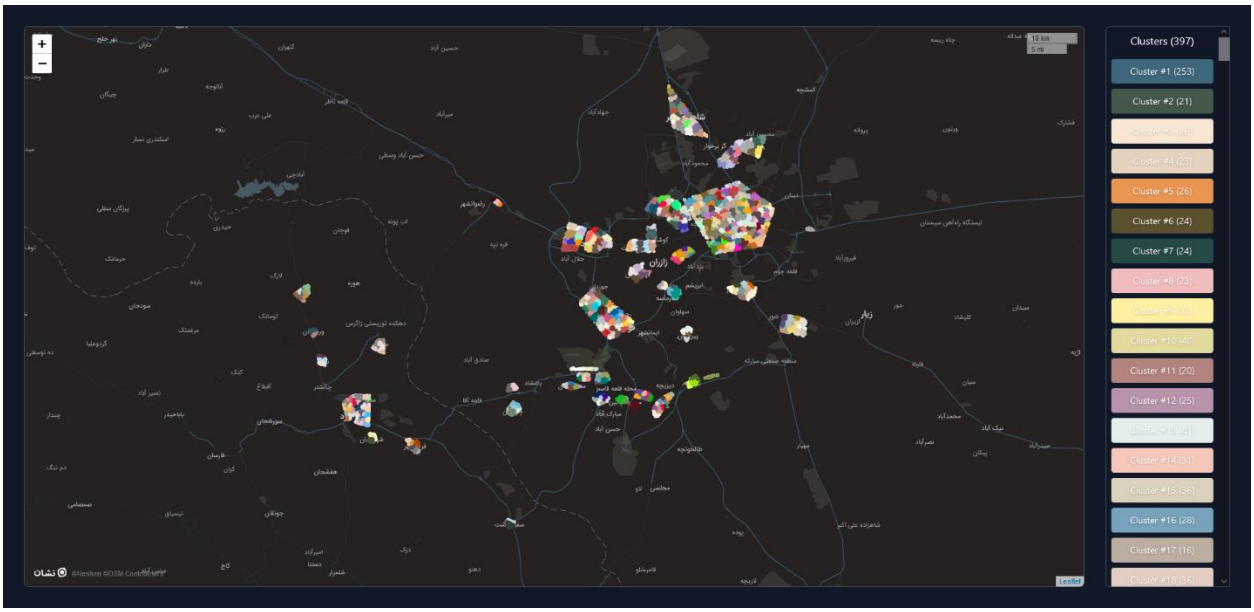
۶. الگوریتم BIRCH

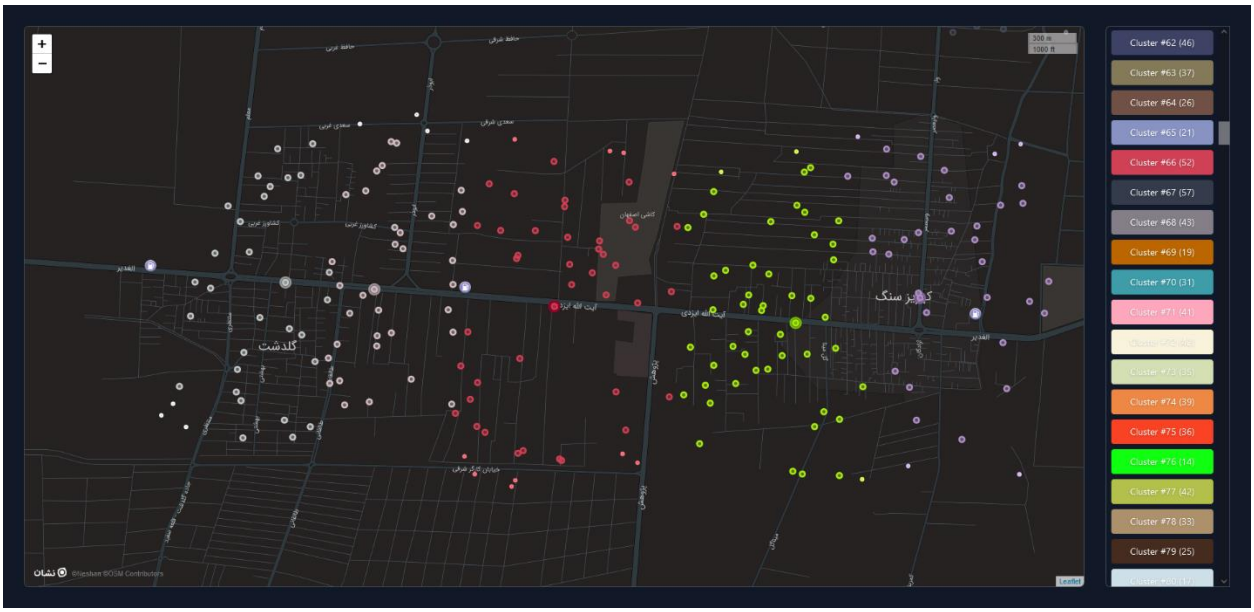
این الگوریتم هم مبتنی بر سلسله‌مراتب و بسیار شبیه به الگوریتم Agglomerative Clustering است. تفاوت اصلی این الگوریتم بهینه بودن آن برای داده‌های حجیم است زیرا میتواند ابتدا آن‌ها را به زیرمجموعه‌های کوچکتر تقسیم و سپس خوشه‌بندی کند، همچنین خوشه‌ها در این الگوریتم دارای مرکز هستند. اما همچنان مانند سایر 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 ۳.۸ 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:

۱. یک خط فرمان (command line, cmd, terminal, powershell, ...) را دقیقا در مسیر پوشه `/backend` اجرا کنید.
۲. 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
```

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

```
poetry run main
```

۵. 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

۶. 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
```

۷. 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:

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

```
npm install
```

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

```
npm run start
```

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

http://localhost:3000/

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