بسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی عمران

تمرین سری اول درس حمل و نقل هوشمند

نگارنده

علی پیرمحمدی

استاد درس

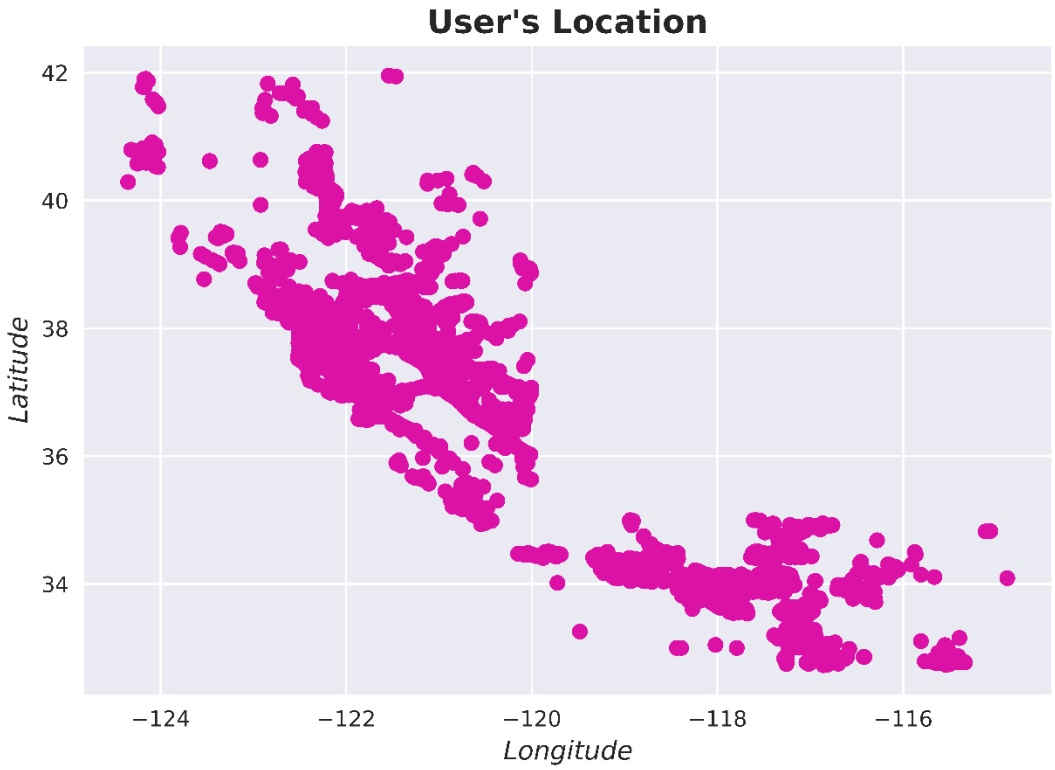
دکتر زهرا امینی

فهرست مطالب

فهرست شکل ها

۱. مصورسازی داده‌ها

۱.۱. مصورسازی داده مکان کاربران



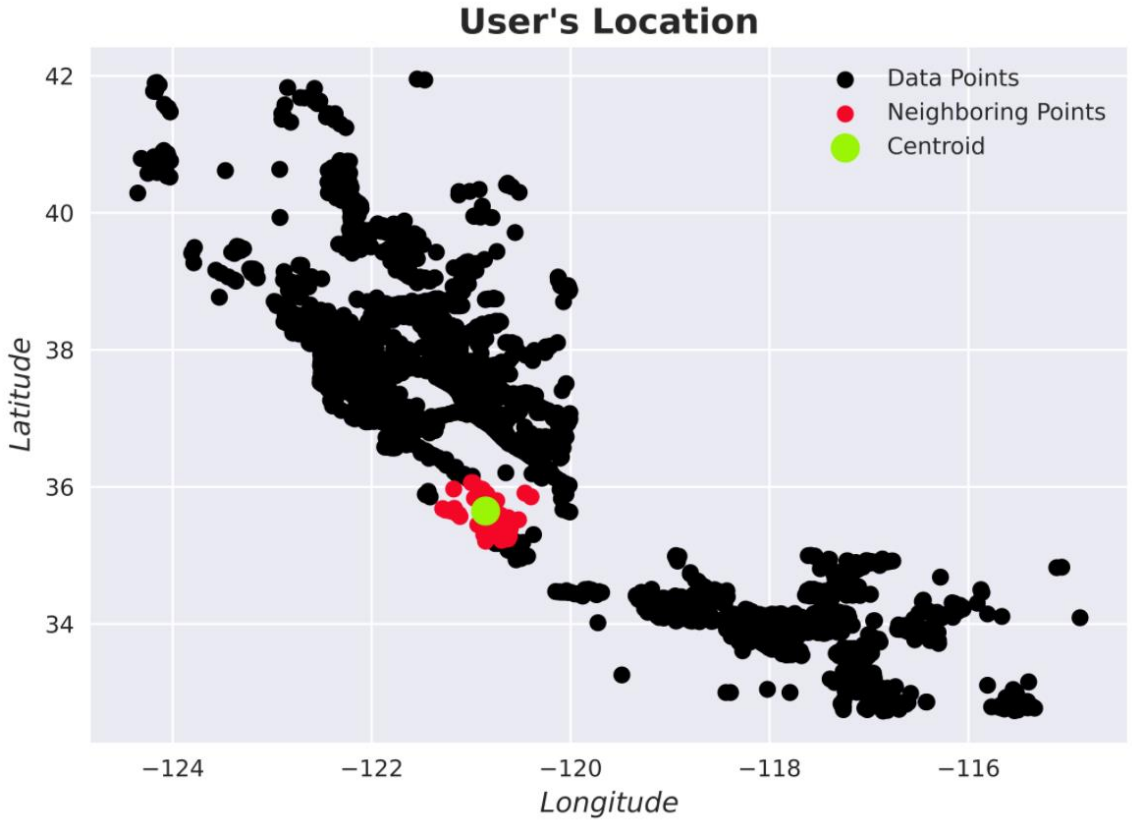
شکل ۱. مصورسازی داده مکان کاربران

۱.۲. توضیحات مربوط به شکل ۱

شکل۱ به نوعی شکل هندسی ایالت کالیفرنیا آمریکا را نشان می‌دهد، که محدوده‌هایی که تعداد نقاط داده متمرکز هستند در واقع شهرها و مناطق شهری پر جمعیت را نشان می‌دهد که کاربران توییتر بیشتری در این مناطق حضور دارند و تعداد پست هایی که از این مناطق ارسال شده است بیشتر از نواحی دیگر ایالت کالیفرنیا می‌باشد. زمان اجرای کد مربوط به این نمودار ۰.۳۶۵۹۱۶۷۲۸۹۷۳۳۸۸۶۷ ثانیه است که در کد نیز مشخص شده‌است.

۲. محاسبه چگالی نقاط

در این قسمت با توجه به رقم آخر شماره دانشجویی مرکز دوم انتخاب می‌شود و فاصله تمام نقاط با مرکز به کمک تابع هاورساین محاسبه شده است که در ستونی به نام 'dist from centroid' این فواصل به کیلومتر ذخیره شده است. تعداد نقاط در همسایگی ۵۰ کیلومتری این مرکز ۲۴۹ می‌باشد.در شکل ۲ مرکز و نقاط همسایه آن مشخص شده است.



شکل۲. مرکز و نقاط همسایگی آن به شعاع ۵۰ کیلومتر

۳. خوشه بندی

۳.۱. الگوریتم K-Means

در این بخش ابتدا به منظور افزایش دقت در مراحل خوشه بندی مکان داده‌ها در سیستم مختصات فرضی دیگری تصویر شده است. که مبدا این سیستم مختصاتی بر اساس کمترین مقدار طول و عرض جغرافیایی نقاط موجود تعیین شده‌است و فاصله هر یک از نقاط از این مبدا فرضی به کمک تابع هاورساین محاسبه شده است . این فواصل بر حسب هزار کیلومتر (یک میلیون متر) محاسبه شده‌اند و در سیستم مختصات جدید طول و عرض هر یک از نقاط اعدادی بین بازه ۰ تا ۱ است. توجه شود که این بازه به منظور درک بهتر از نتایج روش های خوشه بندی و افزایش سرعت الگوریتم انتخاب شده است.

۳.۱.۱. اجرا الگوریتم K-Means و زمان اجرا آن

کدی برای استفاده از روش خوشه بندی k-means با فرض تعداد خوشه‌ها برابر با ۱۰ نوشته شده است. همچنین زمان اجرا این روش با تعداد ۱۰۰ خوشه از طریق دو روش محاسبه شده‌است. با توجه به این که در هر تکرار روش خوشه بندی k-means با توجه به انتخاب تصادفی نقاط اولیه ممکن است زمان اجرا متفاوت باشد و جواب‌های مختلفی نیز محاسبه شود، روش دوم برای محاسبه زمان اجرا پیشنهاد بهتری است چرا که روش چندین بار انجام می‌شود و زمان اجرا متوسط گزارش می‌شود. زمان اجرا با تعداد ۱۰۰ خوشه به طور متوسط $11.1\ s\ \pm 417\ ms$ است.

۳.۱.۲. حداکثر تعداد خوشه‌ها

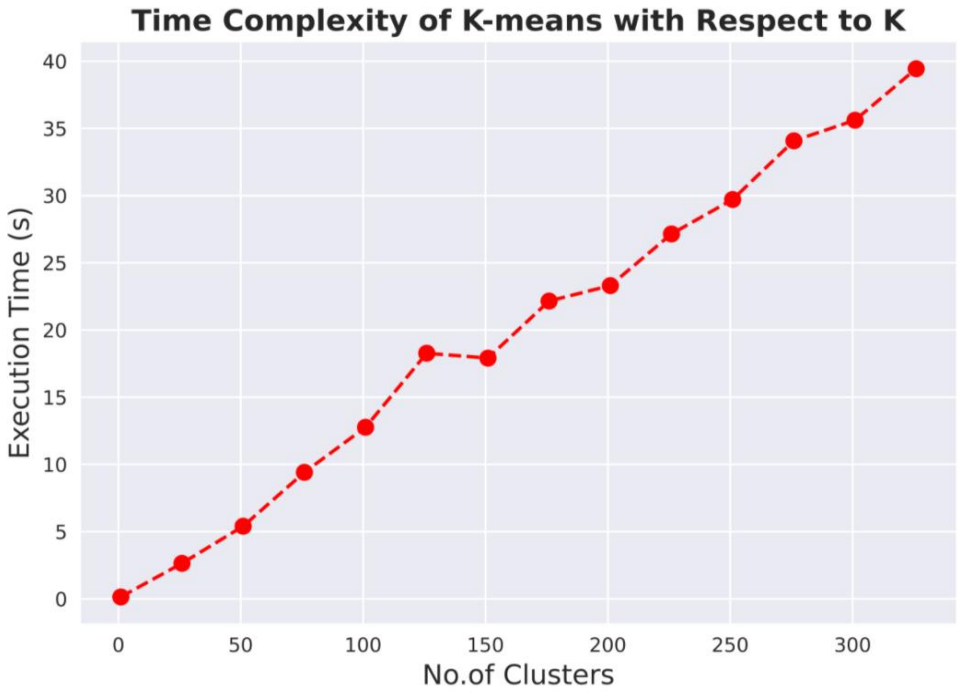
از نظر تئوری حداکثر تعداد خوشه‌ها برابر با تعداد داده‌هایی ورودی است که در این مثال این تعداد برابر با صدهزار می‌باشد، به عبارت دیگر حالتی که هر یک از نقاط داده یک خوشه مجزا باشند. اما این مقدار صرفا یک حد بالای تئوری برای این روش است و در اجرای این روش با تعداد زیاد خوشه‌ها مشکلات محاسباتی وجود دارد که افزایش بیش از حد تعداد خوشه‌ها باعث افزایش زمان اجرای کد خواهد شد. هرچند روش خوشه بندی k-means نسبت به سایر روش‌های دیگر از نظر محاسباتی روشی نسبتا ساده به شمار می‌رود، اما پیچیدگی محاسباتی[۱] روش خوشه بندی k-means به صورت $O(nkdi)$ مشخص می‌شود که n تعداد داده‌های ورودی الگوریتم، k تعداد خوشه‌ها، d تعداد مولفه‌های هر ورودی و i تعداد تکرارهایی است که روش در هر بار اجرا دارد. در این مثال خاص n=100000 ، d=2 ، i=10 (توجه شود که کتابخانه scikit-learn به صورت

[۱] Computational Complexity

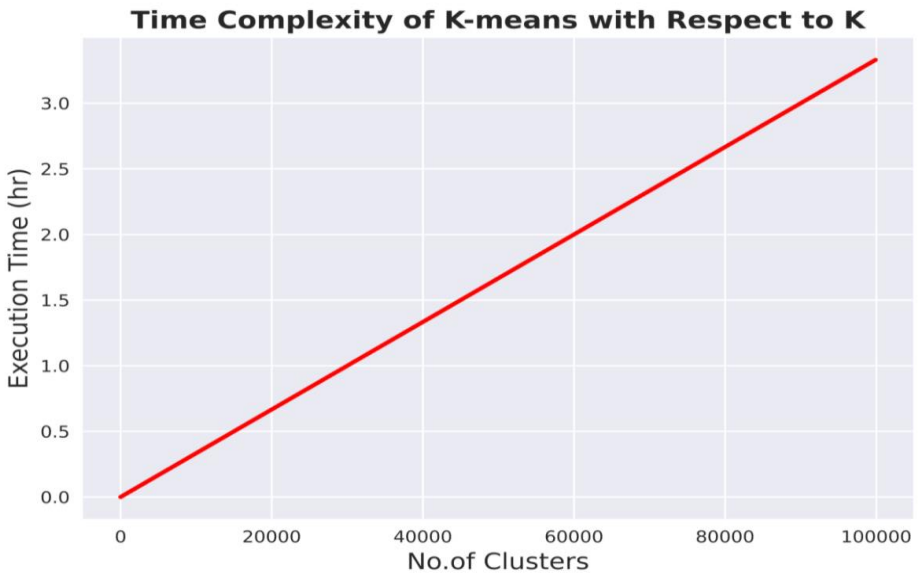
پیش فرض از ۱۰ تکرار برای اجرا این روش استفاده می‌کند) و با تغییر تعداد خوشه ها (k) می‌توان انتظار داشت زمان اجرای این روش به صورت خطی متناسب با افزایش تعداد خوشه‌ها افزایش یابد، این موضوع در شکل ۳.۱ بررسی شده است. با توجه به زمان اجرای این روش که در شکل ۳.۱.۱ نشان داده شده است یک مدل رگرسیون خطی برای تخیمن زمان اجرا بر حسب تعداد خوشه‌ها برازش داده شده است، که نتایج این مدل به صورت زیر می باشد :

$$Execution\ time\ (s) = 0.2348 + 0.119 \times K$$

برای محاسبه تقریبی زمان اجرا روش خوشه بندی k-means برای تعداد بیشتر خوشه ها از این مدل استفاده شده است که در شکل ۳.۱.۲ زمان تقریبی اجرای این روش برای تعداد خوشه های از ۰ تا ۱۰۰۰۰۰ نشان داده شده است، همانطور که در نمودار مشخص است زمان اجرا برای k=30000 تقریبا برابر با ۱ ساعت می‌باشد و زمان اجرا برای تعداد حداکثر خوشه‌ها به صورت تئوری (k=100000) تقریبا در حدود سه ساعت می‌باشد. در این قسمت فرض می شود تعداد حداکثر خوشه‌ها برابر با ۱۵۰ در نظر گرفته می‌شود. البته این عدد صرفا برای کوتاه بودن زمان محاسبات در بخش ۴ انتخاب شده است.



شکل۳.۱.۱ افزایش زمان اجرای روش خوشه بندی k-means با افزایش تعداد خوشه ها



شکل۳.۱.۲ زمان تقریبی اجرای روش خوشه بندی k-means بر اساس تعداد خوشه ها

۳.۱.۳. تعیین مقدار بهینه تعداد خوشه‌ها

با توجه به این که تعداد داده‌ها نسبتا زیاد می‌باشد نمی‌توان به صورت خیلی دقیق به ازای همه مقادیر k مقدار interia را محاسبه کرد و بر اساس روش elbow method مقدار بهینه k را مشخص کرد. برای محاسبه مقدار بهینه K مقدار inertia به ازای تعداد خوشه های از بازه ۲۵ تا ۴۰۰ با فاصله های ۲۵ تایی (k = 25,50,75,...400) محاسبه شده است و نتایج در شکل ۳.۱.۳ نشان داده شده است. در شکل ۳.۳ K=50 و K =75 به نظر انتخاب خوبی برای تعداد خوشه ها می‌باشند. برای انتخاب دقیق تر تعداد خوشه ها از بین چند کاندید موجود از معیاری به نام silhouette coefficient استفاده می شود که به صورت یک کسر تعریف می‌شود، که که صورت کسر معیاری از نوعی نزدیکی داده هایی در یک خوشه قرار دارند می باشد و مخرج آن نیز به نوعی بیان کننده فاصله داده‌های در خوشه‌های متفاوت است، مقدار این کسر عددی بین -۱ تا ۱ می باشد و هر چه این مقدار به ۱ نزدیک تر باشد نشان دهنده این است که داده های موجود هر خوشه به هم نزدیکی بیشتری دارند و همچنین فاصله بیشتری با داده های موجود در خوشه های دیگر دارند که در این صورت کیفیت خوشه بندی مناسب است. همانطور که در کد مشخص است این مقدار برای دو کاندید k=50 , k=75 تفاوت چندانی ندارد و به صورت زیر است:

$$\text{silhouette coefficient } for\ k = 50 : 0.5351539289170328$$

$$\text{silhouette coefficient } for\ k = 75 : 0.5354868434348565$$

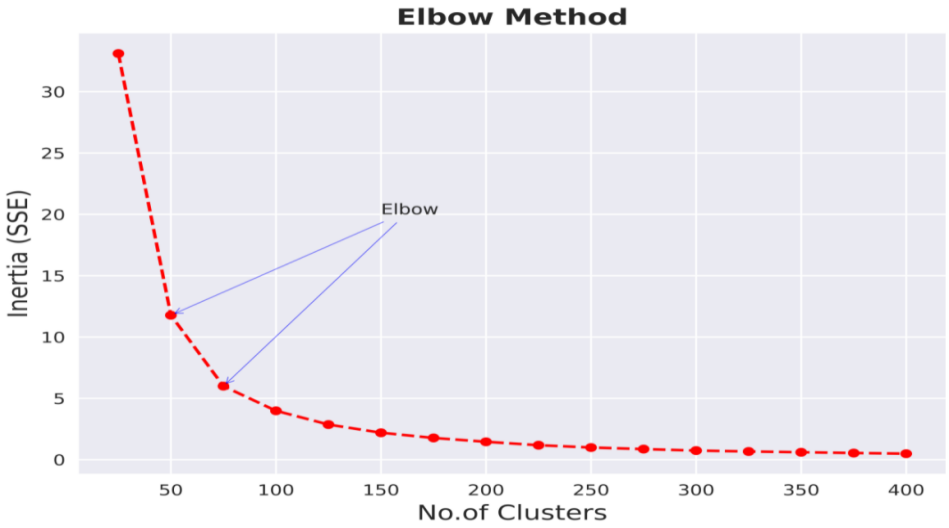
و مقدار این معیار برای $k=75$ به مقدار کمی بهتر است. نتایج خوشه بندی داده ها با $k=75$در شکل ۳.۱.۴ نشان داده شده‌است همانطور که مشخص است با توجه به تعداد زیاد خوشه‌ها به نظر می‌رسد برای افزایش کیفیت خوشه بندی تعدادی از خوشه‌ها کوچکتر می توانند با یک دیگر ترکیب شوند و تعداد خوشه ها کاهش یابد. در ادامه روش$elbow\ method$ برای تعداد خوشه های کمتر از ۵۰ با دقت بیشتر بررسی میشود، در شکل ۳.۱.۵ تغییرات $inertia$ در تعداد خوشه های زیر ۵۰ تا نیز بررسی می شود. با توجه به شکل ۳.۱.۵ به نظر می‌رسد $k=15$ مقدار مناسبی باشد برای بررسی بیشتر معیار silhouette coefficient برای k=15 محاسبه می شود و با نتایج k=75,50 مقایسه می شود تا بتوانیم مقدار k را به صورت مناسبی انتخاب کنیم.

silhouette coefficient $for\ k = 15$: 0.5209169088587667

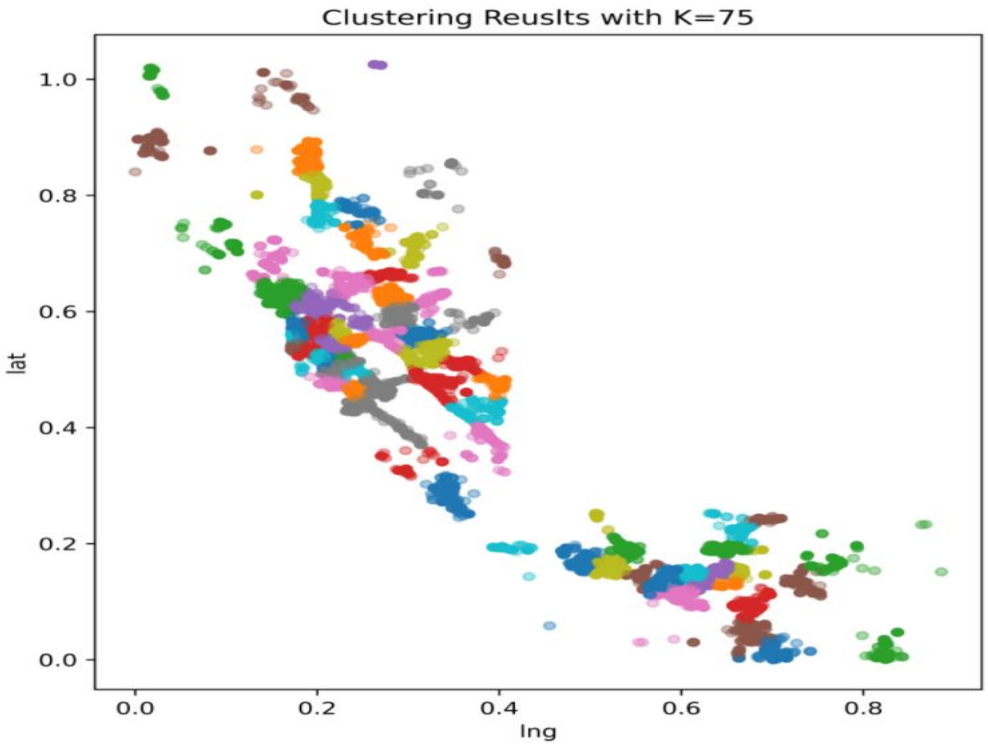
با توجه به این معیار $k=75$ مناسب تر از $k=15$ می باشد اما برای همانطور که در شکل ۳.۱.۴ نیز مشخص است برای بهبود کیفیت خوشه بندی تعداد خوشه ها می تواند کاهش یابد. در ادامه از اطلاعات دیگری برای این منظور استفاده شده است، ایالت کالیفرنیا 58 منطقه $county$ دارد به نظر می‌رسد شاید تعداد $county$ ایالت کالیفرنیا بتواند گزینه مناسبی برای انتخاب تعداد خوشه‌ها باشد با این استدلال که در هر یک از $county$ های این ایالت چگالی جمعیت بیشتر از سایر نقاط است. ابتدا مختصات مراکز این نواحی در قالب فایل GIS از اینترنت دانلود شده‌است و مرکز مختصات $county$ های ایالت کالیفرنیا در سیستم مختصات فرضی محاسبه شده است. در شکل ۳.۱.۶ مراکز این نواحی در میان داده‌ها نشان داده شده است و با توجه به این شکل به نظر می‌رسد $k=58$ عدد مناسبی برای تعداد خوشه ها باشد. برای مقایسه کیفیت خوشه ها در این حالت با حالتی که از روش $elbow\ method$ محاسبه شد $(k=75)$ مقدار silhouette coefficient برای این حالت محاسبه می‌شود. تنها فرق این روش با روش های قبلی این است که با تنظیم پارامترهای kmeans ، خودمان به الگوریتم پیشنهاد میکنیم که از مراکز county ها به عنوان نقاط مرکزی اولیه استفاده کند و بر این اساس خوشه ها را بهبود دهد و همچنین از روش accelerated k-means در مراحل روش خوشه بندی برای سرعت بیشتر استفاده میشود(توجه شود استفاده از روش accelerated k-means لزومی ندارد و صرفا برای کاهش زمان اجرای معیار silhouette coefficient از این روش استفاده شده است.) در این حالت معیار silhouette coefficient نسبت به حالت قبل بهبود داشته است و برابر با زیر می باشد.

silhouette coefficient $for\ k = 58\ and\ taking\ counties\ centroid\ as\ initial\ centroid$: 0.5511617237279225

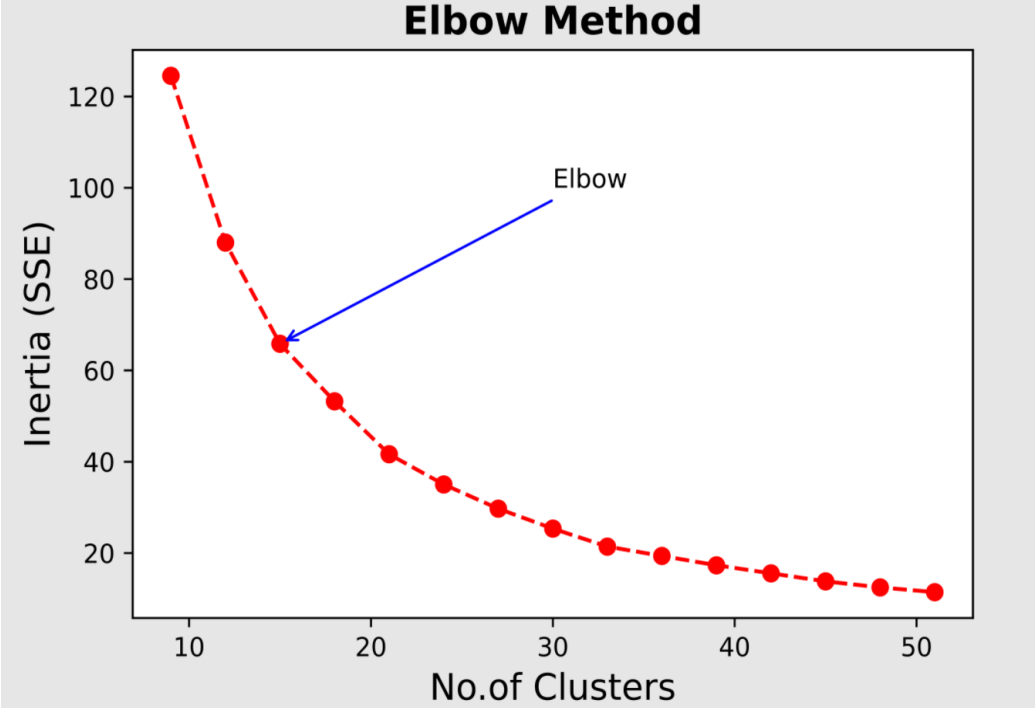
برای درک بهتر از کیفیت خوشه ها در این روش اخیر نتایج در شکل ۳.۱.۷ نشان داده شده است که داده‌های مربوط به هر خوشه با رنگ متفاوت نشان داده شده است.همانطور که در شکل مشخص است خوشه بندی به نحو قابل قبولی صورت گرفته است.



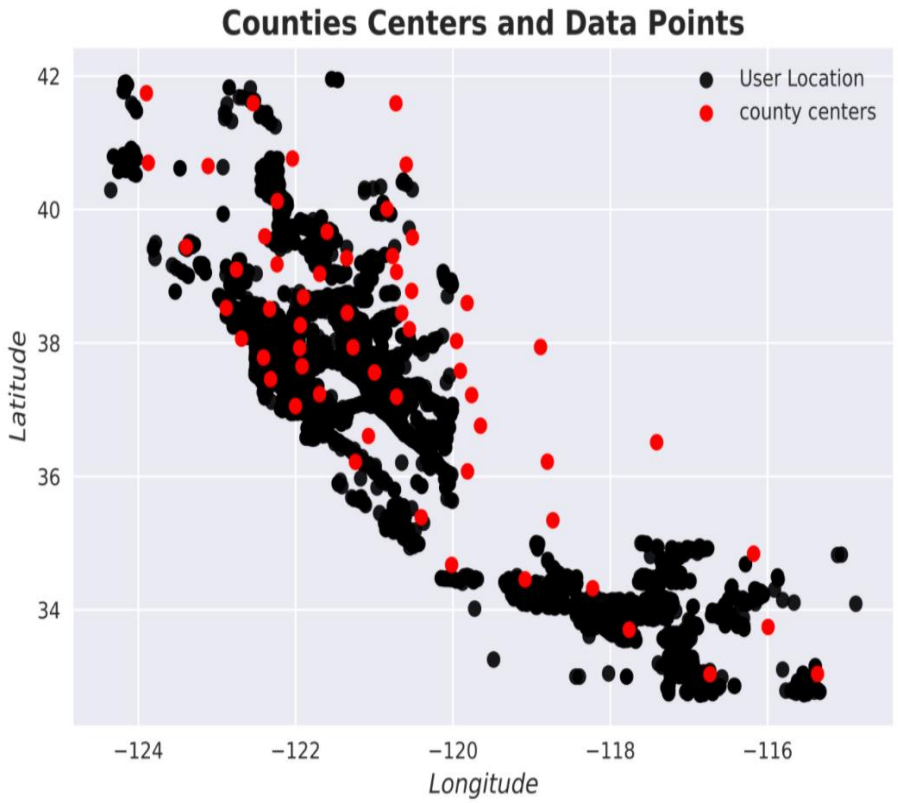
شکل ۳.۱.۳ روش elbow method



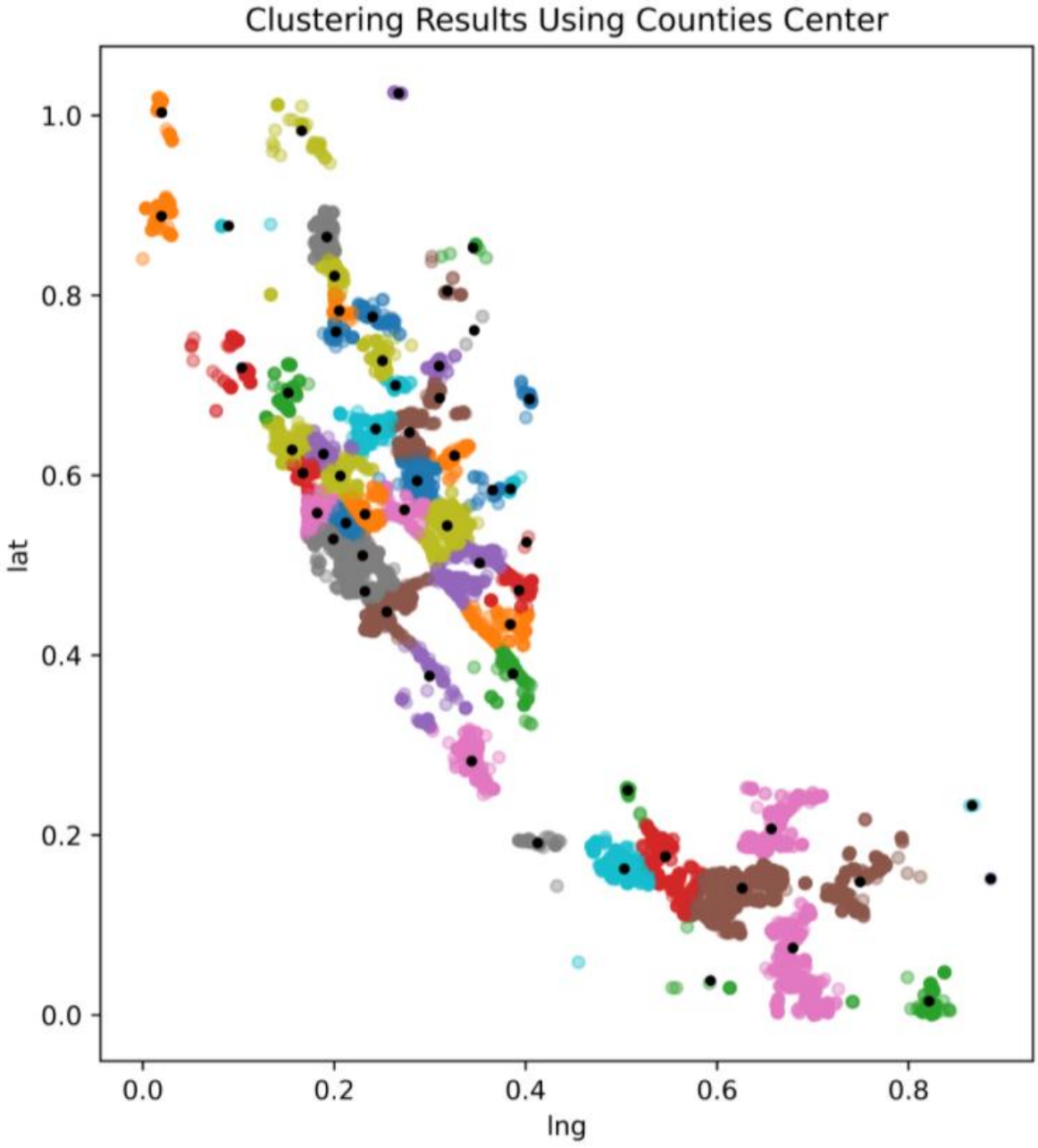
شکل ۳.۱.۴ نتایج خوشه بندی روش K-means برای k=75



شکل ۳.۱.۵. تغییرات مقدار *inertia* برای *k<52*



شکل ۳.۱.۶ مراکز county های ایالت کالیفرنیا



شکل ۳.۱.۷ نتایج مربوط به خوشه بندی داده ها با استفاده از مراکز county

۳.۲. الگوریتم Mini Batch K-Means

۳.۲.۱. اجرا الگوریتم Mini Batch K-Means

در این قسمت از الگوریتم خوشه بندی mini batch k-means استفاده می شود که در این الگوریتم در هر تکرار تعداد مشخصی از داده‌های ورودی در نظر گرفته می‌شوند و نتایج مدل در هر تکرار به روز رسانی می‌شود، این الگوریتم سرعت بسیار زیادی در مقایسه با روش معمول k-means دارد و البته ممکن است کیفیت خوشه‌بندی نهایی نسبت به روش معمول k-means کمتر شود که این موضوع تحت تاثیر عوامل مختلفی قرار دارد که می توان به ویژگی‌های داده و همچنین نحوه تنظیم پارامترها در روش mini batch k-means اشاره کرد. در قسمت اول کدی برای استفاده از این الگوریتم نوشته شده است که با فرض تعداد خوشه های ۱۰۰ تا و همچنین تعداد ۱۰۰ داده های ورودی در هر تکرار ین کد نوشته شده است و زمان اجرا آن نیز محاسبه شده است که برابر با ۰.۴۹۱۷۴ ثانیه می‌باشد که نسبت به به روش معمول k-means بسیار سریع تر عمل می‌کند.

۳.۲.۱محاسبه تعداد مناسب داده های ورودی در هر تکرار (Batch Size)

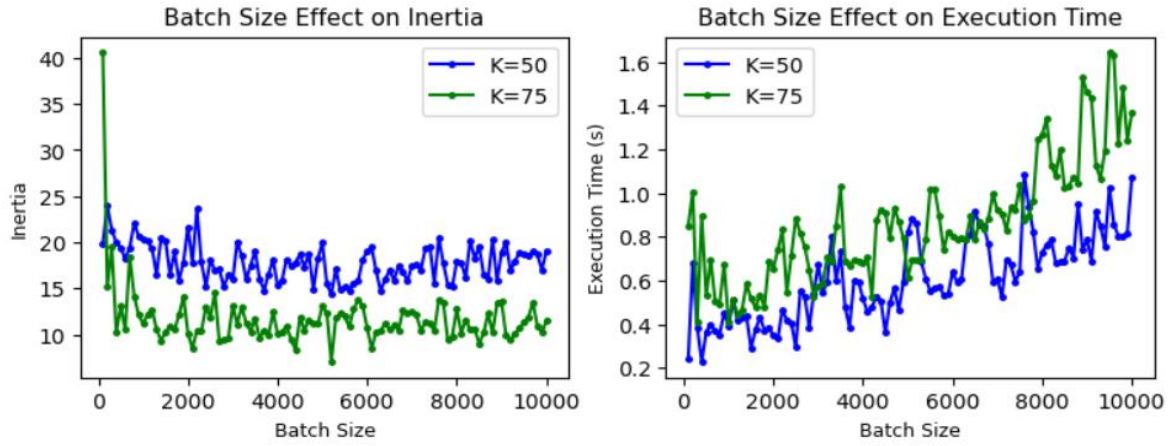
یکی از مهم ترین پارامتر های این الگوریتم batch size است که تاثیر بسیار زیادی در کیفیت خوشه بندی دارد. برای بررسی تاثیر این پارامتر از روش هایی معمول که کیفیت خوشه بندی را به کمک آن‌ها بررسی می‌شود می‌توان استفاده کرد و به طور کلی سه روش مختلف وجود دارد[۲]، در مثال حاضر با توجه به تعداد زیاد داده های نمی توان هر یک از این معیارهای ارزیابی را به ازای مقدار مختلف batch size محاسبه کرد چرا که زمان بسیار زیادی برای اجرا آن نیاز است (البته در این مثال با توجه به اینکه هیچ گونه اطلاعات حقیقی[۳] برای دسته بندی نقاط وجود ندارد تنها از معیار internal و relative می‌توان استفاده کرد که هزینه محاسباتی بالایی دارند.) در نهایت تنها معیار در دسترسی که می‌توان راجع به انتخاب batch size قضاوت کرد تنها مقدارinertia و زمان اجرا است. همچنین برای بهبود فرآیند تصمیم گیری در مورد batch size بر اساس دو معیار inertia و زمان اجرا نتایج در دو حالت برای مقادیر مختلف تعداد خوشه ها تعیین شده است. در شکل ۳.۲.۱ تاثیر batch size در مقدار inertia و زمان اجرا مورد بررسی قرار گرفته است. در این شکل مقدار inertia و زمان اجرا برای batch size مختلف در بازه‌ی(100,200,300,...,10000) محاسبه شده است همانطور که در شکل ۳.۲.۱ مشخص است، برای batch size در بازه بین ۴۰۰۰ تا ۶۰۰۰ مقدار inerti و زمان اجرا مناسب تر است و مقدارbatch size متناسب با حداقل مقادیر inertia در این شکل محاسبه

---

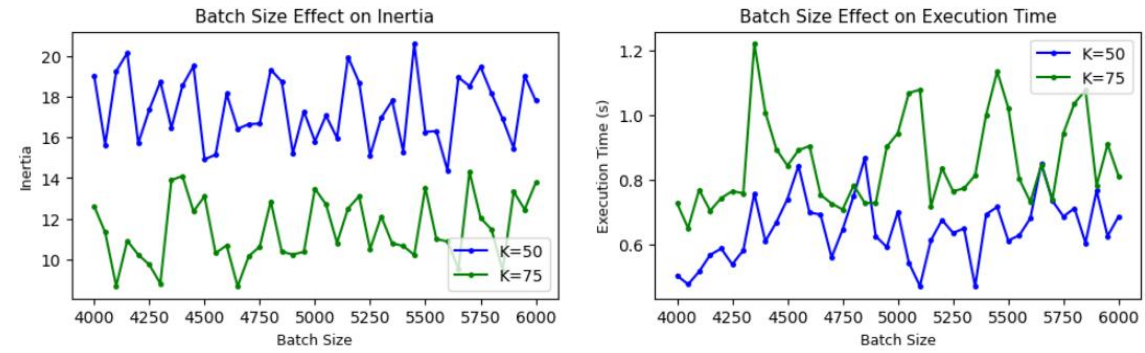
[۲] 1.External Measure, 2.Internal Measure 3.Relative Measure

[۳] Ground Truth

شده است. که در هر دو حالت انجام شده با تعداد مختلف خوشه ها مقدار batch size برابر با ۵۲۰۰ می‌باشد و همین عدد به عنوان batch size مناسب انتخاب می‌شود. البته توجه شود می‌توان برای بررسی دقیق تر بازه بین ۴۰۰۰ تا ۶۰۰۰ با دقت بیشتری بررسی کرد و نتایج در شکل ۳.۲.۲ نمایش داده شده است. با توجه به تصادفی بودن روش می توان انتظار داشت نتایج با هم متفاوت باشند که به همین صورت هم می‌باشد. با توجه به نتایج شکل ۳.۲.۲ batch size = 4600 مقدار مناسبی می‌باشد هر چند همانطور که گفته شد مقایسه بر اساس این معیار حالت مطلوبی نیست و بهتر است در صورت امکان مقایسه بر اساس معیارهایی که قبلا معرفی شد صورت گیرد. اما هر دو انتخاب batch size = 5200 , 4600 مناسب است.



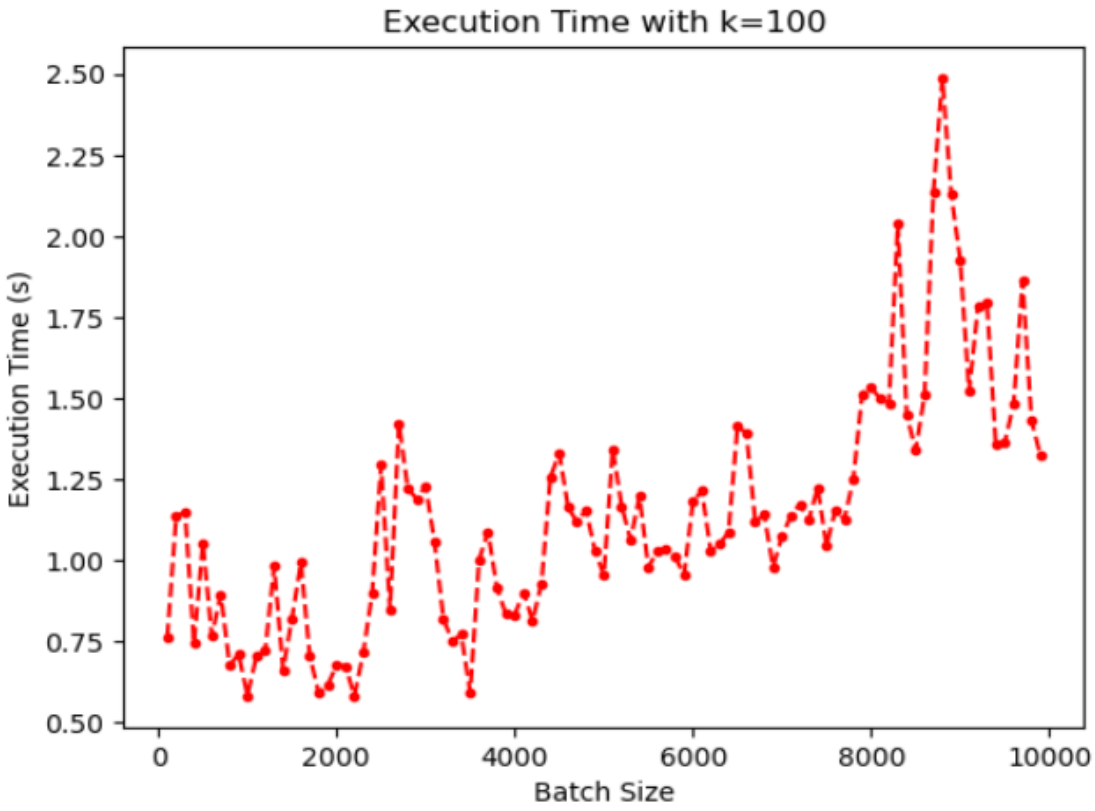
شکل ۳.۲.۱ تاثیر پارامتر batch size در مقدار inertia و زمان اجرای الگوریتم



شکل ۳.۲.۲ تاثیر پارامتر batch size در مقدار inertia و زمان اجرای الگوریتم برای اعداد بین ۴۰۰۰ تا ۶۰۰۰

۳.۲.۳محاسبه زمان اجرا در حالت k=100

در شکل ۳.۲.۳ زمان اجرای روش mini batch k-means به ازای مقادیر مختلف batch size رسم شده است



شکل ۳.۲.۳ زمان اجرای روش mini batch k-means به ازای مقادیر مختلف batch size

۳.۲.۴ تعیین حداکثر تعداد خوشه ها

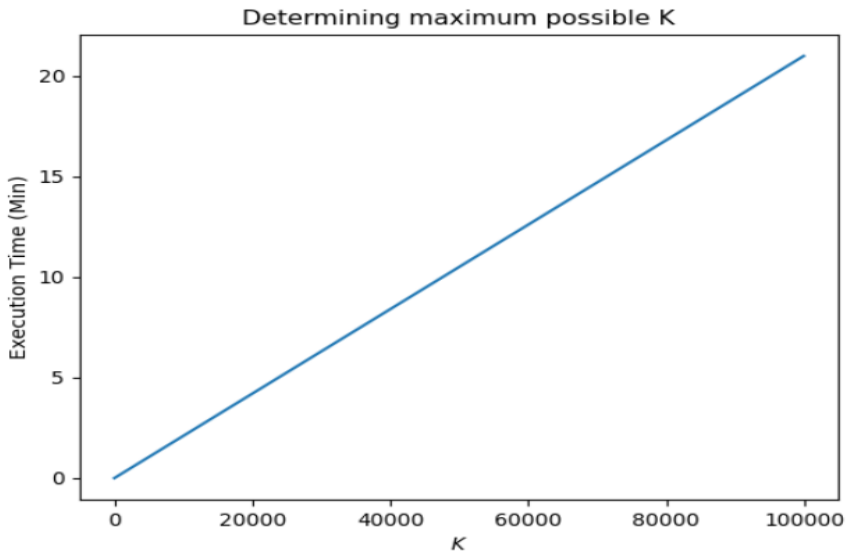
از نظر تئوری حداکثر تعداد خوشه‌ها در روش mini batch k-means برابر با تعداد داده‌ها می‌باشد، البته با توجه به این که در هر تکرار مراکز خوشه ها تعیین و به روز رسانی می‌شود توصیه می‌شود که تعداد خوشه ها حداکثر برابر با batch size باشد البته اگر بزرگتر هم باشد روش در اجرا با مشکلی مواجه نمی‌شود اما کیفیت خوشه بندی کاهش پیدا می کند. در این بخش می توان فرض کنیم که با توجه به قسمت ۳.۲.۲ batch size =5200 انتخاب مناسبی است و برای مشخص کردن حداکثر تعداد خوشه ها امکان پذیر(از

نظر هزینه محاسبات) می‌توانیم زمان اجرای این روش را به ازای تعداد خوشه های متفاوت بررسی کنیم و با مشاهده چگونگی افزایش زمان اجرا تصمیمی درباره حداکثر تعداد خوشه ها بگیریم. در شکل۳.۲.۴ زمان اجرای روش mini batch kmeans به ازای مقادیر متفاوت تعداد خوشه ها محاسبه شده است همانطور که مشخص است تقریبا زمان اجرا به صورت خطی با افزایش تعداد خوشه ها افزایش می‌یابد و با افزایش تعداد خوشه ها از حدی دیگر اجرای این روش از نظر هزینه محاسبات منطقی نخواهد بود. برای محاسبه تقریبی از زمان اجرا خط رگرسیونی برازش داده شده است که به صورت زیر می باشد :

$$Execution\ time\ (s) = -0.62929 + 0.012608 \times K$$

نتایج زمان اجرا بر اساس تخمین خط رگرسیون به صورت شکل ۳.۲.۵ می باشد که در زمان اجرای حداکثر تئوری تعداد خوشه ها (K=100000) تقریبا برابر با ۲۰ دقیقه می باشد و اگر زمان حداکثر قابل قبول برای اجرا را ۱۰ دقیقه در نظر بگیریم تقریبا تعداد حداکثر خوشه‌ها برابر با ۶۰۰۰۰ خواهد بود و این تعداد را می توانیم به عنوان یک حد بالای عملی در نظر بگیریم. اما حداکثر تعداد خوشه ها برای کوتاه بودن زمان اجرای بخش ۴، ۴۰۰ در نظر گرفته می شود.

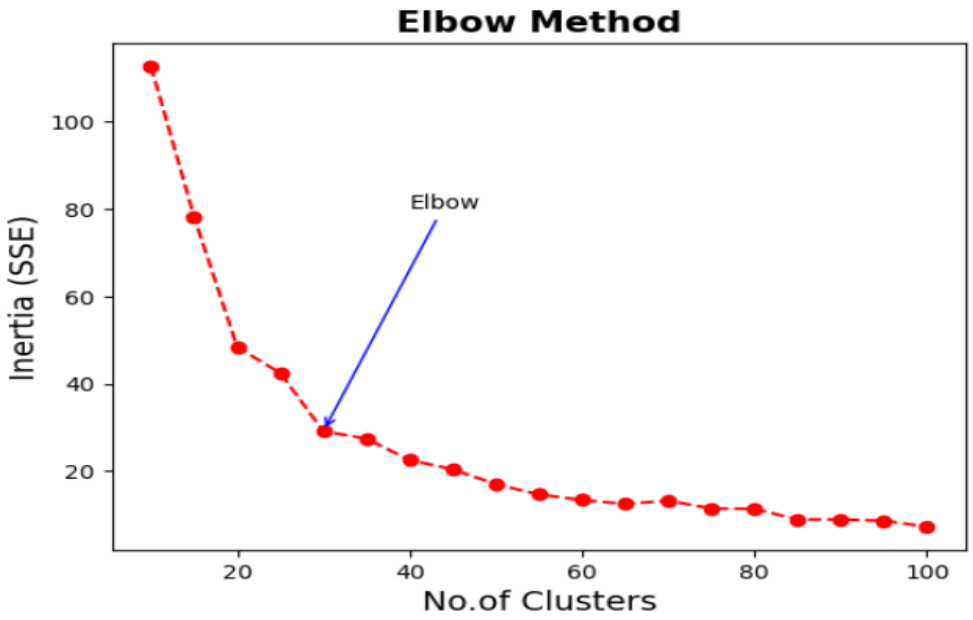
شکل ۳.۲.۴ زمان اجزای mini batch k-means به ازای مقادیر مختلف برای k



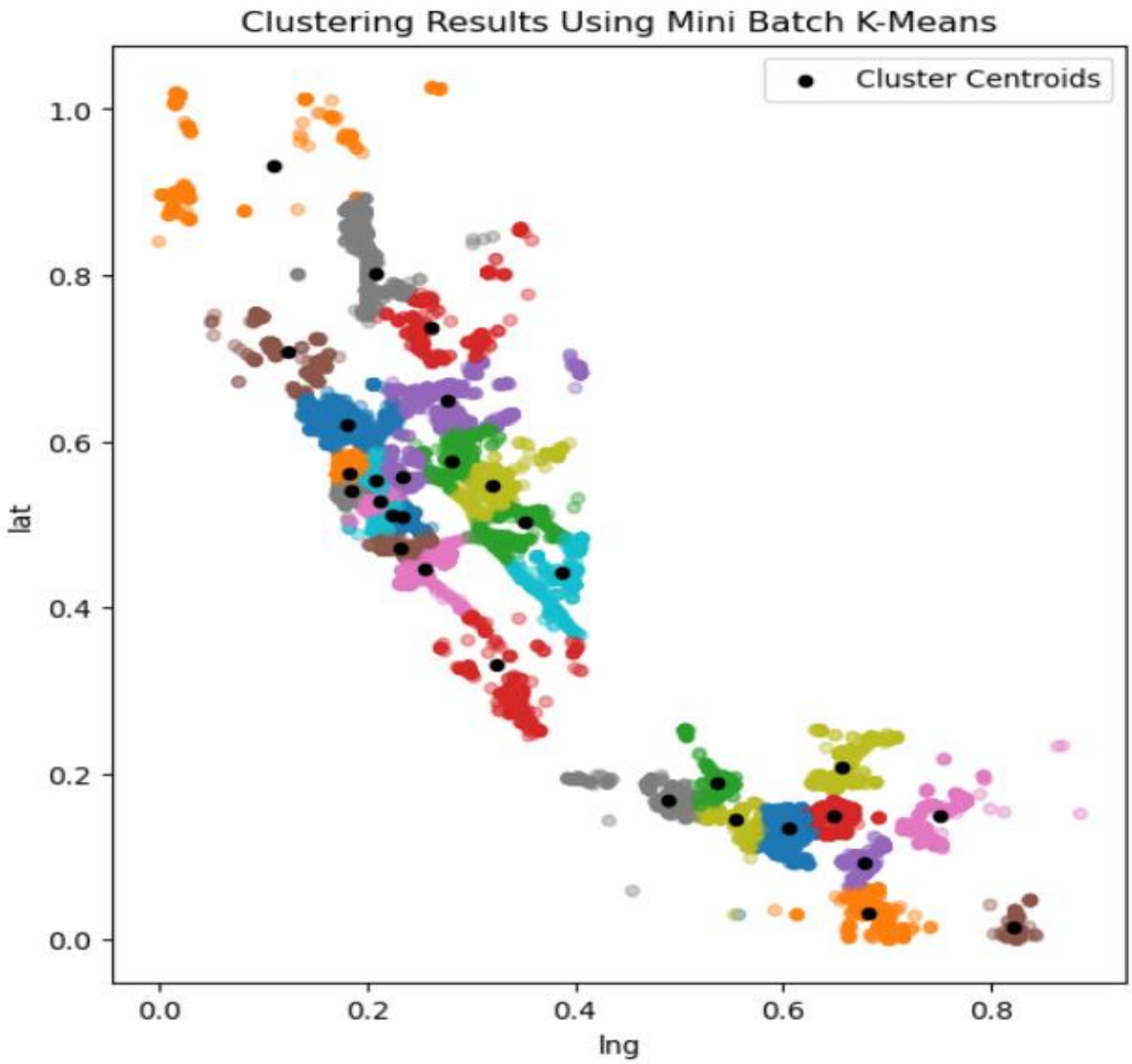
شکل ۳.۲.۵ زمان اجرای تقریبی به ازای مقادیر مختلف k

۳.۲.۴ تعیین مقدار بهینه k

برای محاسبه تعداد بهینه k از روش elbow method استفاده شده است با فرض batch size=5200 مقدار inertia برای مقادیر مختلف k محاسبه شده است همانطور که در شکل ۳.۲.۶ مشخص است k=30 با توجه به این شکل مقدار مناسبی است و در شکل ۳.۲.۷ نتایج خوشه بندی برای k=30 نشان داده شده است.



شکل ۳.۲.۶ مقدار inertia به ازای مقادیر مختلف k



شکل ۳.۲.۷ نتایج خوشه بندی بر اساس روش *mini batch kmeans* با *k=30*

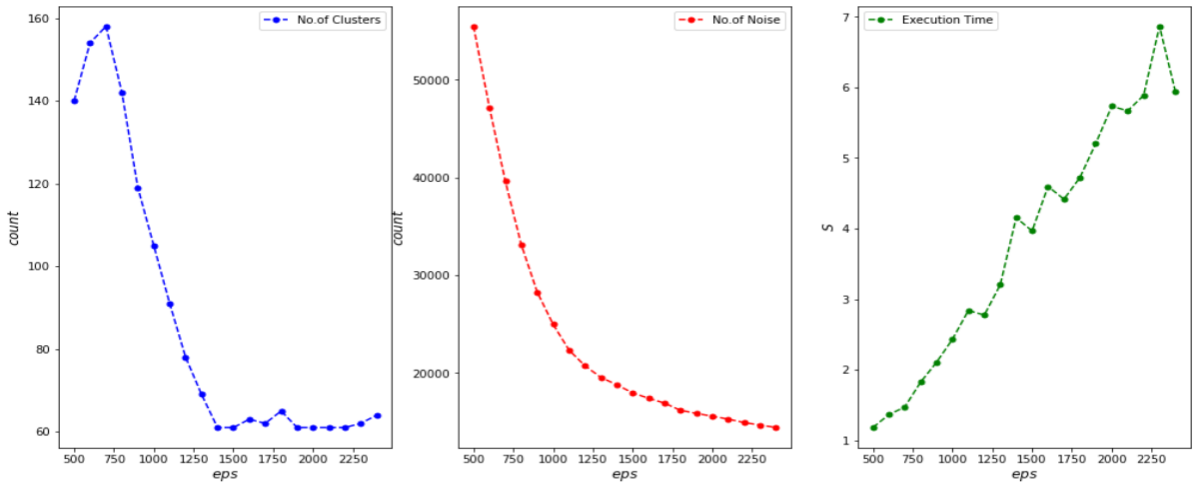
۳.۳. الگوریتم DBSCAN

۳.۳.۱. اجرا الگوریتم DBSCAN و قرار دادن min_samples=100

در این بخش ابتدا کدی برای اجرای این الگوریتم نوشته شده است و همچنین زمان اجرای آن نیز محاسبه شده است. در قسمت بعد پارامتر min_samples = 100 قرار داده شده است و این کد اجرا شده است.

۳.۳.۲. بررسی تاثیر پارامتر eps در الگوریتم DBSCAN

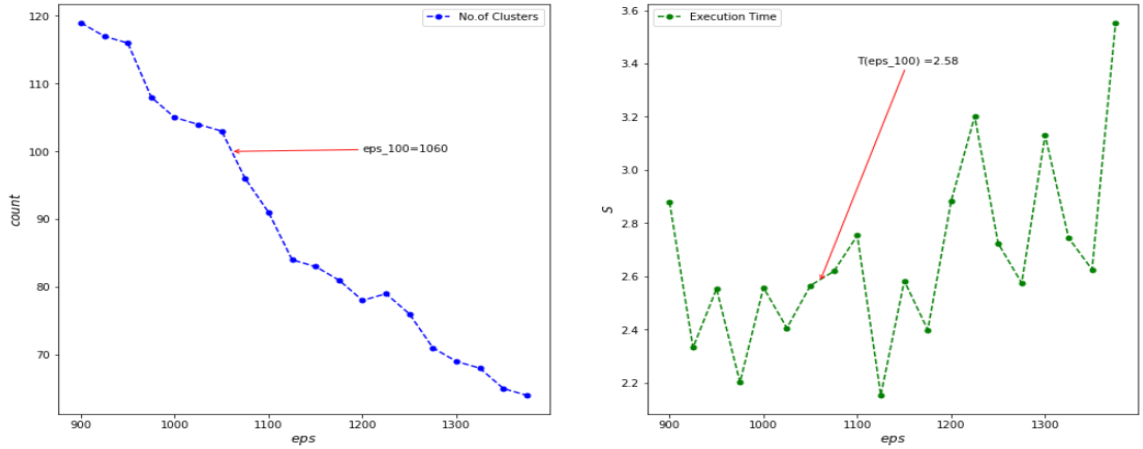
الگوریتم DBSCAN بر خلاف الگوریتم های k-means و mini batch k-means می‌تواند خوشه های با شکل های غیر محدب را نیز شناسایی کند. اما به صورت کلی این الگوریتم تنها برای داده‌هایی عملکرد مناسبی دارد که چگالی نقاط در مراکز پر تراکم تقریبا یکسان باشد چرا که این الگوریتم بر اساس پارامترهای اصلی خود یعنیminPts, eps تنها می‌تواند خوشه بندی را بر اساس فرض یک حالت از چگونگی تراکم انجام دهد و اگر داده‌ها با چگالی های متفاوت در نقاط مختلفی متمرکز شده‌باشند کیفیت خوشه بندی الگوریتم DBSCAN کاهش می‌یابد. یکی از مشکلات اصلی این روش حساسیت شدید نتایج به پارامترها این روش است و این مسئله باعث شده است تعیین پارامترها در این روش یکی از چالش‌های جدی باشد.(البته با کمک روش های دیگر می‌توان بر اساس ویژگی های داده تخمین خوبی برای پارامترهای این روش محاسبه کرد.) در این قسمت تاثیر پارامتر eps با مشاهده نتایج تعداد خوشه‌ها، تعداد داده‌های پرت(noise) و زمان اجرا به ازای مقادیر مختلف این پارامتر در شکل ۳.۳.۱ نشان داده شده‌است. به طور کلی اگر مقدار eps کوچک در نظر گرفته‌شود داده‌هایی که در نقاط با تراکم کمتر و با فاصله زیاد از بقیه نقاط و جود دارند به عنوان noise در نظر گرفته می‌شوند، همانطور که در شکل ۳.۳.۱ نشان داده شده است با افزایش مقدار eps تعداد noise ها کاهش می یابد. اما اگر مقدار eps بزرگ در نظر گرفته شود خوشه های تشکیل شده در این حالت از چندین خوشه کوچکتر تشکیل شده‌اند که ممکن است هر کدام از این زیر خوشه ها چگالی متفاوتی داشته باشند اما به دلیل افزایش شعاع eps همگی در یک خوشه قرار گرفته‌اند، همانطور که در شکل ۳.۳.۱ نشان داده شده است با افزایش مقدار eps تعداد خوشه ها کاهش می یابد. همانطور که توضیح داده شد هر چه eps بزرگتر در نظر گرفته شود الگوریتم نقاط بیشتر را برای خوشه بندی در نظر می‌گیرد به عبارت دیگر تعداد density connected points افزایش می‌یابد و حجم محاسبات افزایش می‌یابد و به همین دلیل با افزایش eps زمان اجرا الگوریتم افزایش می‌یابد.



شکل ۳.۳.۱ بررسی تاثیر پارامتر eps در الگوریتم DBSCAN

۳.۳.۳. محاسبه eps_100

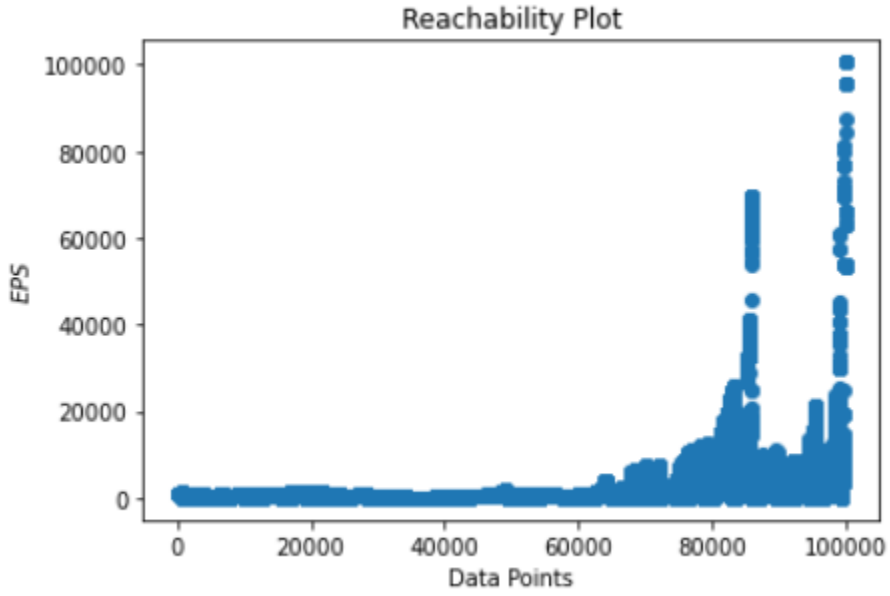
همانطور که در شکل ۳.۳.۱ مشخص است به ازای eps نزدیک به ۱۰۰۰ تعداد خوشه ها ۱۰۰ عدد می‌باشد، بررسی دقیق تر eps متناظر با ۱۰۰ خوشه در شکل ۳.۳.۲ نشان داده شده است. همانطور که در این نمودار نشان داده شده است eps_100 تقریبا برابر با ۱۰۶۰ متر و همچنین زمان اجرای متناظر با آن تقریبا ۲.۵۸ ثانیه می‌باشد.



شکل ۳.۳.۲ محاسبه eps_100 و زمان اجرای متناظر با آن

۳.۳.۴. انتخاب پارامترهای مناسب در DBSCAN

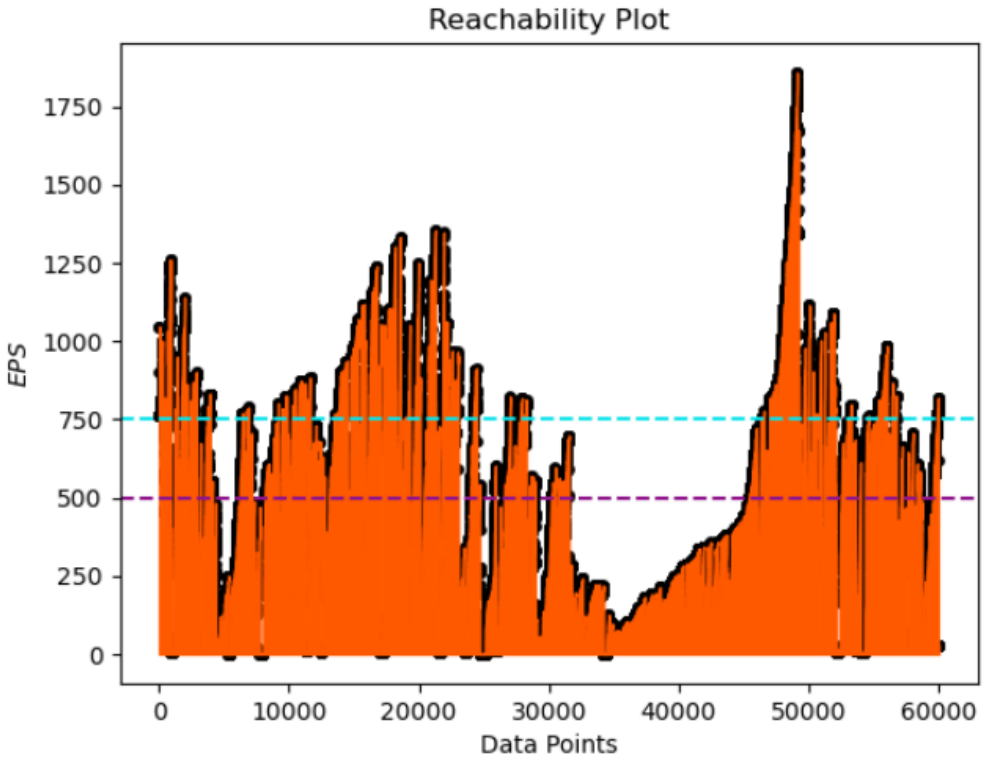
در این بخش سعی می‌شود بهترین نتایج خوشه بندی با توجه به روش DBSCAN مشخص شود. در ابتدا لازم است که برای دو پارامتر minPts, eps تصمیم گرفته شود، پارامتر minPts مطابق با قسمت‌های قبل ۱۰۰ در نظر گرفته می‌شود و مقدار پارامتر eps برای رسیدن به نتیجه بهتر تعیین می‌شود. برای تعیین بهینه پارامتر eps از الگوریتم OPTICS[4] استفاده می‌شود که در واقع این الگوریتم یک ساختار رده‌بندی شده از داده در اختیار ما قرار می‌دهد تا بتوان با توجه به نتایج آن انتخاب بهتری در تعیین پارامترهای DBSCAN انجام دهیم. در نتیجه اجرای این الگوریتم نموداری به نام reachability distance plot به دست خواهد آمد که در واقع تمام داده‌ها با ترتیب مشخصی در محور افقی قرار گرفته‌اند و در محور عمودی eps reachability distance متناظر با هر داده تصویر شده است. اگر توزیع چگالی نقاط داده به صورت یکسان باشد دره‌ها این نمودار در عرض تقریبا یکسانی رخ می‌دهد و می توان نتیجه گرفت نتایج الگوریتمDBSCAN با eps برابر با عرض شکل گیری دره‌ها بسیار مناسب خواهد بود. در شکل ۳.۳.۳ reachability distance plot نشان داده شده‌است.



شکل۳.۳.۳

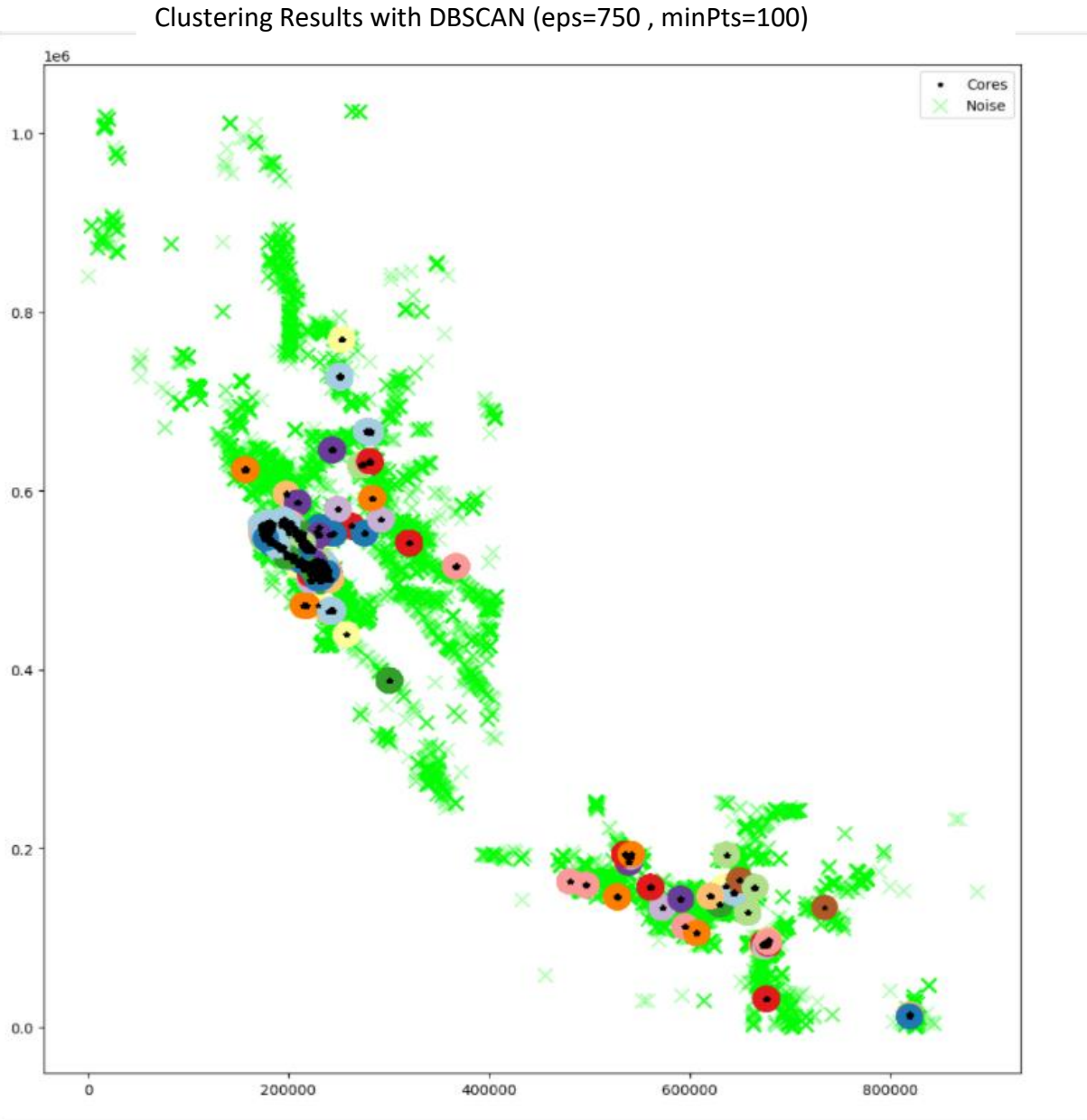
[4] Ordering Points to Identify Clustering Structure

همانطور که در نمودار مشخص است چگالی داده‌ها به صورت یکسان نمی‌باشد و در داده‌های انتهایی نمودار مشخص است که تراکم بسیار کم می شود و دره های شکل گرفته با eps=2000 می‌باشد که با توجه به بزرگی آن از نظر محاسباتی زمان اجرای الگوریتم بسیار طولانی می شود. برای انتخاب eps بهتر به نظر می رسد داده های قبل از ترتیب ۶۰۰۰۰ در شکل ۳.۳.۳ رفتار تقریبا مشابهی داشته باشند. به منظور بررسی دقیق‌تر داده های کمتر قبل از ترتیب ۶۰۰۰۰ در شکل ۳.۳.۴ نشان داده شده اند. با توجه به این شکل به نظر می‌رسد داده‌ها رفتار یکسانی ندارند و به طور کلی استفاده از روش DBSCAN با یک مقدار ثابت eps چندان روش مناسبی برای خوشه بندی نباشد و اما با توجه به شکل ۳.۳.۴ انتخاب eps=500,750 که در نمودار هم مشخص شده است به نظر مناسب می باشد چرا که این مقادیر اکثر دره های اصلی (نقاط با تراکم بالا) در داده را دربر میگیرد و به نظر می رسد عملکرد مناسبی در خوشه بندی مناطق با تراکم های بالا خواهد داشت. البته با انتخاب این مقادیر دره های عمیق تر همگی در یک خوشه قرار خواهند گرفت. در شکل ۳.۳.۵ نتایج خوشه بندی با مقدار eps=750 نشان داده شده است.



شکل۳.۳.۴

در شکل ۳.۳.۵ نتایج خوشه بندی با eps=750 نشان داده شده است که تعداد خوشه ها در این حالت برابر با ۱۴۴ می باشد و نقاط مشکی رنگ نشان دهنده نقاط هسته است و داده های مربوط به هر خوشه با رنگ‌های متفاوتی نشان داده شده است و نقاط سبز رنگ بیان گر noise است. همانطور که در شکل مشخص است با توجه به مقدار eps که نسبتا کم می باشد تعداد noise ها بسیار زیاد است و عموما مناطق شهری پر جمعیت کالیفرنیا به عنوان خوشه های با چگالی بالا در نظر گرفته شده اند.



شکل ۳.۳.۶ نتایج خوشه بندی با روش DBSCAN

۳.۴. جمع بندی الگوریتم‌های مختلف

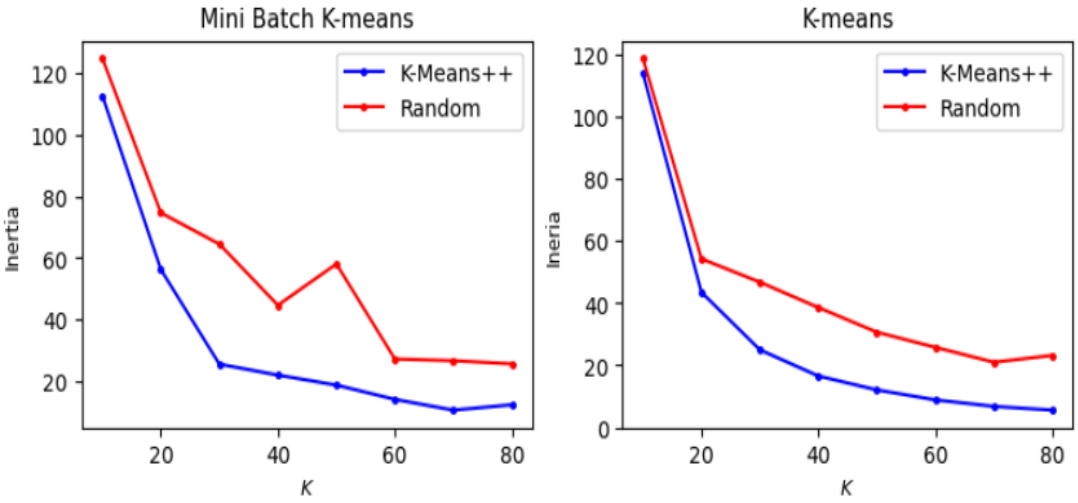
توضیحات مربوط به هر یک از الگوریتم ها و نحوه انتخاب پارامتر های مختلف در هر یک از آن‌ها به صورت کامل در بخش‌ها قبل ذکر شد. در این قسمت به بررسی تاثیر سایر پارامترهای الگوریتم های مختلف پرداخته می‌شود و همچنین نتایج سه روش مختلف از دیدگاه متفاوتی مورد بررسی قرار می‌گیرد.

۳.۴.۱. بررسی تاثیر پارامترهای مختلف در الگوریتم Mini Batch K-Means,K-Means

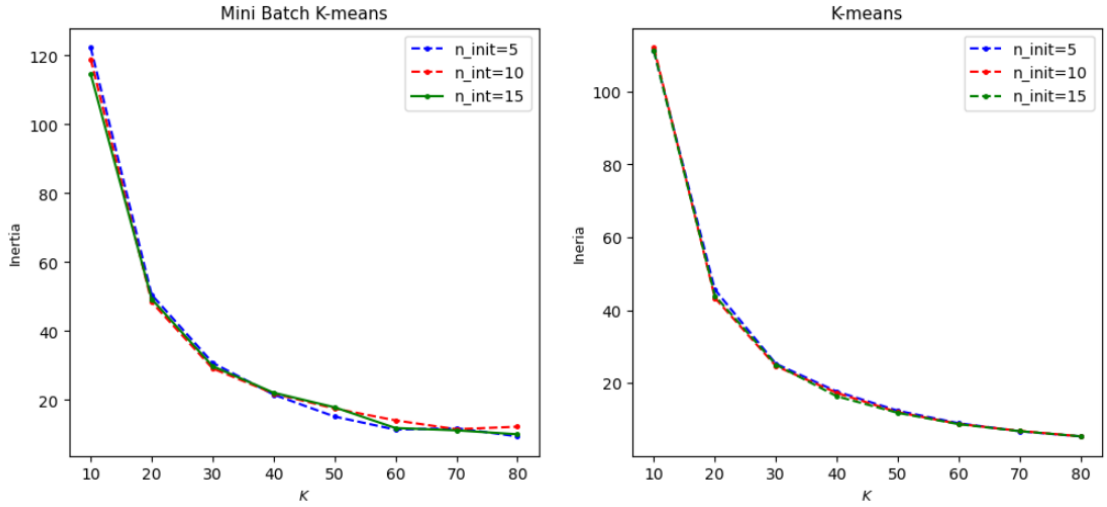
در این بخش پارامترهای init, n_init, max_iter بررسی می‌شوند.

- init : پارامتر init مربوط به نحوه انتخاب اولیه مراکز خوشه‌هاست که به دو روش random, kmeans++ انجام می‌شود. در روش random انتخاب مراکز به صورت تصادفی انجام می‌شود که بر اساس الگوریتم اصلی k-means است اما در حالت kmeans++ انتخاب مراکز به صورت هوشمندانه ای انجام می‌شود که مراکز خوشه ها در یک محدوده جمع نشوند و روش به این صورت است که ابتدا یک مرکز به صورت تصادفی انتخاب می‌شود و در ادامه فاصله تمام نقاط با آن مرکز محاسبه می‌شود و احتمالی براساس فاصله هر نقطه از مرکز فعلی برای انتخاب هر نقطه به عنوان یک مرکز دیگر محاسبه می‌شود و نقطه ای در مرحله بعد به عنوان مرکز انتخاب می‌شود که دورترین نقطه به مرکز انتخاب شده فعلی باشد(احتمال آن بیشتر باشد) و به همین صورت مراکز اولیه انتخاب می‌شوند. محققان این روش اثبات کردند هزینه محاسباتی اولیه این روش نسبت به الگوریتم اصلی در تعداد تکرار های کمتر این روش نسبت به روش اصلی قابل جبران است و در کل این روش نسبت به الگوریتم اصلی سریع تر است. کتابخانه scikit-learn به طور پیش فرض از روش kmeans++ استفاده شده است. در شکل ۳.۴.۱ تاثیر این پارامتر در مقدار inertia نشان داده شده است و همانطور که مشخص است نتایج kmeans++ در هر دو روش بهتر است.
- n_init : در الگوریتم های Mini Batch K-Means,K-Means هر بار که الگوریتم اجرا میشود به تعداد n_init بار این روش با مراکز اولیه مختلف اجرا می شود و در نهایت بهترین نتیجه از بین چند بار اجرا بر حسب کمتر بودن مقدارinertia گزارش می‌شود طبیعی است که هر چه مقدار آن بیشتر باشد نتایج خوشه بندی بهتر است. در کتابخانه scikit-learn مقدار این پارامتر به صورت پیش فرض ۱۰ در نظر گرفته شده است. در شکل ۳.۴.۲ تاثیر این پارامتر در دو دوش Mini Batch K-Means,K-Means نشان داده شده است. همانطور که در شکل مشخص است نتایج چندان با هم تفاوتی ندارند.
- Max_iter : این پارامتر مشخص کننده تعداد تکرار هایی است که الگوریتم های Mini Batch K-Means,K-Means برای رسیدن به جواب نهایی طی می کنند. در کتابخانه scikit-learn مقدار این

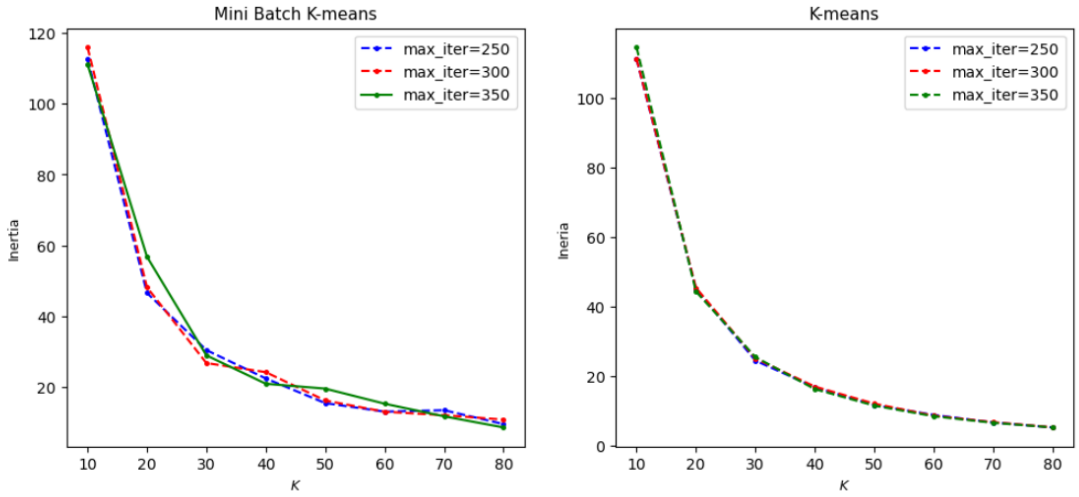
پارامتر به صورت پیش فرض ۳۰۰ در نظر گرفته شده است. در شکل۳.۴.۳ تاثیر این پارامتر در دو دوش Mini Batch K-Means,K-Means نشان داده شده است.



شکل ۳.۴.۱ تاثیر پارامتر init



شکل ۳.۴.۱ تاثیر پارامتر n_ init



شکل ۳.۴.۳ تاثیر پارامتر *max_iter*

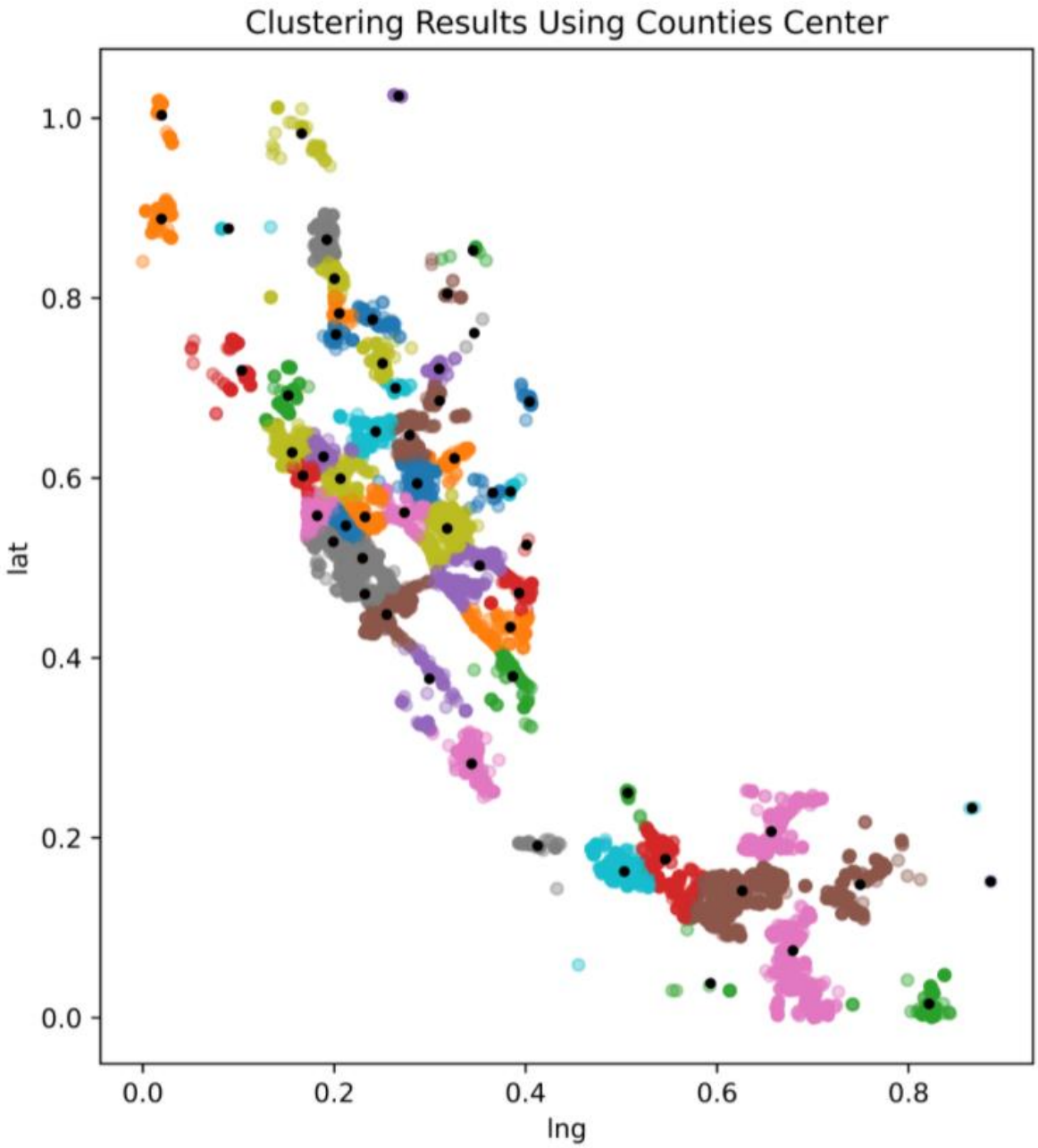
۳.۴.۱. مقایسه نتایج الگوریتم‌های مختلف

DBSCAN	Mini Batch K-Means	K-Means	
minPTS = 100 eps=750	K=30, batch_size =5200	K=58 Using counties center as intialization	Parameter of best result
-	23.42	5.126	Inertia
-	-	0.55116172373	silhouette coefficient
2.71	0.49	11.1	Execution Time
-	$-0.62929 + 0.012608 \times K$	$0.2348 + 0.119 \times K$	Estimated Time
Worst case scenario O(n^2)	O(n)	O(nkdi)	Computational Complexity

۳.۴.۲. نتایج نهایی مربوط به روش های مختلف

نتایج مربوط به روش k-means

K = 58 و از نقاط مراکز county های کالیفرنیا به عنوان نقاط اولیه مراکز استفاده شده است

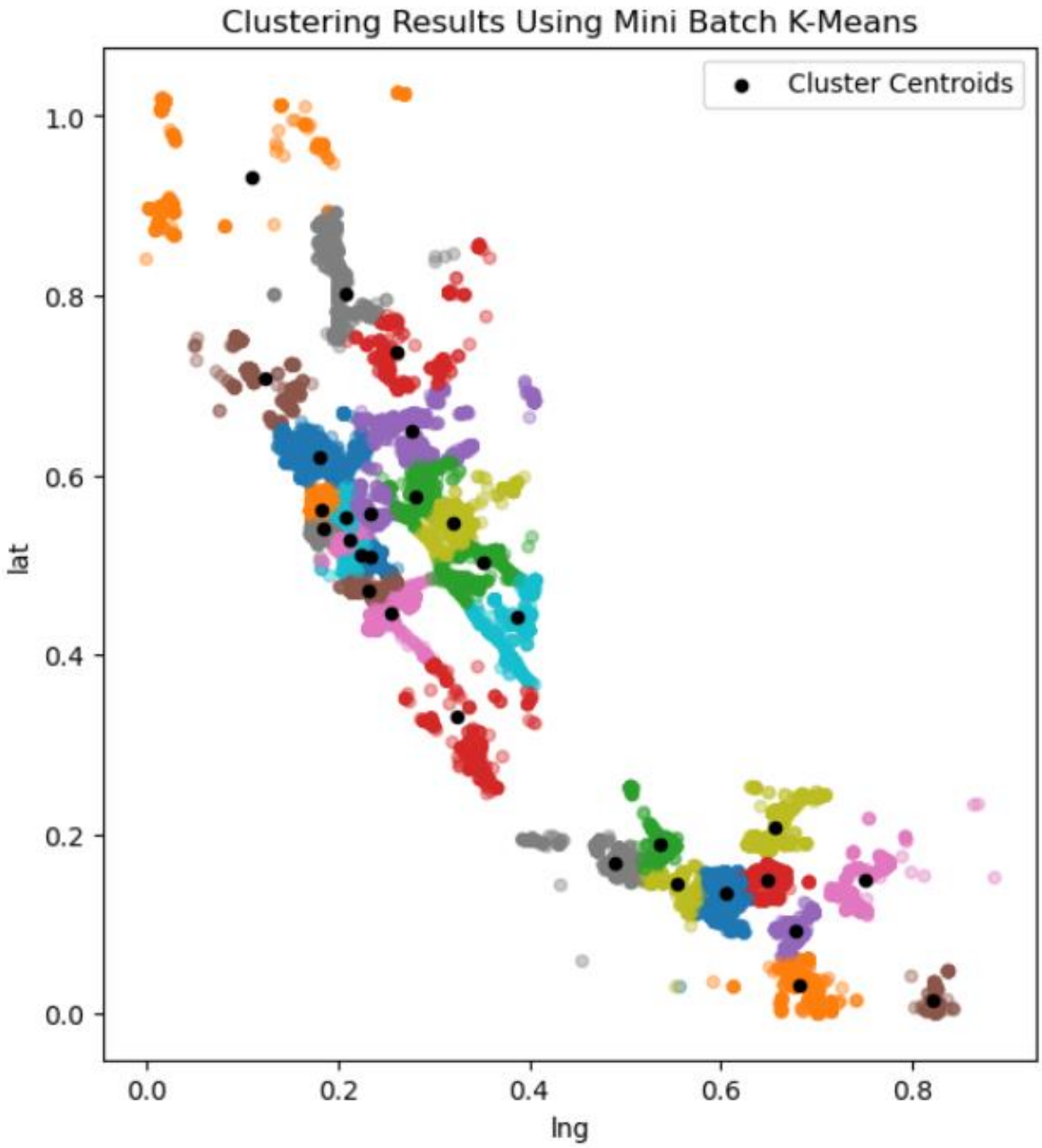


نتایج مربوط به روش k-means

K = 58 و از نقاط مراکز county های کالیفرنیا به عنوان نقاط اولیه مراکز استفاده شده است

نتایج مربوط به Mini Batch K-means

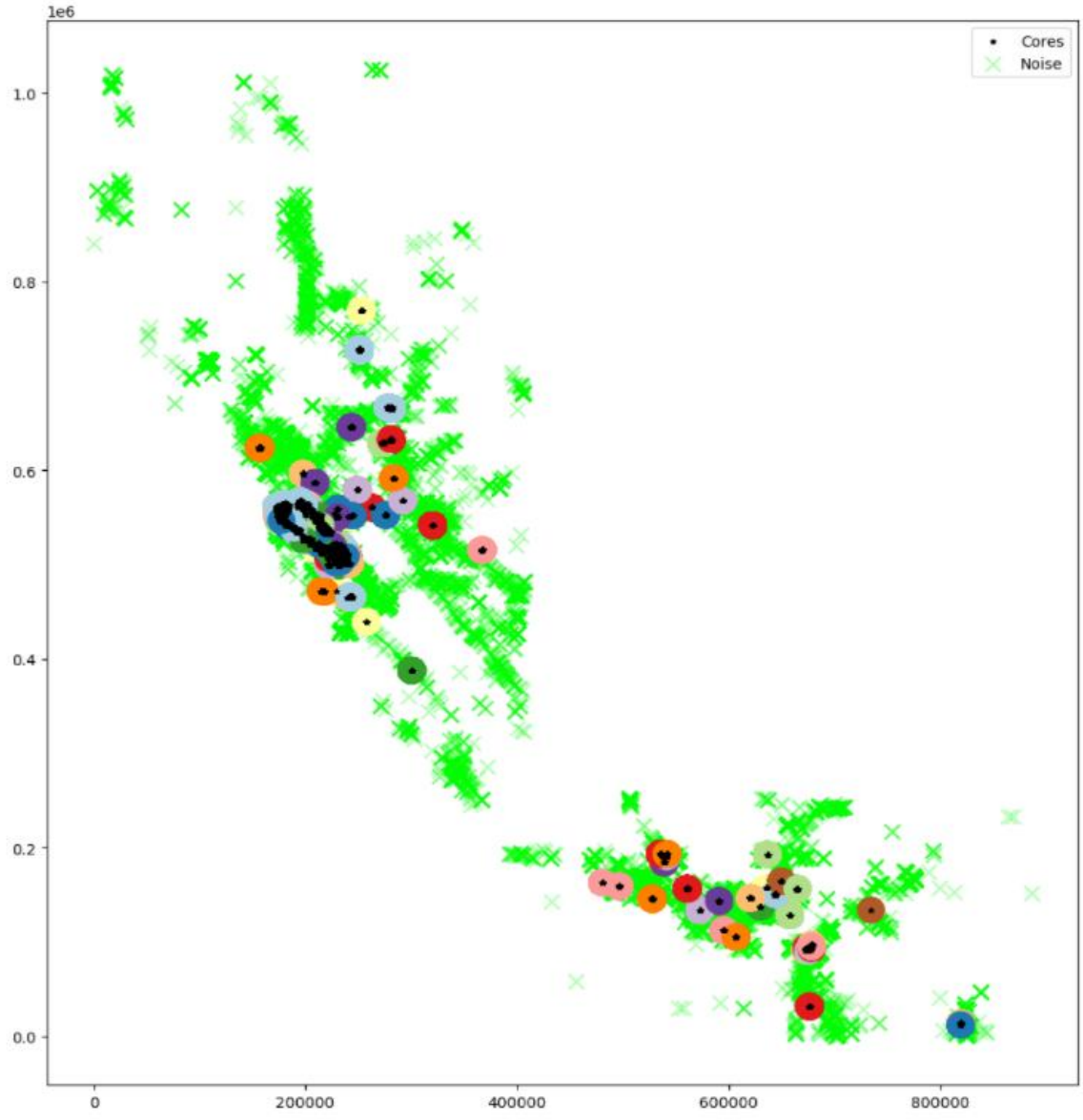
batch size = 5200, K = 30



نتایج مربوط به DBSCAN

MinPts = 100, eps = 750, K = 144

Clustering Using DBSCAN



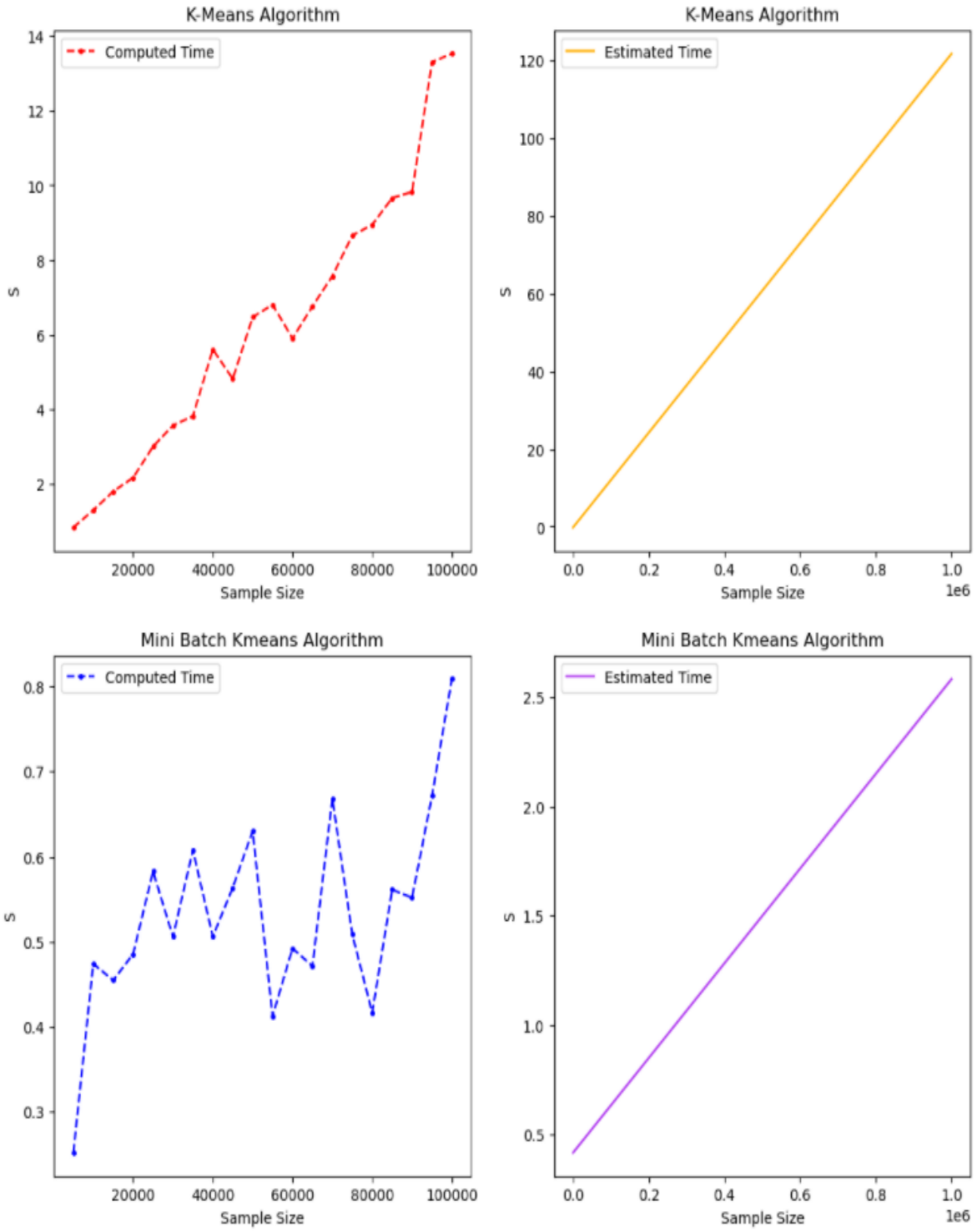
۴. مقیاس پذیری روش های خوشه بندی

۴.۱. الگوریتم K-Means و Mini Batch k-means

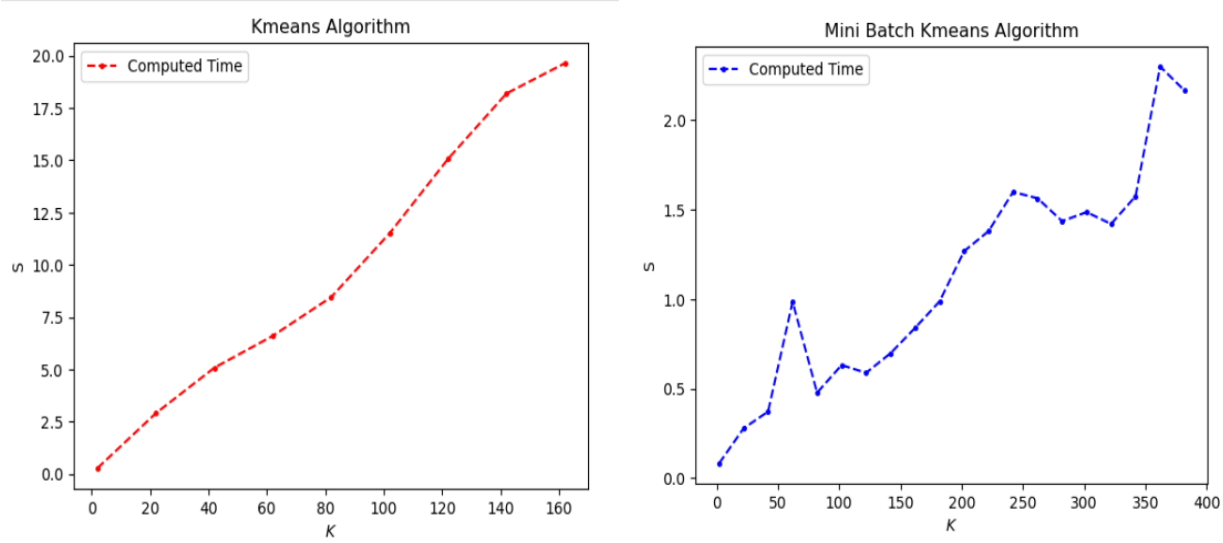
در شکل ۴.۱ زمان اجرای دو الگوریتم k-means , mini batch kmeans برای سایز مختلف داده های ورودی نشان داده شده است، همانطور که در شکل مشخص است تقریبا زمان اجرا به صورت خطی با افزایش تعداد داده های ورودی افزایش می یابد که این موضوع را می توان با توجه به پیچیدگی زمانی این الگوریتم ها نیز اثبات کرد. برای محاسبه زمان تقریبی این دو الگوریتم برای تعداد یک میلیون داده خط رگرسیونی بر زمان اجرا برازش داده شده‌است که نتایج آن در شکل ۴.۱ نشان داده شده است. به طور تقریبی زمان اجرای الگوریتم kmeans برای ۱ میلیون داده برابر با ۲ دقیقه است و همچنین زمان تقریبی الگوریتم mini batch kmeans برای این تعداد داده ورودی برابر با ۲.۵ ثانیه خواهد بود.

در شکل ۴.۲ زمان اجرای دو الگوریتم k-means , mini batch kmeans برای مقادیر حداکثر تعداد خوشه ها در بخش های قبل تعیین شد نشان داده شده است، حداکثر تعداد خوشه‌ها به ترتیب ۱۶۰ و ۴۰۰ در نظر گرفته شده است. البته همانطور که قبلا توضیح داده شد این حد بالا صرفا به گونه ای تعیین شده‌است که زمان محاسبات خیلی زیاد نباشد و در بخش ۳ برای بررسی چگونگی افزایش زمان اجرا به ازای تعداد مختلف خوشه‌ها به صورت مفصل توضیحاتی بیان شده است.

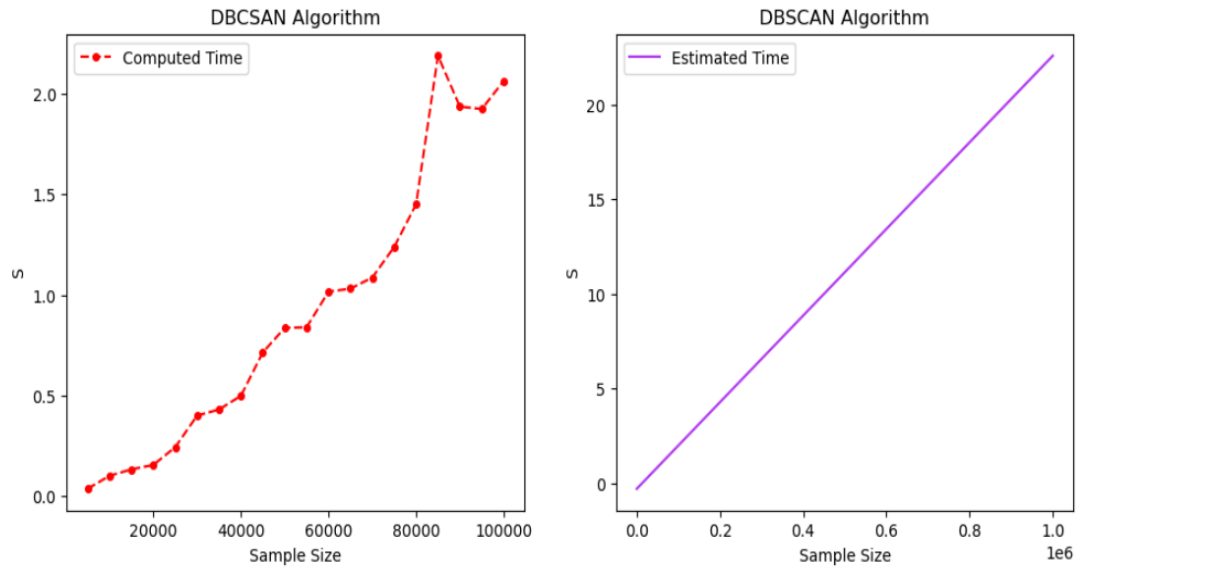
در شکل ۴.۳ زمان اجرای الگوریتم DBSCAN برای سایز مختلف داده های ورودی نشان داده شده است، همانطور که در شکل مشخص است تقریبا زمان اجرا به صورت خطی با افزایش تعداد داده های ورودی افزایش می یابد البته اثبات می‌شود که پیچیدگی محاسباتی الگوریتم DBSCAN در بدترین حالت O(n^2) است اما در مثال خاص به نظر می‌رسد زمان اجرا با افزایش تعداد داده های ورودی تقریبا به صورت خطی رشد می‌کند.
برای محاسبه زمان تقریبی این الگوریتم برای تعداد یک میلیون داده خط رگرسیونی بر زمان اجرا برازش داده شده‌است که نتایج آن در شکل ۴.۳ نشان داده شده است. به طور تقریبی زمان اجرای الگوریتم DBSCAN برای ۱ میلیون داده کمی بیشتر از ۲۰ دقیقه است.



شکل ۴.۱. تغییرات زمان اجرای الگوریتم kmeans , mini batch kmeans به ازای افزایش تعداد داده



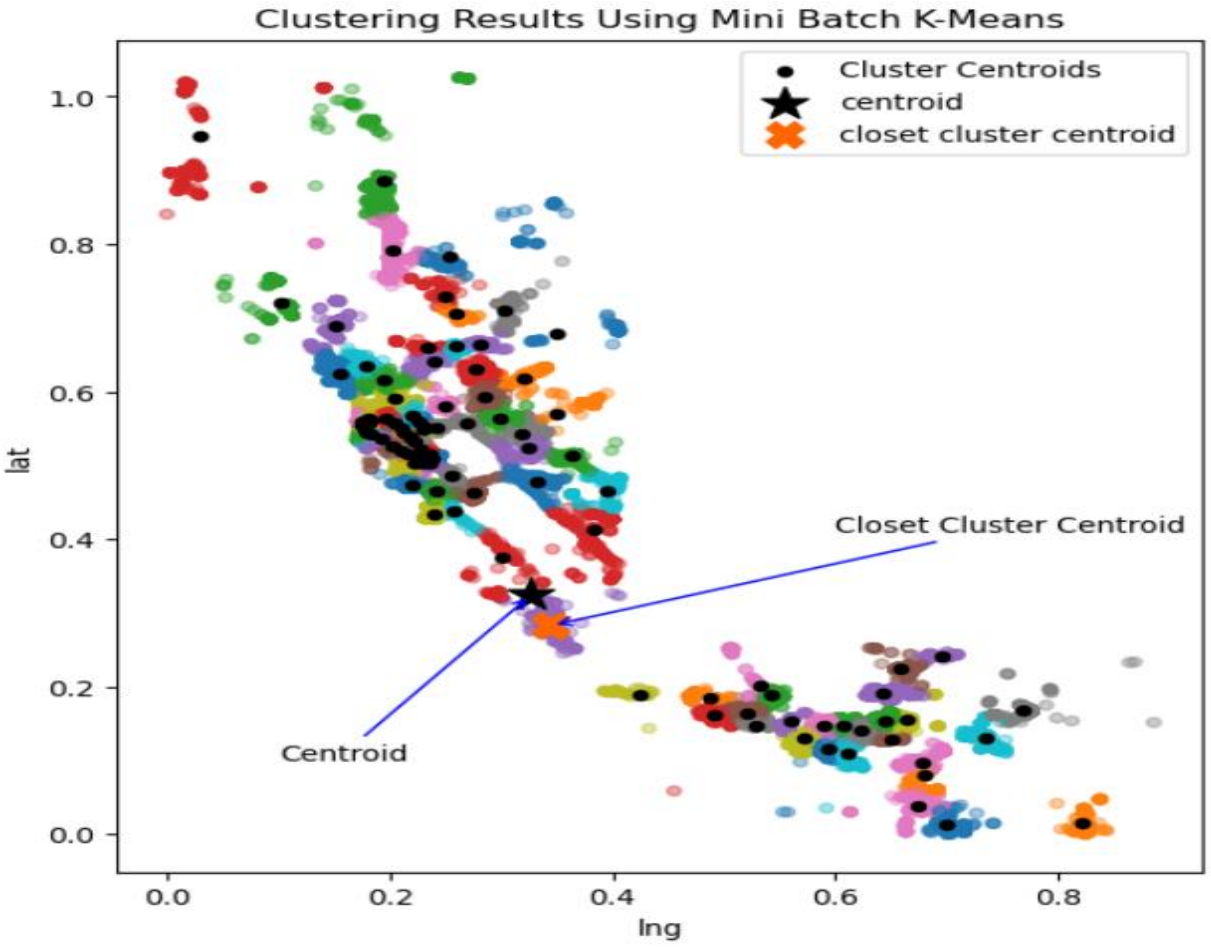
شکل ۴.۲. تغییرات زمان اجرای الگوریتم kmeans , mini batch kmeans به ازای افزایش تعداد خوشه ها



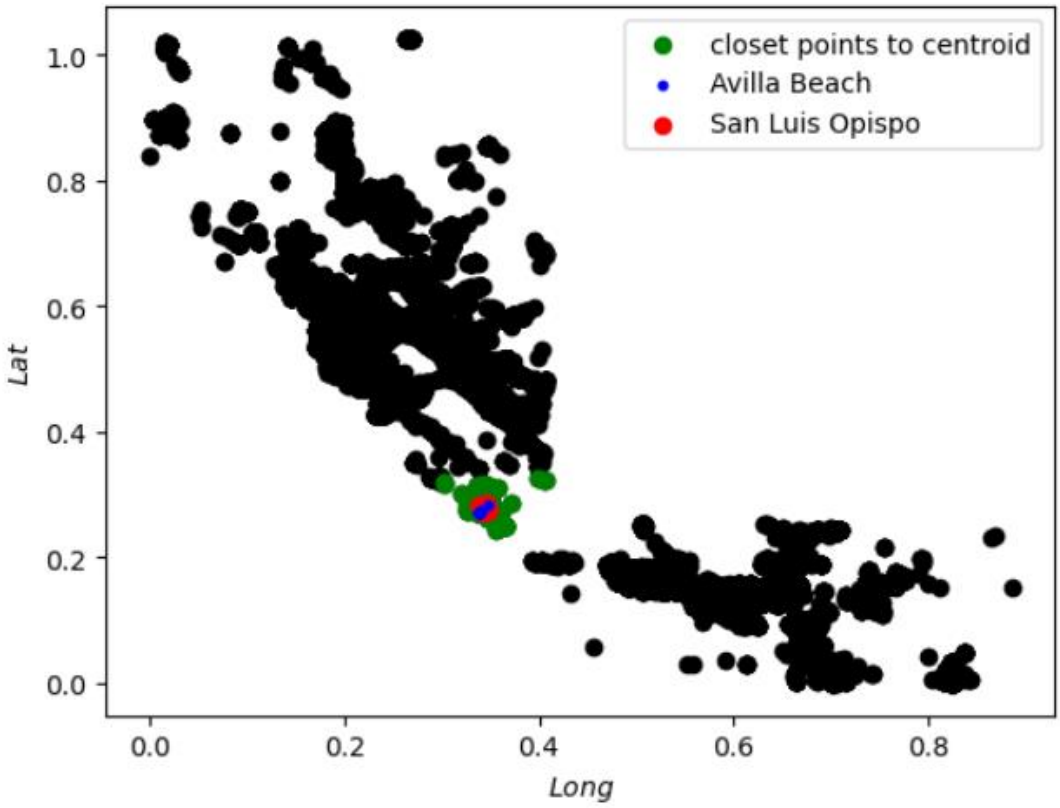
شکل ۴.۳ تغییرات زمان اجرای الگوریتم DBSCAN به ازای افزایش تعداد داده

۵. یافتن ارتباط

برای حل این قسمت از نتایج خوشه بندی روش Mini batch K-means استفاده شده است، در شکل ۵.۱ مختصات مرکز شماره ۲ و همچنین مرکز نزدیک ترین خوشه به این نقطه نشان داده شده است. اطلاعات کامل از جمله مختصات مرکز و برچسب مربوط به آن در کد وجود دارد. همچنین در شکل نقاط موجود در نزدیک ترین خوشه به این نقطه پیدا شده اند که نتایج در شکل ۵.۲ نشان داده شده است. در مجموع تعداد ۲۷۶ داده در این خوشه قرار دارند. از نظر جغرافیایی این خوشه در San Luis Obispo قرار دارد. پس از بررسی های انجام شده در متن توییت ها ۱۴ توییت به Avilla Beach اشاره شده که یکی از سواحل در نزدیکی این منطقه است و همچنین در ۸ توییت به San Luis Obispo اشاره شده است که در واقع نام این منطقه است، همچنین در ۶ توییت به شرایط آب و هوایی بد اشاره شده است و در ۳ توییت به pismo beach که نام ساحلی در این منطقه است. در شکل ۵.۲ امکان این توییت ها مرتبط نشان داده شده است.



شکل ۵.۱ نقطه مرکز و مرکز نزدیک ترین خوشه به آن



شکل ۵.۲ مرکز و نقاط موجود در نردیک ترین خوشه به آن و نقاط مربوط به هم