مبانی داده کاوی

دکتر ناظرفرد

گزارش تمرین دوم

حسین محمدی
۹۵۳۳۰۸۱

سوال اول)

هنگامی که اندازه داده ها بالا میرود ، به دلیل کوچک بودن حافظه نهان پردازنده ، امکان استفاده از همه داده ها یکجا نیست و موجب درخواست های زیاد روی حافظه اصلی میشود. در اثر این اتفاق زمان اجرای الگوریتم افزایش می یابد.

از راه حل های ممکن برای این مشکل استفاده از Mini Batch K-means میباشد. در این روش ما از دسته های کوچک تصادفی از داده ها با اندازه ثابت استفاده میکنیم. بنابراین هر دسته در حافظه نهان مستقر میشود و الگوریتم خوشه بندی فقط با آن دسته از داده سر و کار دارد.

در هربار اجرای الگوریتم از آن دسته برای به روز رسانی خوشه ها استفاده میشود و این روند استفاده از دسته ها تا زمان همگرایی ادامه می یابد.

هر دسته با استفاده از تابعی محدب از مقادیر نمونه های اولیه و داده ها و با استفاده از یک نرخ یادگیری که با تعداد تکرارها کاهش می یابد خوشه ها را به روز می کند.

که این نرخ یادگیری معکوس تعداد داده های اختصاص داده شده به یک خوشه در طی اجرای الگوریتم است.در نتیجه با افزایش تعداد تکرارها ، اثر داده های جدید کاهش می یابد و میتوان همگرایی را در خوشه ها تشخیص داد.

از معایب این روش آن است که ممکن است برخی داده ها از دست بروند. یعنی ممکن است خوشه بندی دارای خطا باشد و برخی داده ها در خوشه هایی جدا از خوشه ی اصلی بیفتند.

اما از آنجا که این خطا بسیار ناچیز است بنابراین در مقابل صرفه جویی زمان نادیده گرفته میشود.

اثر اندازه دسته ها ، هنگامی که تعداد خوشه ها افزایش می یابد ، بیشتر میشود.

این بدان معناست که با افزایش تعداد خوشه ها ، شباهت جواب Mini Batch K-means به جواب K-means کم میشود.

با وجود این که توافق بین خوشه ها با افزایش تعداد خوشه ها کاهش می یابد ولی عملکرد هدف با همان نرخ کاهش نمی یابد .معنی این حرف آن است که خوشه های نهایی در عین تفاوت از نظر کیفی به هم نزدیک تر هستند.

سوال دوم)

برای بهبود انتخاب نقاط اولیه در خوشه بندی K-means از روش های زیر میتوانیم استفاده کنیم.

۱ – انتخاب مراکز به صورت تصادفی از بین خود داده ها به جای انتخاب تصادفی از محدوده ی داده ها

۲ – K-means++

در این روش سعی میشود مراکز اولیه با بیشترین فاصله نسبت به یکدیگر انتخاب شوند. بدین صورت که ابتدا یک نقطه به صورت تصادفی انتخاب میشود سپس برای انتخاب نقاط بعدی از احتمال استفاده میکنیم.

هر چه نقطه جدید نسبت به نقطه قبلی و نزدیکترین نقطه فاصله اش بیشتر باشد آنگاه احتمال انتخاب آن نقطه بیشتر خواهد بود.

نقطه به نقطه جلو می رویم تا همه مراکز اولیه مورد نیاز ما حاصل شوند.

سوال سوم)

A) درست. هسته نقطه ای است که حداقل تعداد نقطه همسایه در شعاع اپسیلون وجود داشته باشد و ارضا کند.همچنین نقاطی وجود دارند که در همسایگی اپسیلون آنها نقاط همسایه هست اما شرط حداقل تعداد همسایه در اطراف آنها رعایت نشده.اگر این نقاط در همسایه هایشان ، یک هسته داشته باشند آنگاه جزو خوشه به حساب آورده میشوند. در غیر این صورت outlier محسوب میشوند و از خوشه خارج میشوند.

B) نادرست. این الگوریتم توانایی شناسایی خوشه های فارغ از شکل خوشه ها و داده ها را داراست.

C) نادرست. این الگوریتم از مرتبه زمانی O(n log(n)) میباشد.

D) درست. نیازی به دانش قبلی درباره تعدا خوشه ها ندارد و صرفا با داده های همسایه نقاط سر و کار دارد.

E) درست. این الگوریتم توانایی تشخیص چگالی داده ها را دارد و میتواند داده های که در محیط غیرچگال وجود دارند را تشخیص داده و به عنوان داده outlier معرفی کند.

سوال چهارم)

a) single Link

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

d((A , B) , C) = min(d(A , C) , d(B , C)) = 0.25

d((A , B) , D) = min(d(A , D) , d(B , D))= 0.16

d((A , B) , E) = min(d(A , E) , d(B , E))= 0.28

d((A , B) , F) = min(d(A , F) , d(B , F))= 0.34

	A,B	C	D	E	F
A,B	0				
C	0.25	0			
D	0.16	0.14	0		
E	0.28	0.70	0.45	0	
F	0.34	0.93	0.20	0.67	0

d((C , D) , (A , B))= min(d(C , (A , B)) , d(D , (A , B))) = 0.16

d((C , D) , E)= min(d(C , E) , d(D , E)) = 0.45

d((C , D) , F)= min(d(C , F) , d(D , F))= 0.2

	A,B	C,D	E	F
A,B	0			
C,D	0.16	0		
E	0.28	0.45	0	
F	0.34	0.20	0.67	0

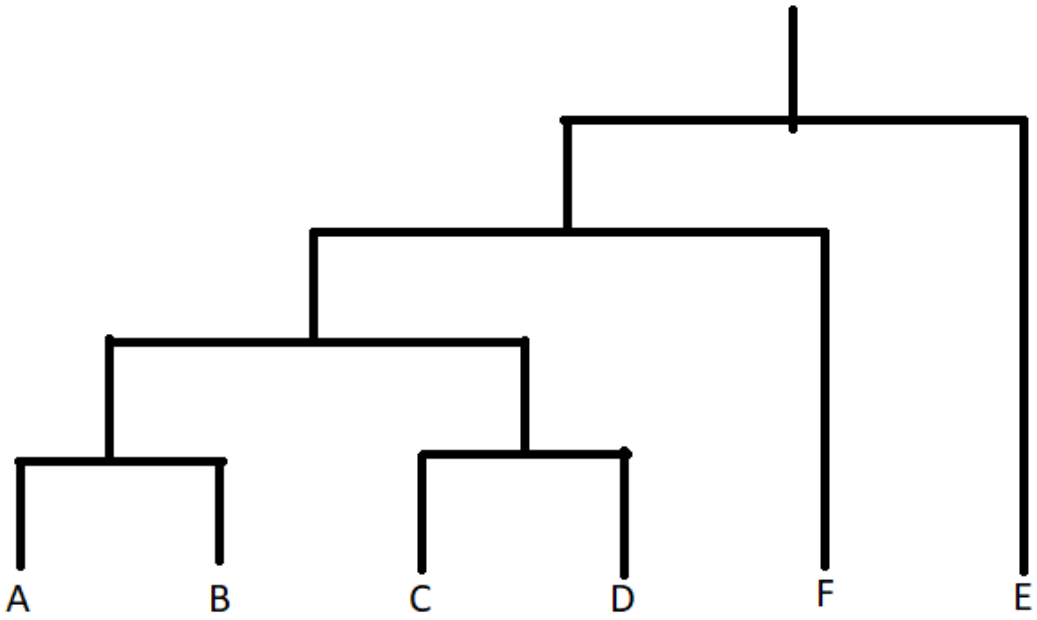
d((A , B , C , D) , E)= min(d((A , B) , E) , d((C , D) , E)) = 0.28

d((A , B , C , D) , F)= min(d((A , B) , F) , d((C , D) , F))= 0.20

	A,B,C,D	E	F
A,B,C,D	0		
E	0.28	0	
F	0.20	0.67	0

d((A , B , C , D , F) , E) = min(d((A , B , C , D) , E) , d((A , B , C , D) , F)) = 0.28

	A,B,C,D,F	E
A,B,C,D,F	0	
E	0.28	0



b) Complete Link

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

d((A , B) , C) = max(d(A , C) , d(B , C)) = 0.51

d((A , B) , D) = max(d(A , D) , d(B , D)) = 0.84

d((A , B) , E) = max(d(A , E) , d(B , E)) = 0.77

d((A , B) , F) = max(d(A , F) , d(B , F)) = 0.61

	A,B	C	D	E	F
A,B	0				
C	0.51	0			
D	0.84	0.14	0		
E	0.77	0.70	0.45	0	
F	0.61	0.93	0.20	0.67	0

d((C , D) , (A , B))= max(d(C , (A , B)) , d(D , (A , B))) = 0.84

d((C , D) , E)= max(d(C , E) , d(D , E)) = 0.7

d((C , D) , F)= max(d(C , F) , d(D , F))= 0.93

	A,B	C,D	E	F
A,B	0			
C,D	0.84	0		
E	0.77	0.70	0	
F	0.61	0.93	0.67	0

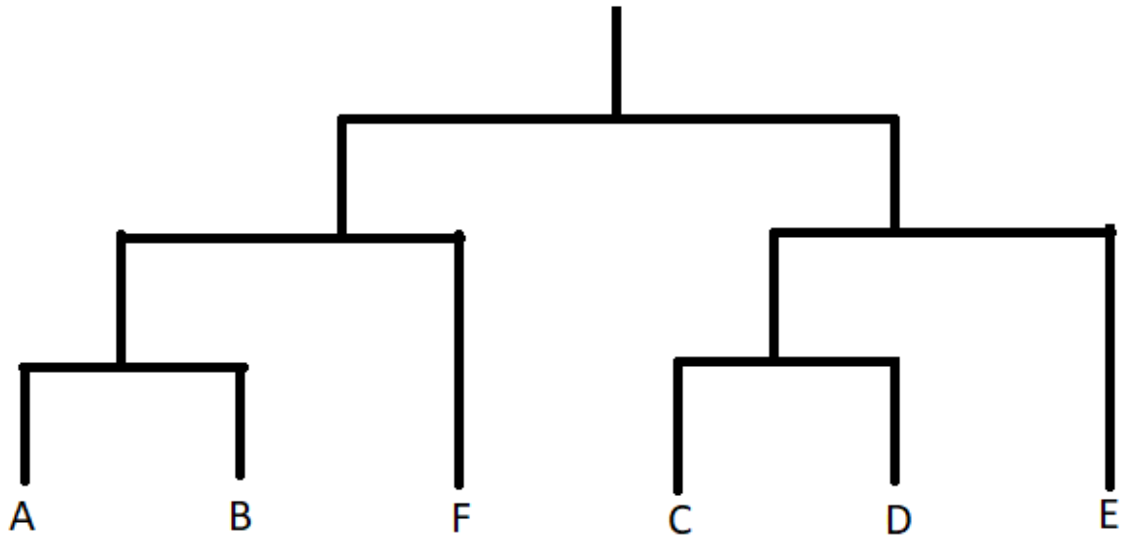
d((A , B , F) , (C , D))= max(d((A , B) , (C , D)) , d(F , (C , D))) = 0.93

d((A , B , F) , E) = max(d((A , B) , E) , d(F , E)) = 0.77

	A,B,F	C,D	E
A,B,F	0		
C,D	0.93	0	
E	0.77	0.70	0

d((A , B , F) , (C , D , E)) = max(d((A , B , F) , (C , D)) , d((A , B , F) , E)) = 0.93

	A,BF	**C,D,E**
A,B,F	0	
C,D,E	0.93	0



سوال پنج)

A) فقط DBSCAN. زیرا این شکل به صورت خطی قابل تقسیم نیست.

B) هردو. زیرا هم بین خوشه ها نواحی چگال وجود ندارد و هم آنکه با ذات K-means که همه جهات برای از اهمیت یکسان برخوردارند سازگاری دارد.

C) K-means. زیرا بین خوشه ها و نواحی عملا مرزی وجود نداره و چگالی یکسانی در مزرها با داخل خوشه ها وجود دارد که DBSCN احتمال بسیار بالا آنها را به هم متصل خواهد نمود.

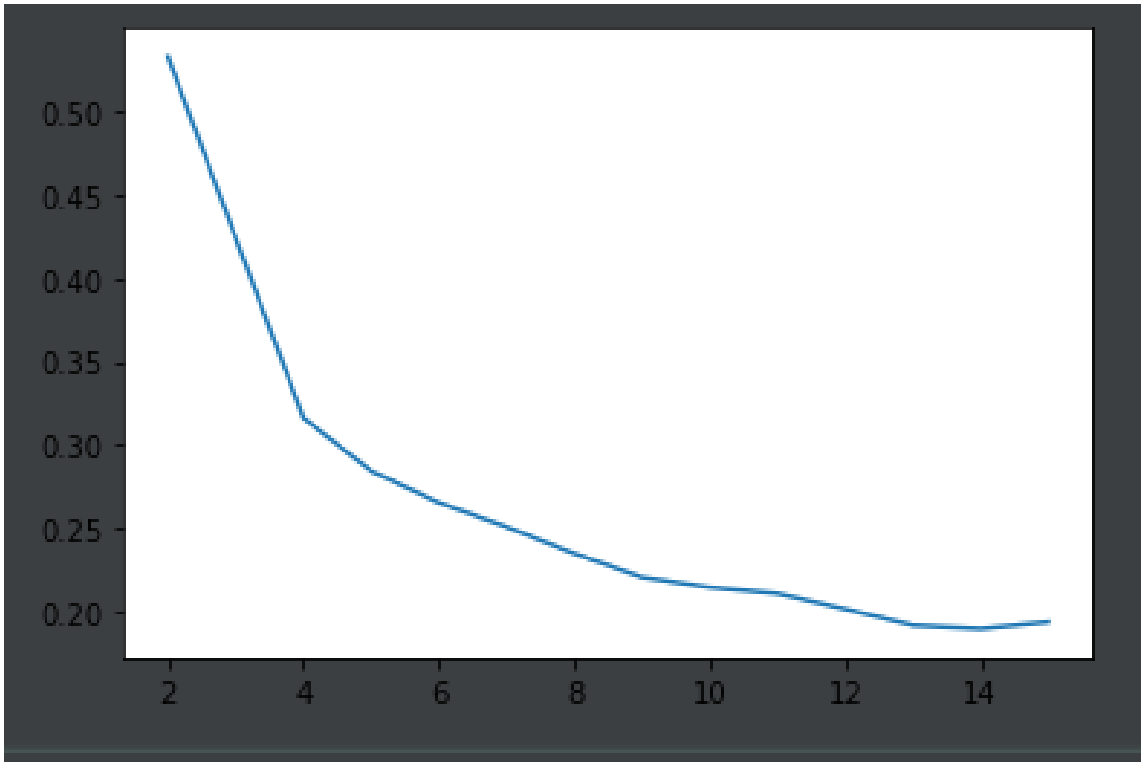
D) DBSCAN. به دلایلی که در قسمت اول شرح داده شد.

گزارش پیاده سازی)

۱ – الگوریتم k-means

در پیاده سازی این قسمت برای مشخص کردن مراکز اولیه از میان داده ها ، به صورت تصادفی مراکز انتخاب میشوند.

الگوریتم را برای تعداد خوشه های بین ۲ تا ۱۴ اجرا میکنیم که نمایش خوشه ها برای همه تعداد خوشه ها در فایل kmean.ipynb موجود است. در اینجا صرفا مقدار خطای خوشه بندی را بررسی میکنیم.

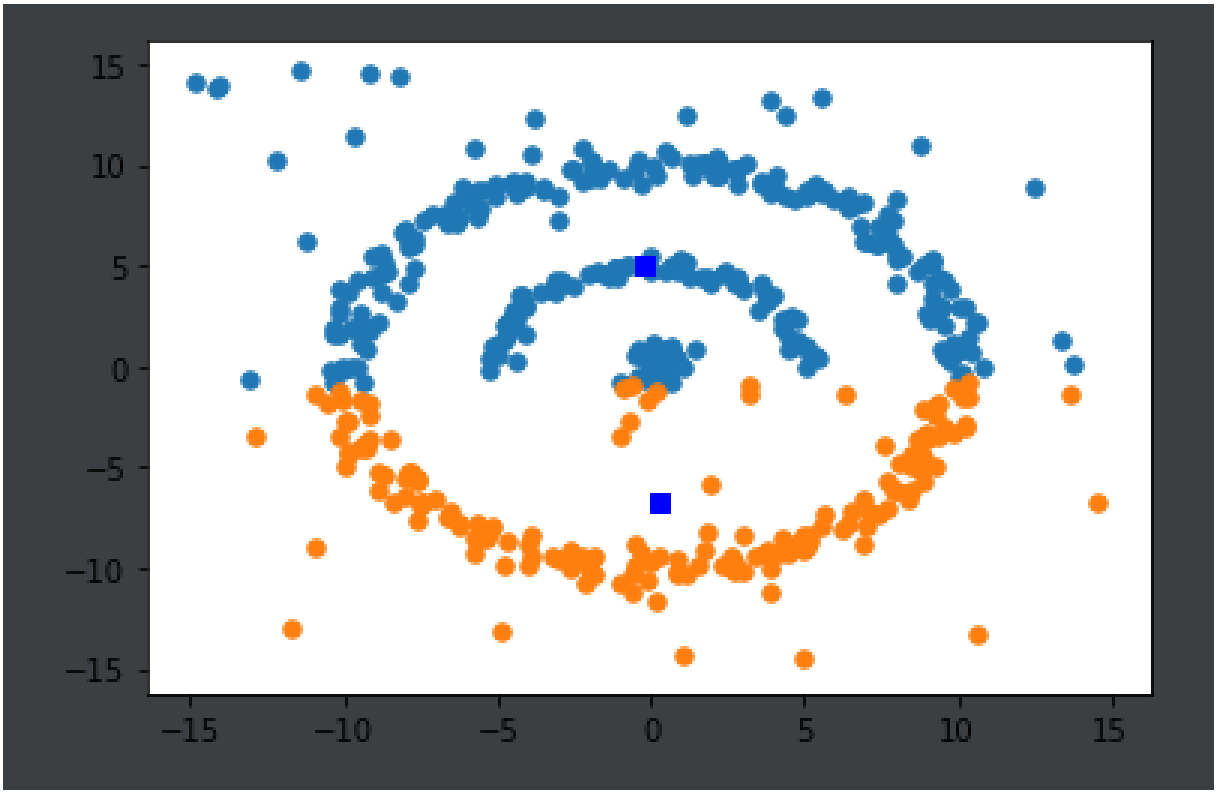


همان طور که میبینیم از تعداد خوشه ۴ به بعد تغییر خطا به اندازه تغییر آن از ۲ به ۴ نیست.

حال فراخور زمان و کیفیت خوشه بندی مورد نیاز میتوان حتی تعداد خوشه ها را تا ۸ خوشه نیز افزایش داد. زیرا از ۴ به ۸ نیز شاهد کاهش خطای مناسبی هستیم. از ۸ به بعد نیز شیب نمودار کم میشود که نشان دهنده تغییر جزئی خطا میباشد و از آنجا که با افزایش تعداد خوشه ها زمان الگوریتم افزایش می یابد ، افزایش تعداد خوشه ها بیشتر از ۸ منطقی نمیباشد.

در مورد DataSet2 نیز با بررسی چشمی متوجه وجود دو خوشه میشویم.

اما همانطور که در زیر میبینیم الگوریتم با تعداد خوشه ۲ خروجی زیر را نشان میدهد که اصلا مناسب نیست.



میبینیم که حتی داده های پرت نیز جزو خوشه ها به حساب می آیند.

دلیل این امر نیز آن است که الگوریتم K-means تنها با فاصله نقاط سر و کار دارد و جهت نقاط برایش مهم نیست. صرفا سعی میکند مراکز را به وسط نواحی چگال ببرد. فارغ از هرشکلی که داشته باشد.

برای همین نیز در شکل بالا میبینیم که مراکز را در درون دایره اصلی مشخص کرده و خوشه ی داده را به نزدیکترین مرکز نگاشت میکند.

برای مثال یک نقطه دربالای دایره ای بیرونی فرض میکنیم و یک نقطه در پایین دایره بیرونی که هردو در یک خوشه قرار دارند.میدانیم فاصله ی بین این دو نقطه به مراتب بیشتر از فاصله نقطه بالایی از از مرکز خوشه اش است.

اینجاست که ضعف الگوریتم در مبنا قرار دادن فاصله به عنوان تنها فاکتور نمایان میشود.

۲- DBSCAN

در این قسمت از هردو فایل covid و covid-sample استفاده شده.

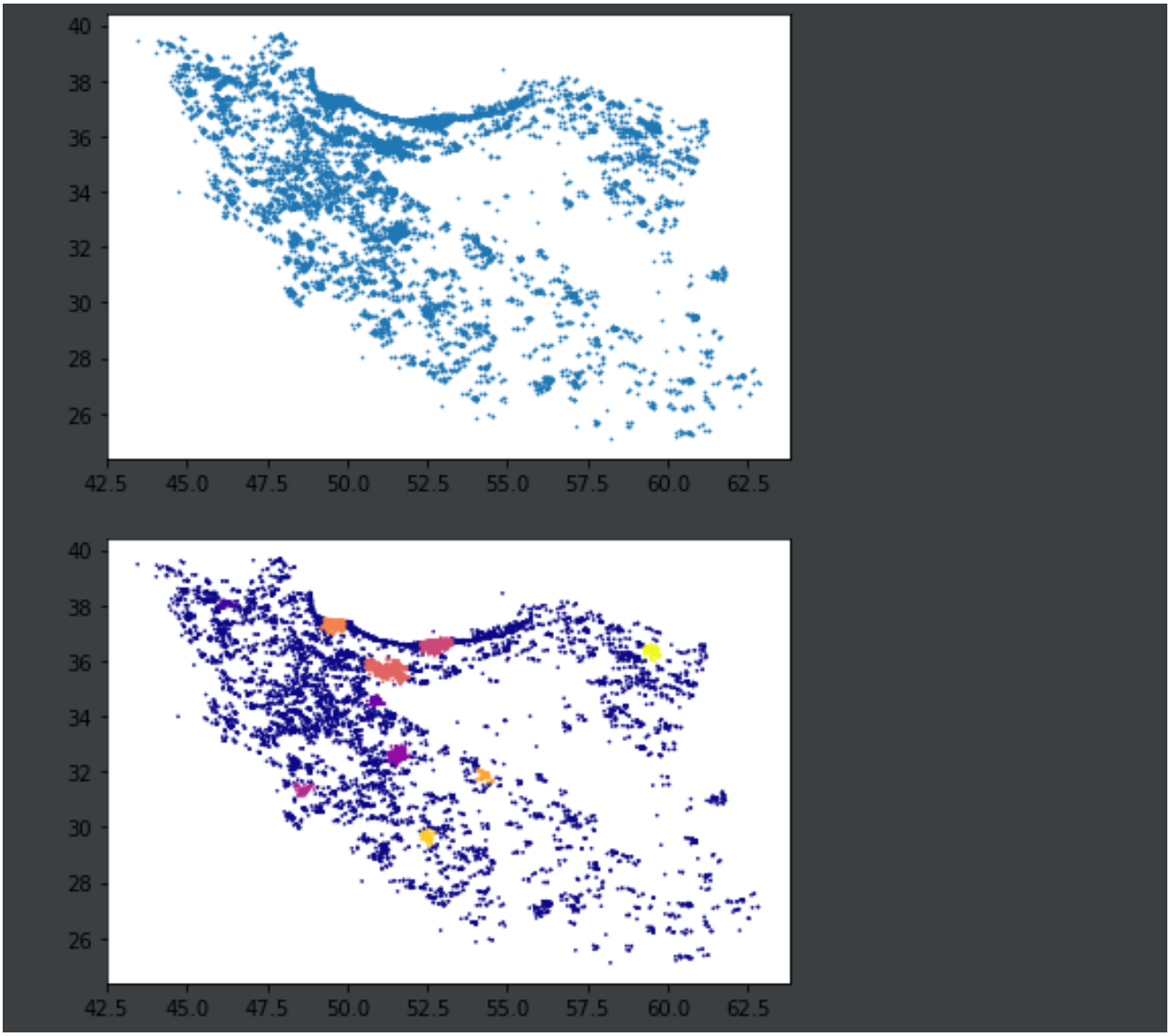
(برای اجرا حتما از فایلهای درون پوشه استفاده شود.زیرا فایلهای اولیه نام ستون نداشتند)

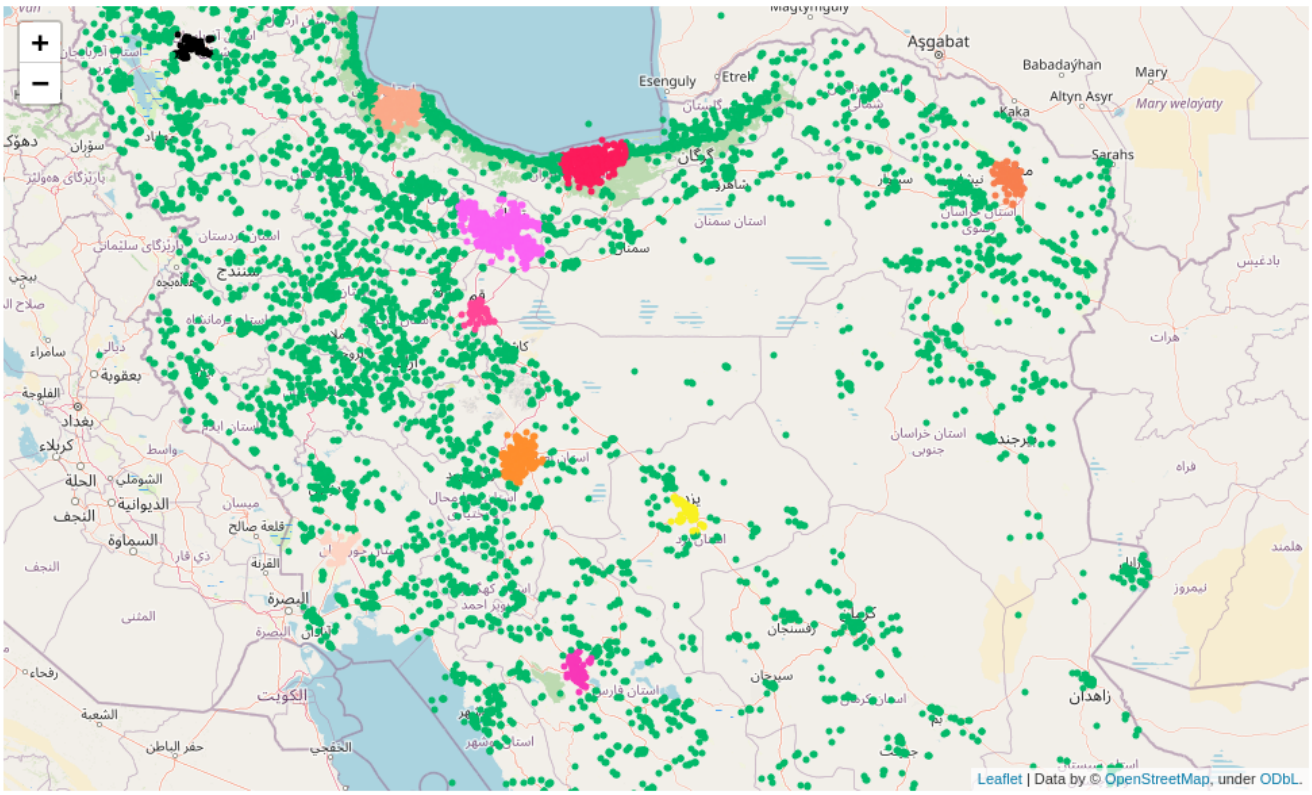
از آنجا که داده ها به صورت اعداد اعشاری هستند پس میتوان فضای داده ها را یک فضای دوبعدی فرض کرد و از الگوریتم استفاده کرد.

برای شناسایی مراکز شیوع مقدار با سعی و خطا و همچنین در نظر گرفتن اندازه طول و عرض جغرافیایی محدوده شهرها (۰/۵) به اعداد زیر برای دو دیتا میرسیم.

covid.csv

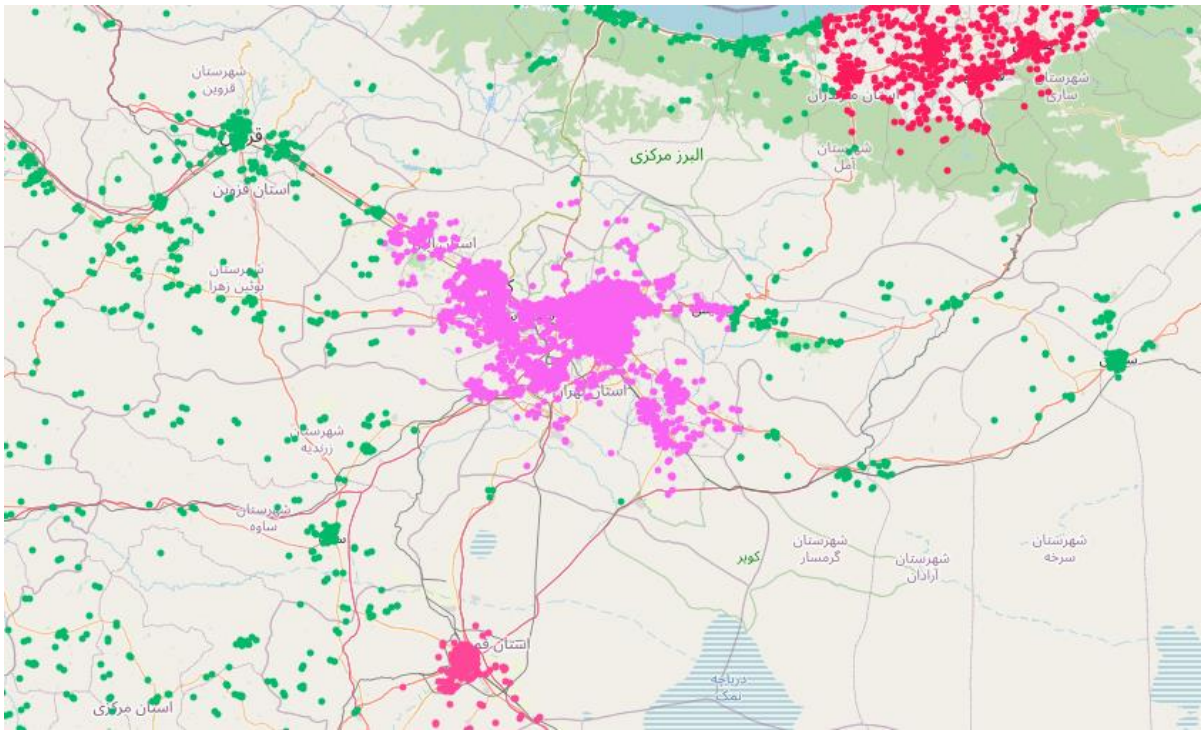
```
dbscan = DBSCAN(eps=0.2, min_samples=200)
```





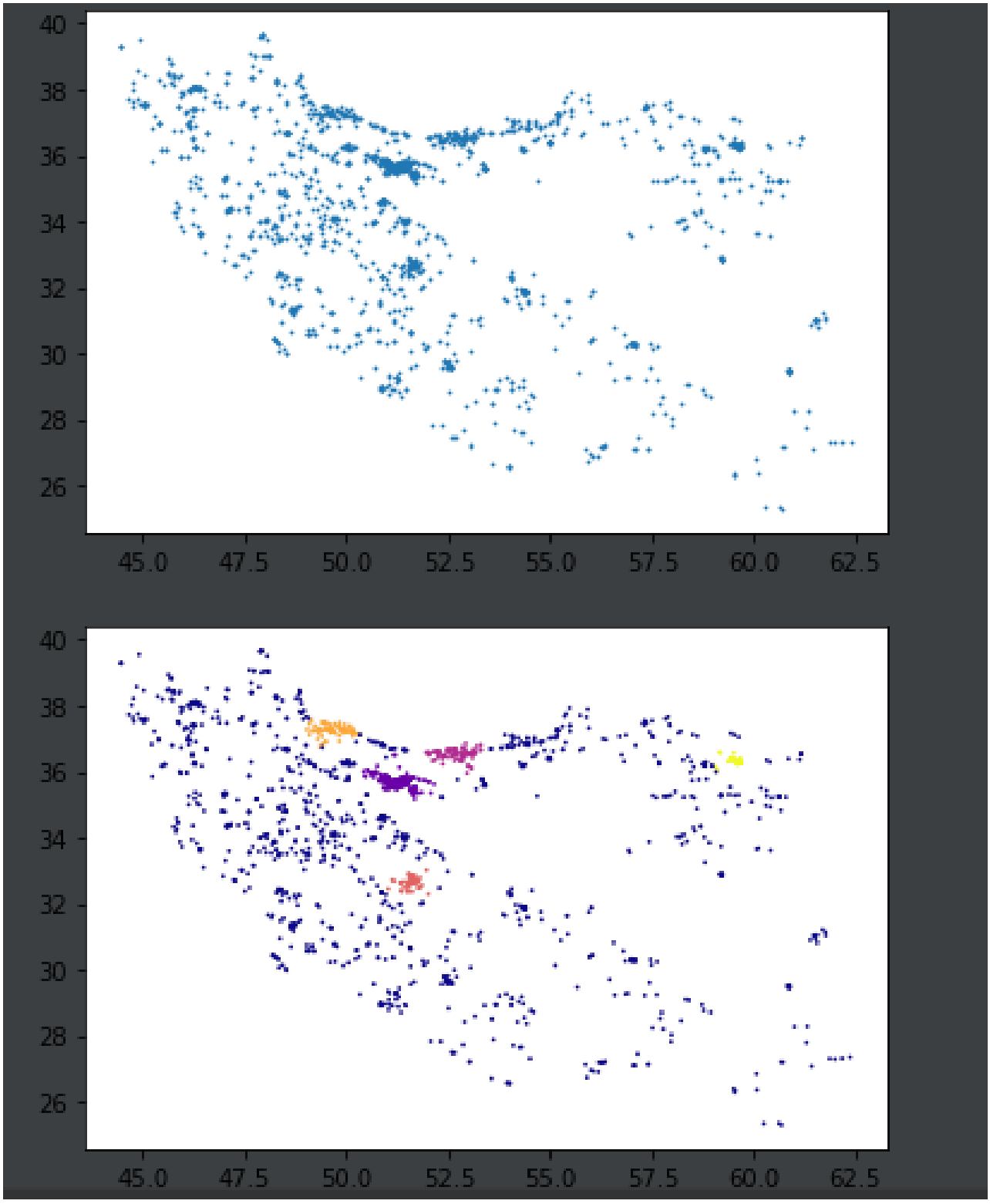
که شاهد شناسایی تعداد زیادی از مراکز استان به عنوان کانون شیوع هستیم.

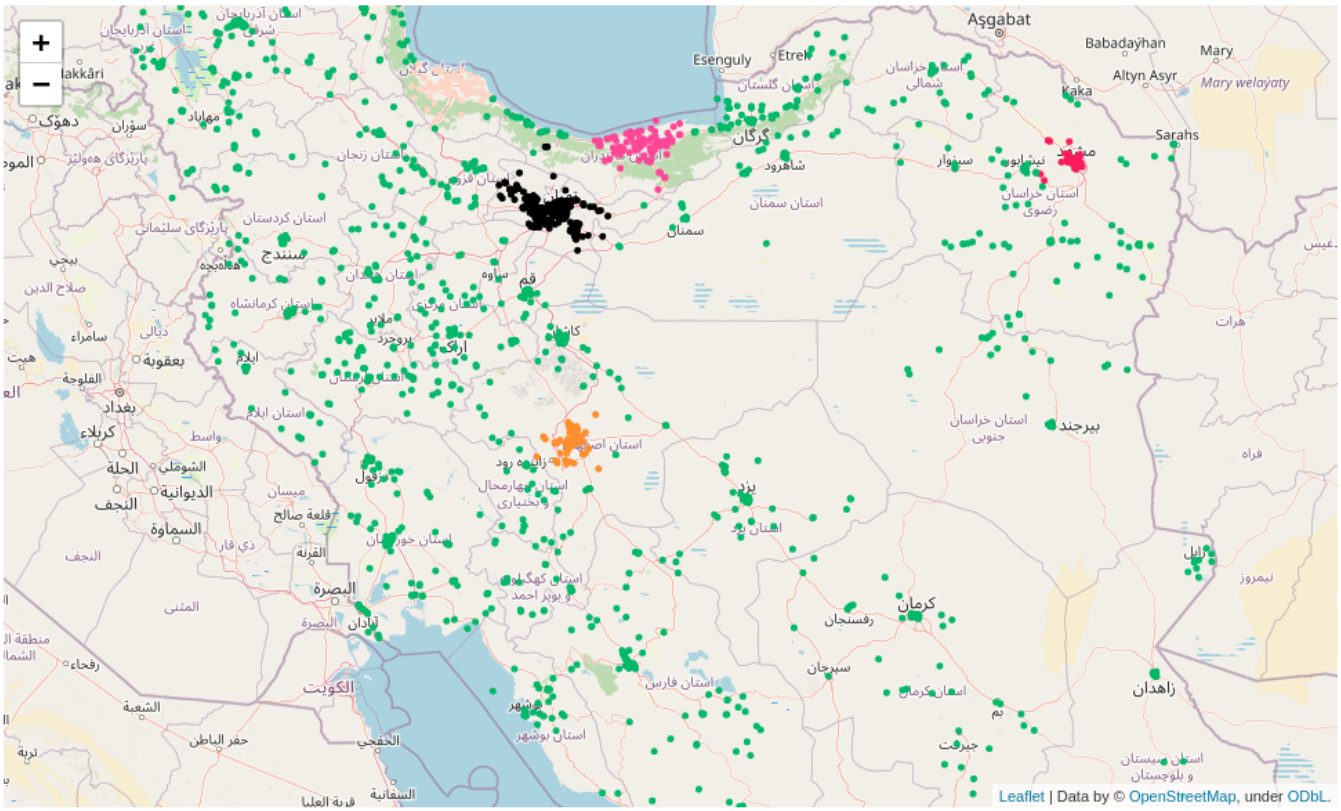
همچنین در شمال کشور به دلیل فاصله کم شهرها از یکدیگر میتوانیم کانون بزرگتری نیز شناسایی کنیم اما به قدری نیست که در الگوریتم شناسایی شود و با افزایش شعاع یا کاهش حداقل نقطه میتوان آنها را به کانون شیوع افزود.



covid-sample.csv

dbscan = DBSCAN(eps=0.4, min_samples=50)




Aşgabat
Babadaýhan
Mary
Altyn Asyr
Mary welaýaty
Esenguly
Etrek
Kaka
Sarahs
گرگان
مشهد
سبزوار
شاهرود
سمنان
قم
ساوه
کاشان
اراک
همدان
کرمانشاه
سنندج
بعقوبة
بغداد
کربلاء
الحلة
النجف
السماوة
الديوانية
الشوملي
البصرة
الفرنة
قلعة صالح
العمارة
الكويت
الشعيبة
حفر الباطن
الجهراء
السفانية
بيرجند
يزد
کرمان
رفسنجان
سیرجان
زاهدان
جیرفت
بم
Leaflet | Data by © OpenStreetMap, under ODbL.

۳ – image compression

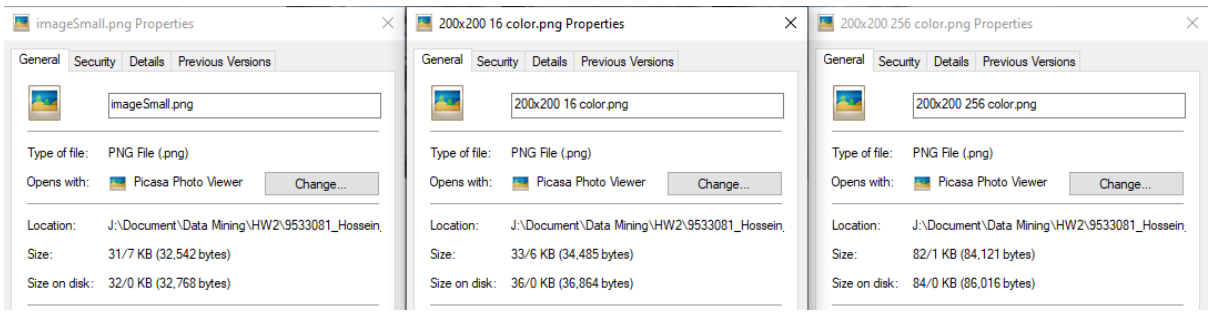
در این مرحله فشرده سازی تصویر را انجام میدهیم.

شایان ذکر است به دلیل ضعف منابع سخت افزاری تصویر بزرگتر با مقادیر رنگ کمتری فشرده سازی شده است.

در اینجا تصاویر و فشرده شده آن ها را مشاهده و مقایسه میکنیم.

تصویر کوچکتر ۲۰۰*۲۰۰

Original | 16 color | 256 color



همانطور که مشاهده میکنیم حجم ۱۶ رنگ ۳۳ کیلوبایت و حجم ۲۵۶ رنگ ۸۲ کیلوبایت میباشد که نشان دهنده اثر فشرده سازی میباشد.

البته چون روشهای ذخیره عکس متفاوت هستند شاهد برابری حجم ۱۶ رنگ و عکس اصلی هستیم.میتوان نتیجه گرفته عکس اصلی نیز از قبل فشرده شده است.

تصویر بزرگتر ۸۰۰*۸۰۰

Original

8 color

24 color