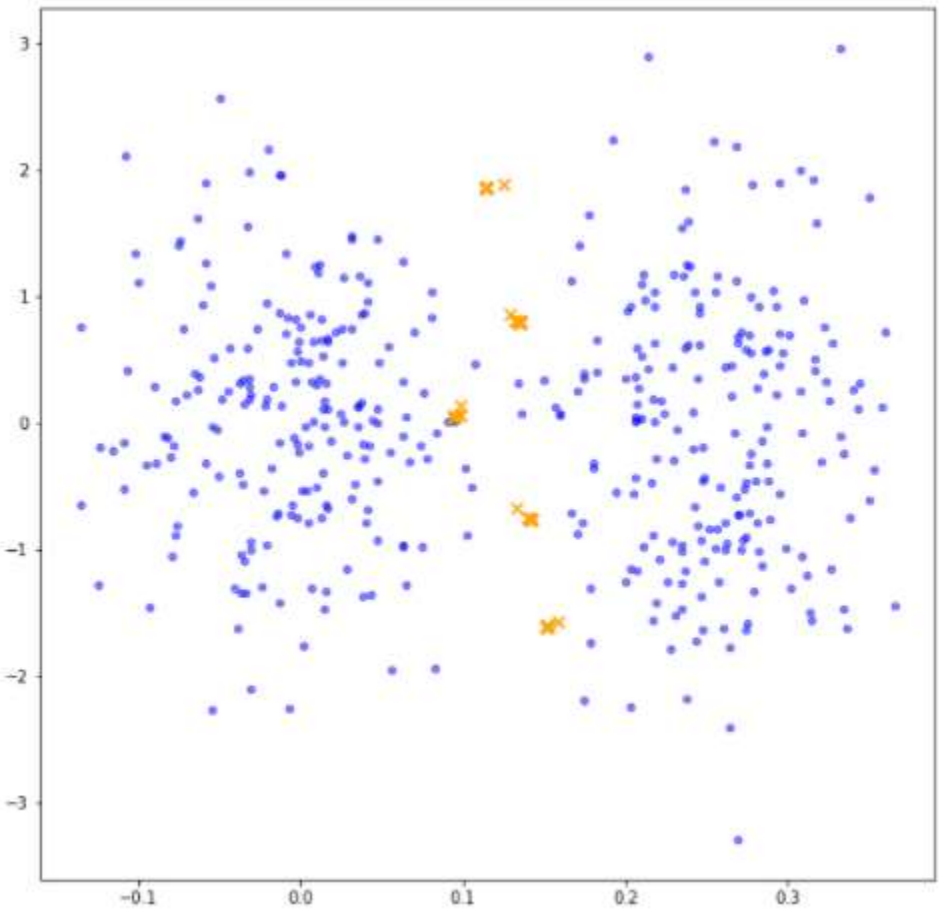
فایل ضمیمه : Clustering.ipynb

برای اینکه درک بهتری از چگونگی تقسیم بندی داده ها داشته باشیم، نتیجه ی آخرین خوشه بندی انجام شده در حلقه را در بخش های a,b,e,f روی نمودار با مرکز داده ی نظیرش در پایان توضیحات این بخش ها رسم می‌کنم و در c,d هم یک خوشه بندی با اعداد حاصل از نمودار می‌سازم و رسمش می‌کنم. بخش کد تابع plot_clusters در اول کار آمده در پایان این قسمت ها استفاده شده و نتیجه آن را اینجا در آخر هر بخش می‌آورم.

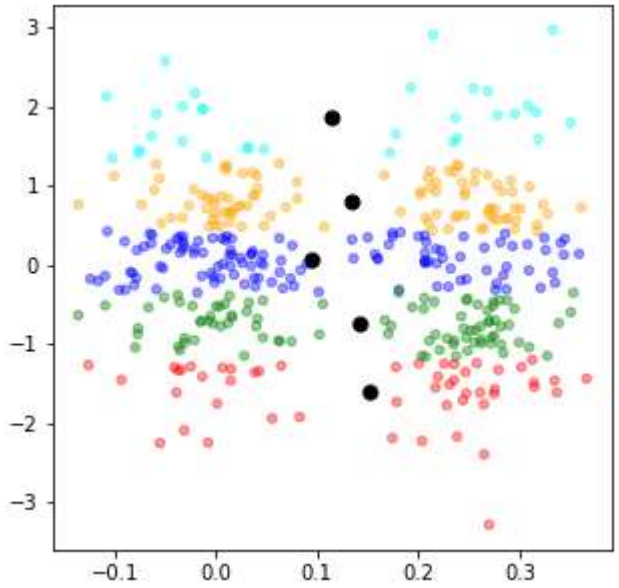
a) فایل first_clustering_dataset.csv به صورت یک نمودار پراکنش نگاست از ویژگی های ستون اول و دوم آن رسم شده. نقاط به صورت دایره های شفاف آبی رسم شده اند تا نقاط تمرکز داده ها و داده های روی هم افتاده بهتر نمایان شوند. با sklearn.cluster.KMeans که پیاده سازی همین Lloyd’s algorithm هست به تعداد ۲۰۰ بار متوالی با روش kmeans با انتخاب مرکز خوشه های اولیه ی تصادفی داده ها را به ۵ خوشه تقسیم کردم و هر بار ۵ مرکز به دست آمده را روی نمودار به صورت ضربدر نارنجی شفاف رسم کردم. این مراکز هر بار نزدیک به همدیگر پیدا شدند چنان که روی شکل دیگر حالت شفاف ندارند. به ازای هر یک از این ۵ مرکز جواب مساله ۲ نقطه ی تمرکز پیدا شده که آن ها هم نزدیک هم هستند.



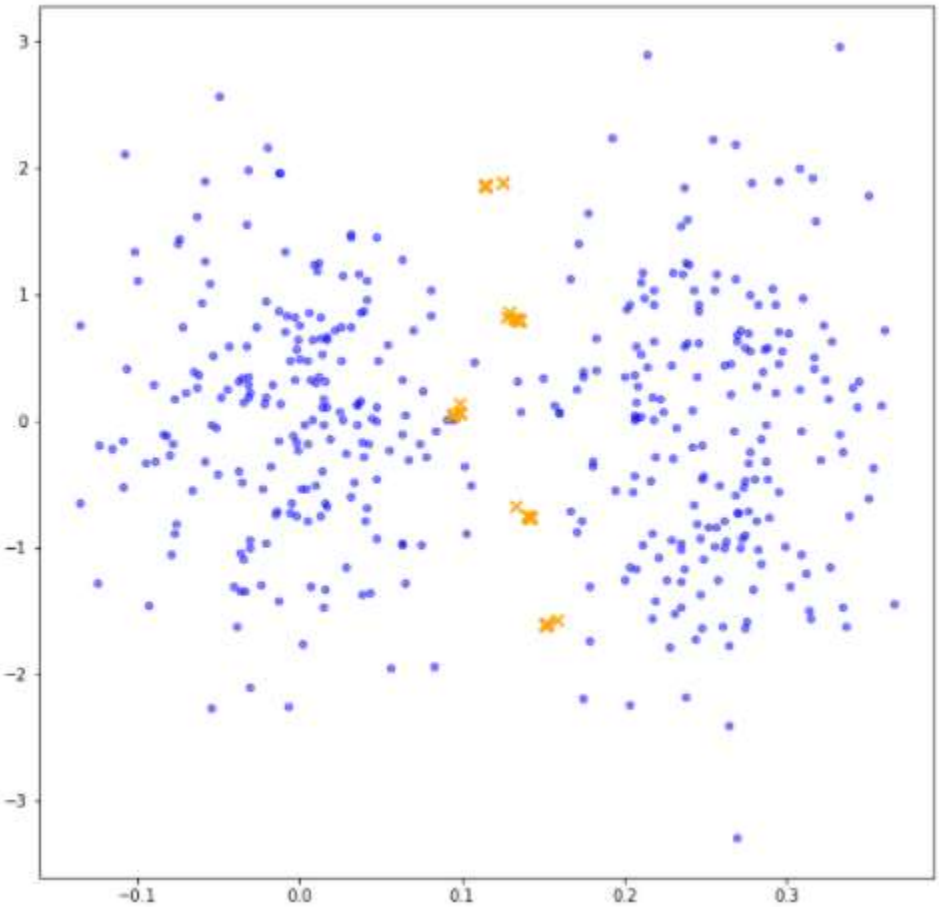
در هر بار خوشه بندی مقدار تابع هزینه $J(C,L)$ به دست آمده را در آرایه ی sse نگه داشتم. در نهایت روی این عناصر مقدار کمینه و میانگین و انحراف معیار را حساب کردم. میانگین خطا کم نیست اما کوچک بودن انحراف معیار همان تمرکز مرکزهای به دست آمده را که در شکل دیدیم نشان می‌دهد. با وجود تصادفی بودن انتخاب اول و حساسیت این الگوریتم روی این مقادیر اولیه ولی در این مساله چنان حساسیتی مشاهده نشد. داده های این مساله به چشم بیشتر به دو خوشه نزدیک اند.

```
within-cluster sums of squares over 200 runs :
minimum =  37.98746173779577
mean =  37.99487235518184
standard deviation =  0.0030894236022031905
```

نمودار آخرین خوشه بندی انجام شده روی حلقه :



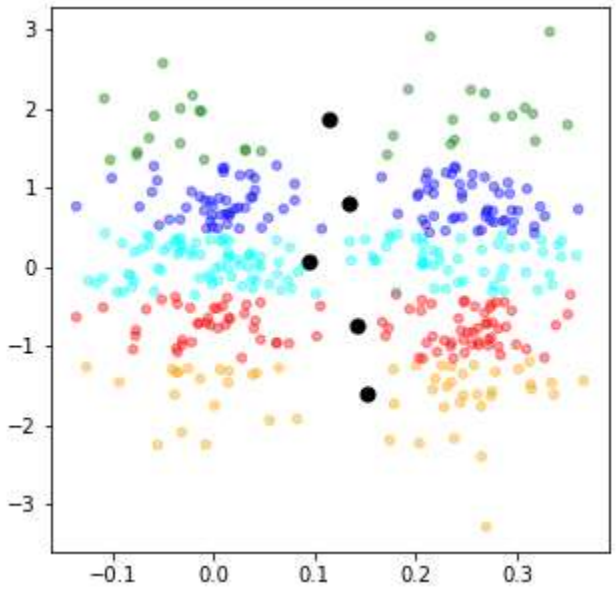
b) همان داده های قبلی این بار با روش مقداردهی اولی را به جای تصادفی، به روش هوشمنداده تری از روش Kmeans++ انتخاب کردیم. پیاده سازی این بخش هم عینا مانند بخش قبل است. نمودار را در زیر میبینید:



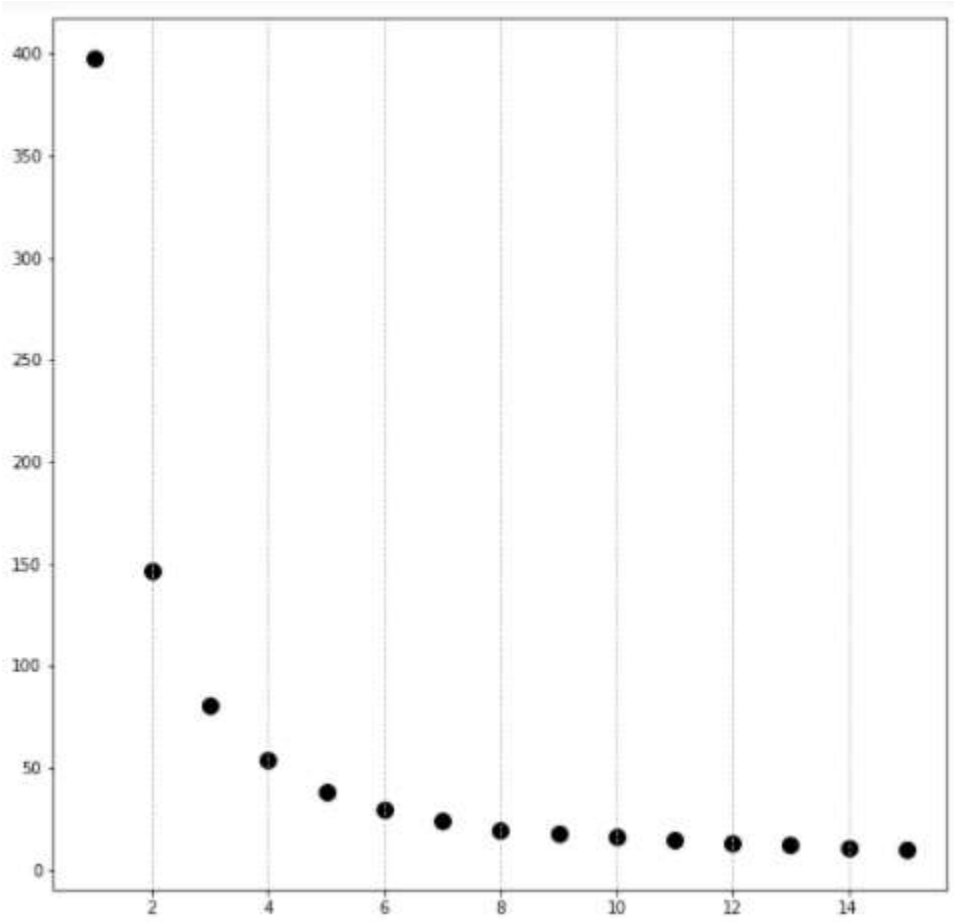
تفاوت این حالت و حالت قبلی بسیار کوچک و قابل چشم پوشی است. می‌توان روی این مساله هر دو روش را معادل دانست.

```
within-cluster sums of squares over 200 runs :
minimum =  37.98746173779577
mean =  37.9951726801767
standard deviation =  0.003841282711142167
```

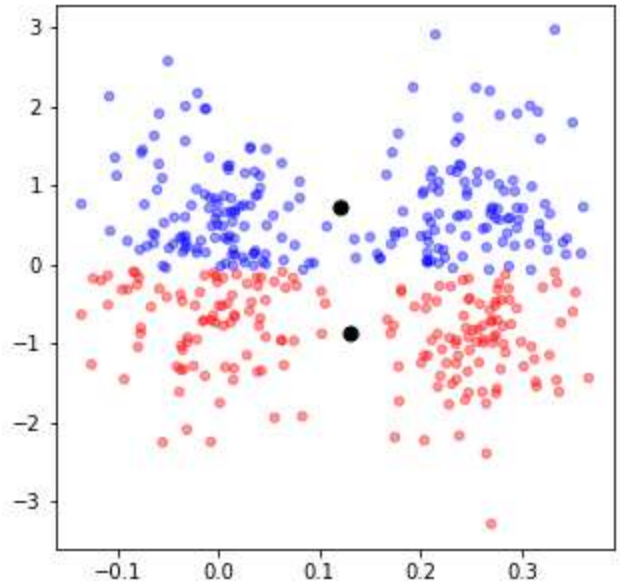
نمودار آخرین خوشه بندی انجام شده در حلقه :



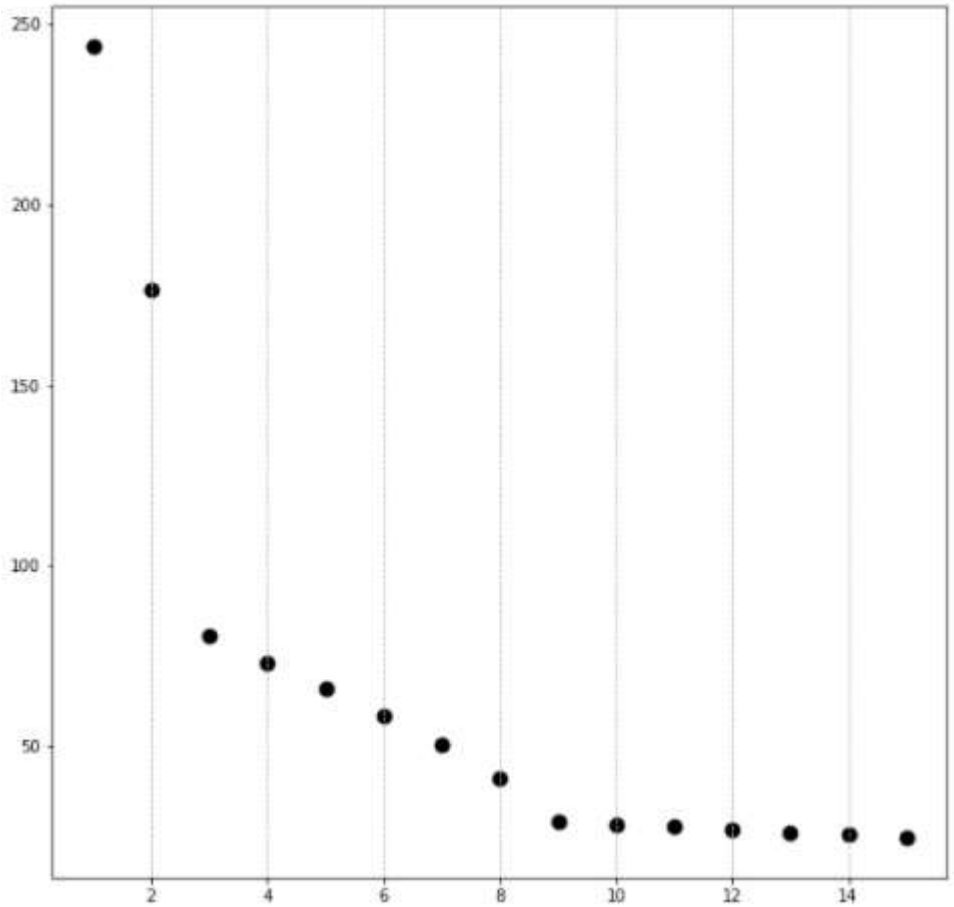
c) با همان داده های قبلی این بار هدف پیدا کردن آن نقطه ی knee روی نمودار $J(C,L)$ بر حسب تعداد خوشه است. برای هر k بین ۱ تا ۱۵ به تعداد ۲۰۰ بار به روش قسمت a با مرکز اولیه تصادفی خوشه بندی انجام دادم و کمترین مقدار $J(C,L)$ به دست آمده را روی نمودار ثبت کردم. برای این داده ها این عدد برابر ۲ هست. اگر به شکل داده ها که در بخش قبلی آمد هم نگاه کنیم این عدد اولین چنددستگی داده های این مساله ست که روی نمودار به چشم می‌آید.



یک خوشه بندی با k=2 برای مثال :

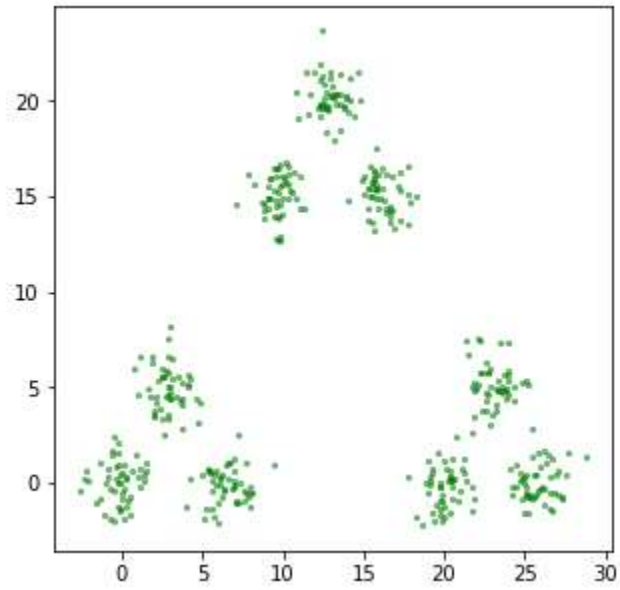


d) بخش c را این بار با داده های فایل *second_clustering_dataset.csv* *تکرار می‌کنم با این تفاوت که به جای کمینه ی* $J(C,L)$ مقدار جذر آن را که پیشنهاد شده روی نمودار ثبت می‌کنم. همان طور که در سوال گفته شده این نمودار دو نقطه‌ی منظور دارد که اولی که اختلاف بزرگتری را نشان می‌دهد برابر ۳ و دومی روی ۹ قرار دارد.

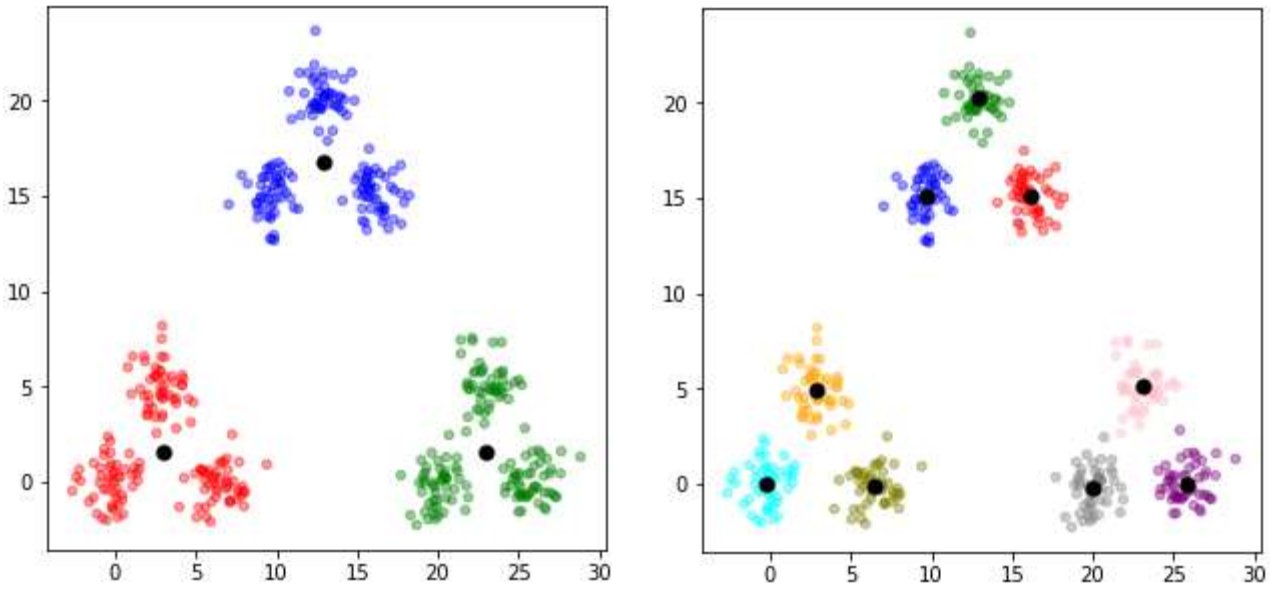


نمودار داده های مساله را که رسم کنم دلیل این اتفاق واضح تر می‌شود. در این داده ها سه خوشه ی بزرگ وجود دارد که کاملا جدا از یکدیگرند و اگر به صورت سلسله مراتبی بالا به پایین خوشه بندی را ادامه دهیم یا اینکه روی هر خوشه دوباره خوشه بندی انجام دهیم هر کدام از این ها می تواند به سه خوشه ی دیگر تقسیم شوند. یعنی داده های مساله یک خوشه بندی بهینه ی با ۳ خوشه دارد و این خوشه ها هرکدام ۳ زیرخوشه دارند که تمام این روی شکل داده های مساله قابل تشخیص است. حالا اگر با این توضیح به نمودار ریشه ی مجموع

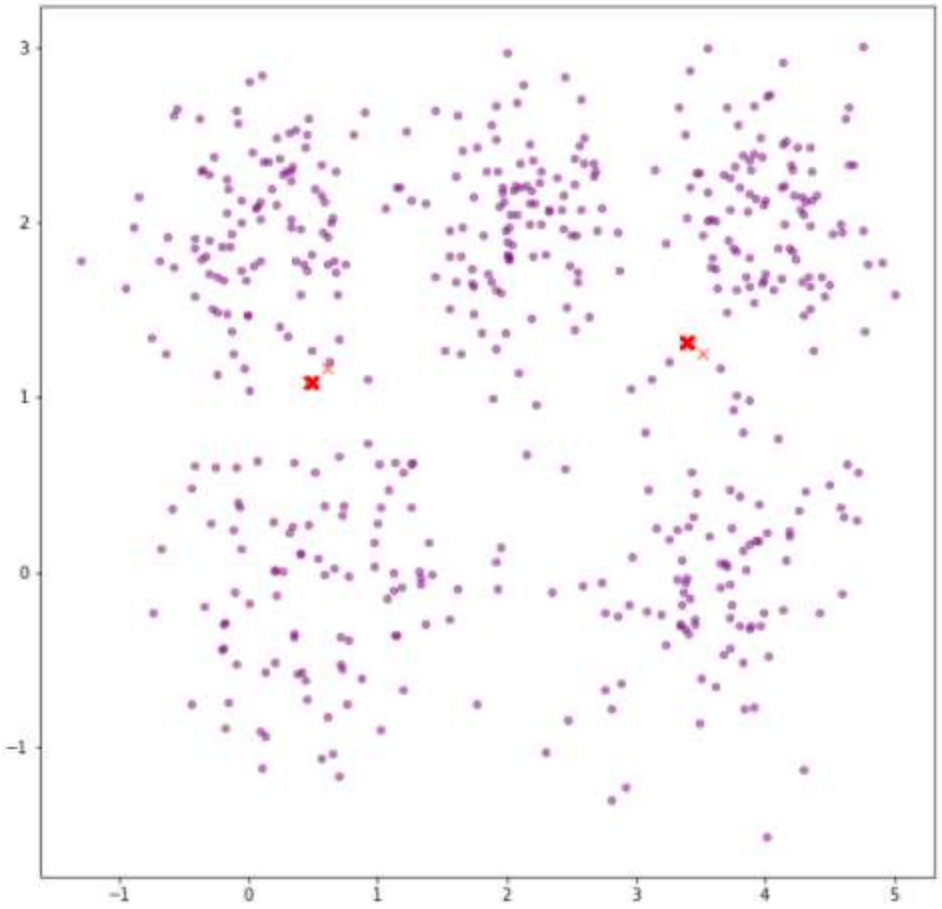
مربعات برگردیم اولین نقطه ای که افت مقدار شدید است روی ۳ هست که برابر با حالتی است که الگوریتم ۳ خوشه ی اصلی را تشخیص می دهد. و اگر به نقطه ی دوم کاهش ناگهانی شیب نگاه کنیم این عدد روی ۹ هست که برابر حالتی است که الگوریتم هر یک از زیرخوشه هایی که گفتیم را شناسایی کرده و حالا هر کدام را به عنوان یک خوشه گرفته.



و برای مثال یک خوشه بندی با اندازه ی ۳ و ۹ را روی این داده ها:

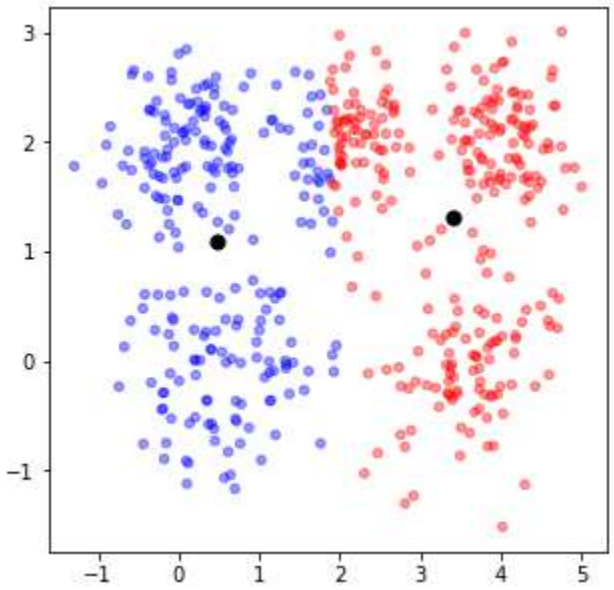


e) برای این بخش همان مراحل بخش a را برای داده های فایل third_clustering_dataset.csv تکرار می‌کنم. در این بخش تعداد خوشه ها را ۲ و تعداد دفعات خوشه بندی را ۵۰۰ بار با روش انتخاب مرکز اولیه تصادفی انجام می‌دهم.

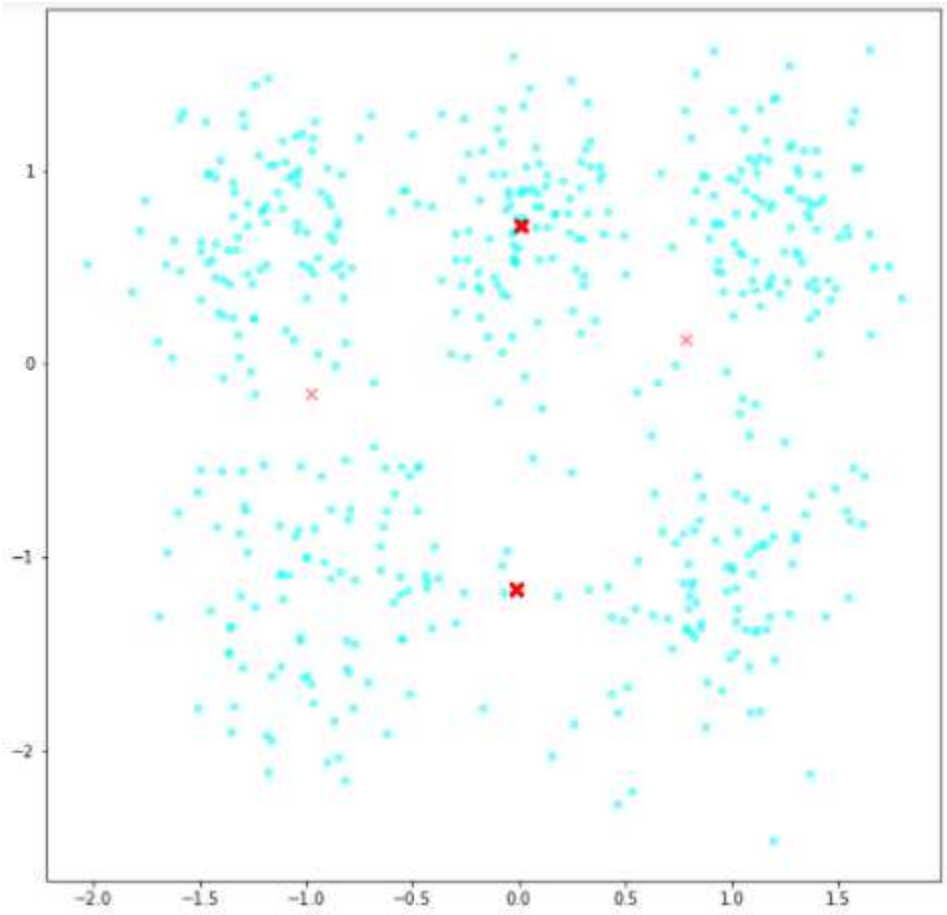


۵۰۰ مرکز به دست آمده برای هر خوشه در هر بار چنان نزدیک اند که روی این نمودار با این مقیاس می‌توانیم یک نقطه فرض کنیم (به استثنای یک مورد) . این دو نقطه روی محور عمودی چندان تفاوتی باهم ندارند پس می‌توان گفت که داده ها به صورتی خوشه بندی شده اند که از روی مقدار متغیر ستون اول از هم جدا شده اند. روی محور افقی یک خوشه سمت چپ و یک خوشه سمت راست داریم. اگر به داده ها دقت کنیم روی نمودار یک حالت ۵ دستگی قابل مشاهده ست.

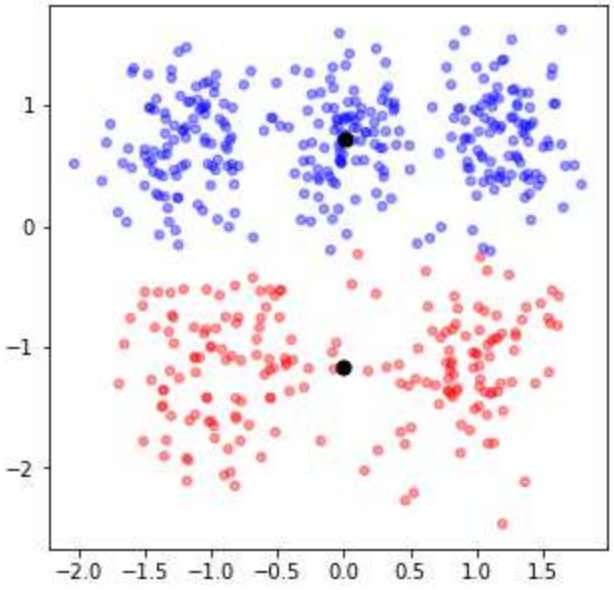
تصویر آخرین خوشه بندی انجام شده :



f) بخش e را این بار با یک مرحله اضافه تر تکرار می‌کنم و اول داده ها را با preprocessing.StandardScaler استاندارد می‌کنم چنان که روی هر ستون میانگین صفر و واریانس یک شود.



تفاوتی به لحاظ نسبت داده ها به همدیگر که ایجاد نشده. شکل نمودار همان قبلی است و همان ۵ دستگی روی آن مشاهده می شود و تنها مقیاس تغیر کرده اما اگر مراکز خوشه ها به یکدیگر را بگیریم در بیشتر اجراهای این بخش موقعیت مرکز ها نسبت به بقیه داده ها در مقایسه با حالت قبلی طوری است که انگار روی خط فرضی عمود بر آن قرار دارد. ۲ خوشه برای این مساله بهترین حالت نیست ولی با همین انتخاب در حالت قبلی می شود گفت که داده ها در دو خوشه ی چپ و راست نمودار تقسیم شده بودند و دو حالت بعد از نرمال سازی به دو خوشه ی بالا و پایین بخش شدند. که به نظر می‌رشد نتیجه بهتری باشد و چنان بدیهی تر از حالت اول است که از روی نمودار هم می شود متوجه شد. یعنی این بار بر خلاف بخش قبل عامل تعین کننده تر برای خوشه بندی راستای متغیر نظیر ستون دوم داده هاست. (البته در یکی از نتیجه ها از بین اجراهای مختلف مراکز شبیه بخشe هستند و این شاید به تصادفی بودن مدل برای انتخاب مرکز اولیه برمیگردد) مثلا همین که در خالت اولی بدون نرمال سازی یک توده ی داده ها در بالا و وسط نمودار دو دسته می شد در حالی که نزدیکی طبیعی به هم داشتند و بهتر بود در یک خوشه جا بگیرند. ولی در حالت دوم بعد از نرمال سازی تصور یک خط فرضی افقی جدا کننده ی دو خوشه روی داده ها منطقی تر به نظر می‌رسد. تصویر آخرین مدل خوشه بندی این قسمت را با e مقایسه کنیم موید همین نتیجه ست.


1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5