به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس شناسایی آماری الگو

استاد رحمتی

تمرین چهارم

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

سوال ۱

a و b و c) با درنظرگرفتن مرکزهای اولیه، داده‌های هر خوشه عبارت است از:

$$\mu_1^0 = \begin{bmatrix}2.2\\2.8\end{bmatrix} \rightarrow C_1 = \left\{\begin{bmatrix}2.2\\4.2\end{bmatrix}, \begin{bmatrix}3.8\\3.6\end{bmatrix}, \begin{bmatrix}4.2\\2.9\end{bmatrix}, \begin{bmatrix}4.0\\2.3\end{bmatrix}\right\}$$

$$\mu_2^0 = \begin{bmatrix}2.0\\2.0\end{bmatrix} \rightarrow C_2 = \left\{\begin{bmatrix}3.4\\1.8\end{bmatrix}, \begin{bmatrix}1.6\\1.4\end{bmatrix}, \begin{bmatrix}2.4\\1.0\end{bmatrix}, \begin{bmatrix}1.5\\0.8\end{bmatrix}, \begin{bmatrix}2.8\\0.2\end{bmatrix}\right\}$$

$$\mu_3^0 = \begin{bmatrix}1.2\\3.0\end{bmatrix} \rightarrow C_3 = \left\{\begin{bmatrix}0.4\\4.5\end{bmatrix}\right\}$$

بر همین مبنا مرکز جدید سه خوشه تغییر خواهد کرد:

$$\mu_1^1 = \begin{bmatrix}3.55\\3.25\end{bmatrix}$$

$$\mu_2^1 = \begin{bmatrix}2.34\\1.04\end{bmatrix}$$

$$\mu_3^1 = \begin{bmatrix}0.4\\4.5\end{bmatrix}$$

d) باتوجه به مراکز جدید خوشه‌ها، داده‌های خوشه‌ها در دومین گام عبارت است از:

$$\mu_1^1 = \begin{bmatrix}3.55\\3.25\end{bmatrix} \rightarrow C_1 = \left\{\begin{bmatrix}2.2\\4.2\end{bmatrix}, \begin{bmatrix}3.8\\3.6\end{bmatrix}, \begin{bmatrix}4.0\\2.3\end{bmatrix}, \begin{bmatrix}4.2\\2.9\end{bmatrix}\right\}$$

$$\mu_2^1 = \begin{bmatrix}2.34\\1.04\end{bmatrix} \rightarrow C_2 = \left\{\begin{bmatrix}1.5\\0.8\end{bmatrix}, \begin{bmatrix}1.6\\1.4\end{bmatrix}, \begin{bmatrix}2.4\\1.0\end{bmatrix}, \begin{bmatrix}2.8\\0.2\end{bmatrix}, \begin{bmatrix}3.4\\1.8\end{bmatrix}\right\}$$

$$\mu_3^1 = \begin{bmatrix}0.4\\4.5\end{bmatrix} \rightarrow C_3 = \left\{\begin{bmatrix}0.4\\4.5\end{bmatrix}\right\}$$

باتوجه به اینکه جای داده‌ها تغییر نکرده است پس مراکز خوشه‌ها هم تغییری نخواهد کرد و لذا الگوریتم در پایان گام دوم خاتمه می‌یابد.

e) حالت بهینه سراسری زمانی رخ می‌دهد که یک خوشه دارای داده‌ی ۳- و ۱ باشد و خوشه دیگر دارای داده ۸. اگر شرایط به گونه‌ای پیش برود که یک خوشه بدون داده بماند و یا اینکه یک خوشه دارای ۸ و ۱ و دیگری دارای ۳- باشد به بهینه سراسری دست پیدا نخواهیم کرد. لذا می‌توان این شرایط را در نظر گرفت:

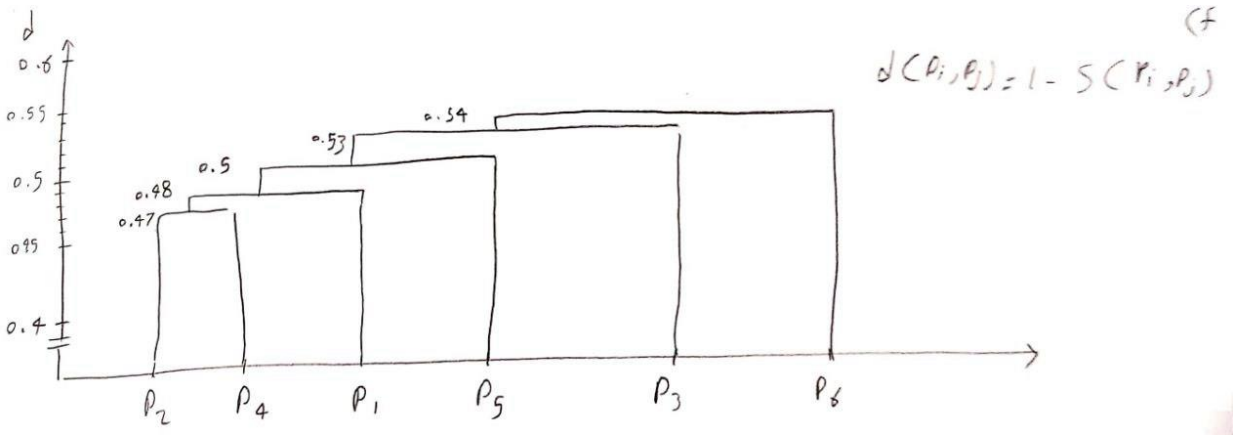
$$\begin{cases} |\mu_1 - 1| < |\mu_2 - 1| \\ |\mu_1 - 8| > |\mu_2 - 8| \\ \mu_1 < \mu_2 \end{cases}$$

با اشتراک‌گیری از شرایط بالا به بازه‌های زیر می‌رسیم:

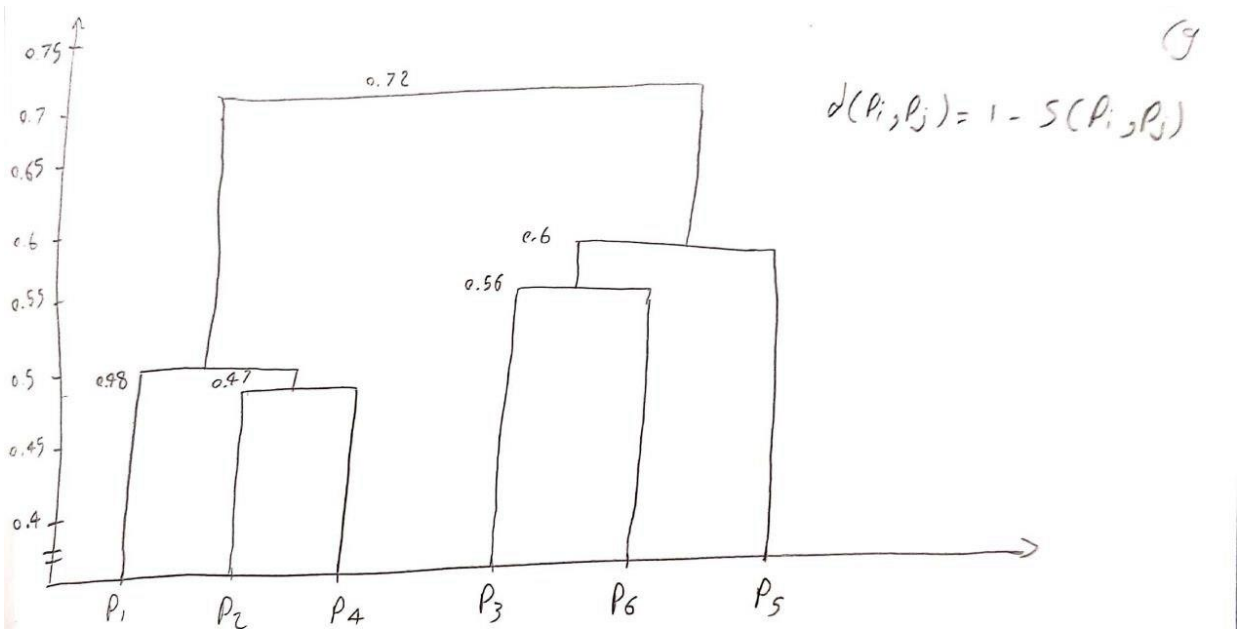
$$1 \leq \mu_1 < \mu_2 \leq 8$$

$$\mu_1 \leq 1 \;\&\; 1 - \mu_1 < \mu_2 - 1 \rightarrow \mu_1 \leq 1 \;\&\; 2 \leq \mu_1 + \mu_2$$

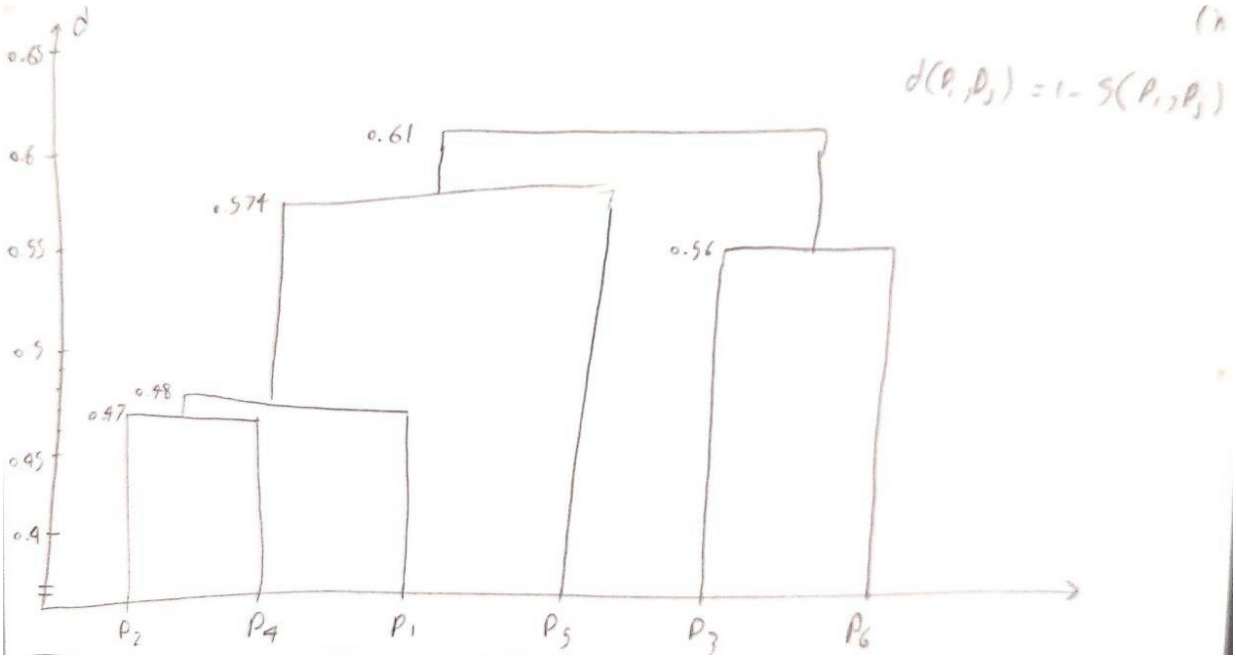
f)



(g


0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.72
0.6
0.56
0.98
0.47
P1
P2
P4
P3
P6
P5

(h


0.65
0.6
0.55
0.5
0.45
0.4
0.61
0.574
0.56
0.98
0.97
P2
P4
P1
P5
P3
P6

سوال ۲

b) در پیاده‌سازی من به همواره به دقت ۸۵/۴۱٪ می‌رسم. این نشان می‌دهد که در این مثال خاص احتمالا تعیین تصادفی مرکز نقش مهمی در خروجی الگوریتم ندارد.

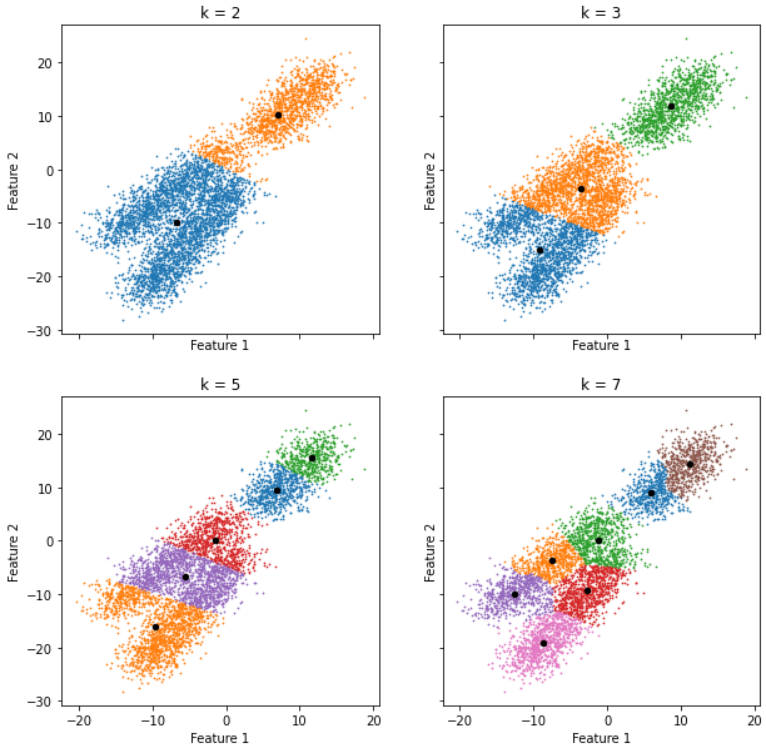
c) در این حالت هم دقت ۸۵/۴۱٪ میشود.

d) به صورت عادی این مراکز، مراکزی هستند که همگرا شده‌اند و اگر الگوریتم خوشه‌بندی با این مراکز شروع شود حد آستانه همگرایی شکسته می‌شود و در همان گام اول الگوریتم به پایان می‌رسد. یک احتمال دیگر هم که وجود دارد آن است که تعداد گام درنظر گرفته شده برای خوشه‌بندی اول کافی نباشد. در این صورت ممکن است در دفعه دوم خوشه‌بندی مراکز کمی تغییر کند.

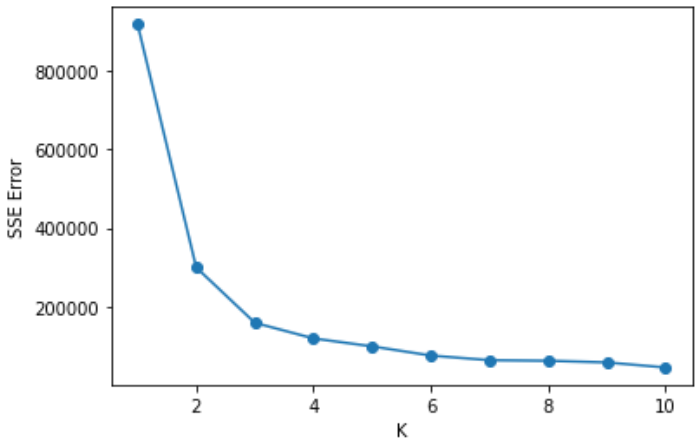
e) اگر داده‌های بابرچسب داشته باشیم، استفاده از یک الگوریتم نظارت‌شده معمولا به نتایج بهتری منجر می‌شود؛ چراکه در این حالت دانش بیشتری وجود دارد که یک الگوریتم نظارت‌شده از آن استفاده می‌کند ولی یک الگوریتم بدون نظارت آن را نادیده می‌گیرد. در مقایسه یک الگوریتم غیرنظارت‌شده دیگر با k-means هم به قطعیت نمی‌توان نظری داد و بسته به نوع داده‌ها و پارامترهای هر الگوریتم ممکن است k-means بهتر باشد و ممکن است الگوریتم دیگر عملکرد بهتری داشته باشد.

سوال ۳

(a



(b

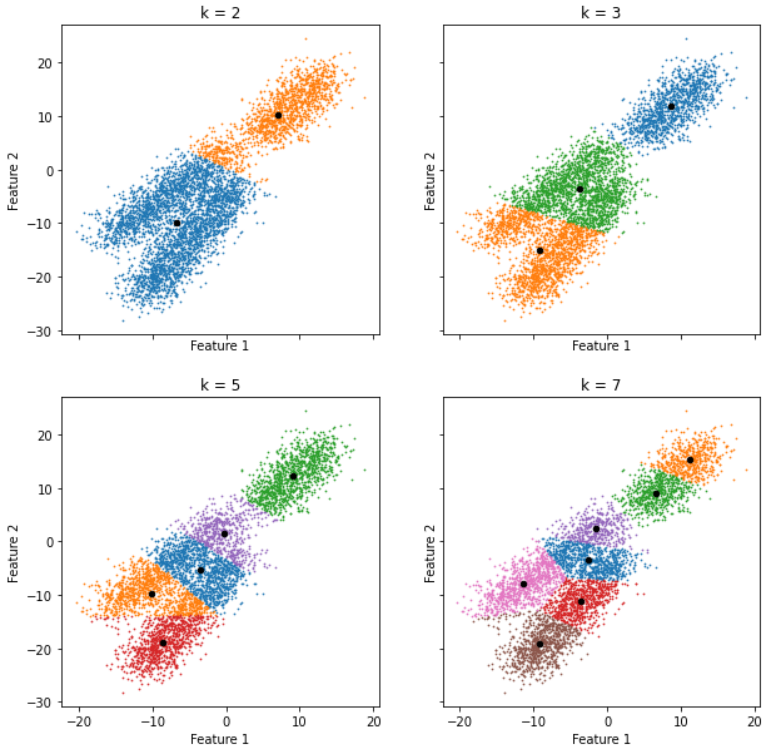


در این نمودار انتظار می‌رود که با افزایش مقدار k خطای خوشه‌بندی هم کاهش یابد اما مقدار کاهش روند ثابتی را ندارد؛ در ابتدا کاهش شدید است تا به نقطه‌ای برسیم که تعداد خوشه‌های متناسب با داده‌ها بدست آید و از آنجا به بعد کاهش کمتری رخ

خواهد داد. لذا یک شکستگی در نمودار دیده می‌شود که تعداد خوشه مناسب را نمایان می‌کند. در این نمودار هم ۴ خوشه به نظر مناسب می‌آیند. چراکه تقریبا بعد از آن شیب خط ثابت می‌شود.

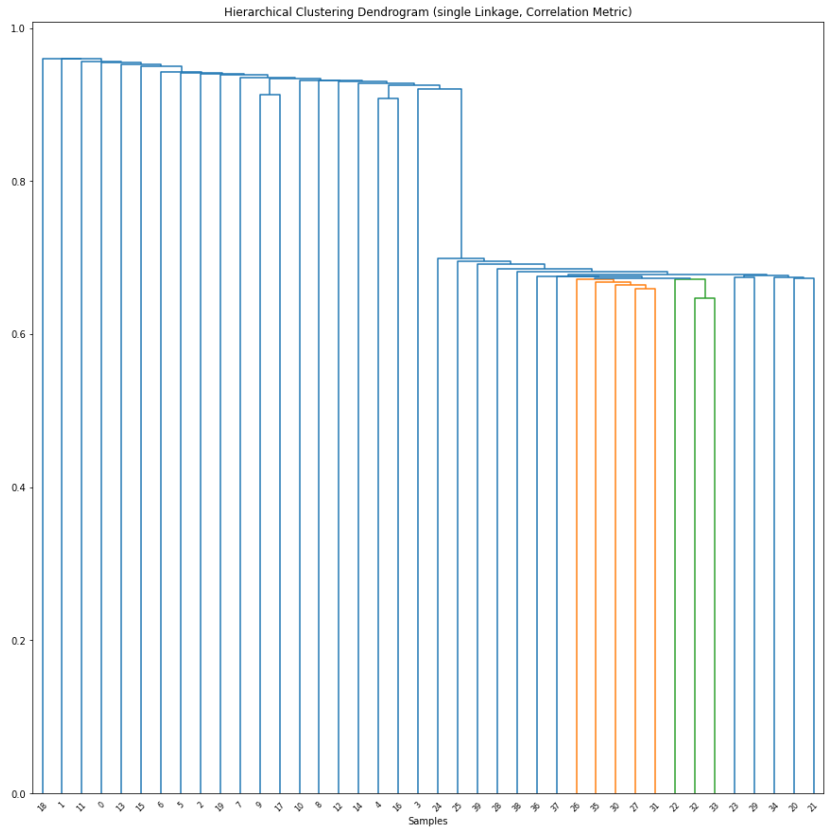
c) باتوجه به کندبودن پیاده‌سازی من، برای اجرای الگوریتم ۵ درصد داده‌ها را درنظر گرفتم و سپس از روی مراکز خوشه سایر داده‌ها را تعیین کردم که نمودار زیر حاصل شد:

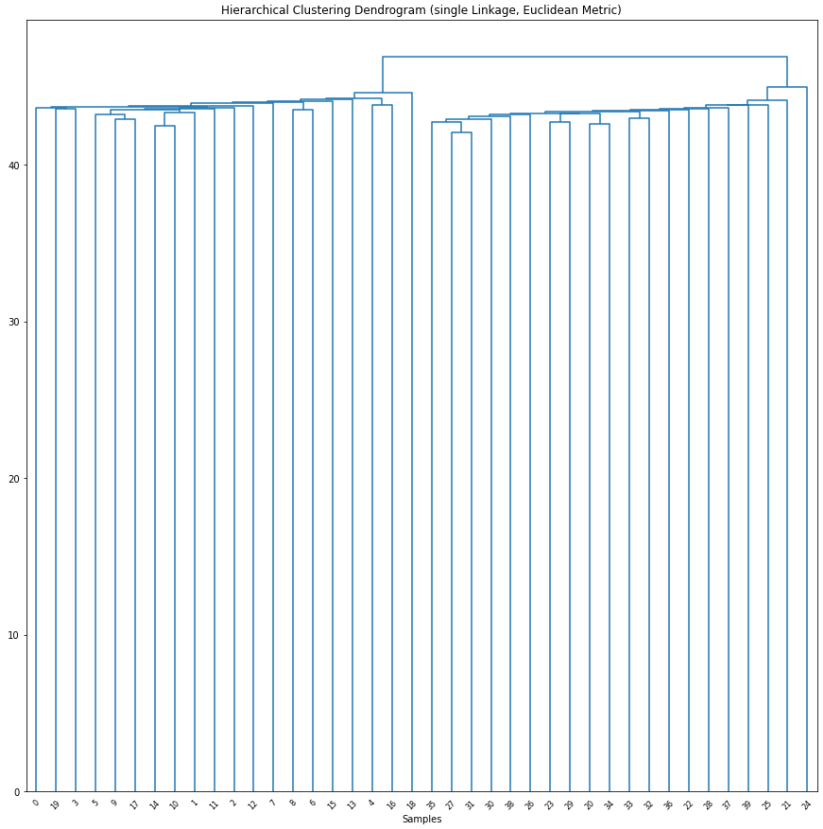
تفاوت خاصی با قسمت a مشاهده نمی‌شود؛ تنها برای قسمت k=5 می‌توان گفت نتیجه k-medoids بهتر است و برای k=7 نتیجه k-means. البته احتمالا نقاط اولیه متفاوت می‌توانست باعث تغییر وضعیت برای شود.

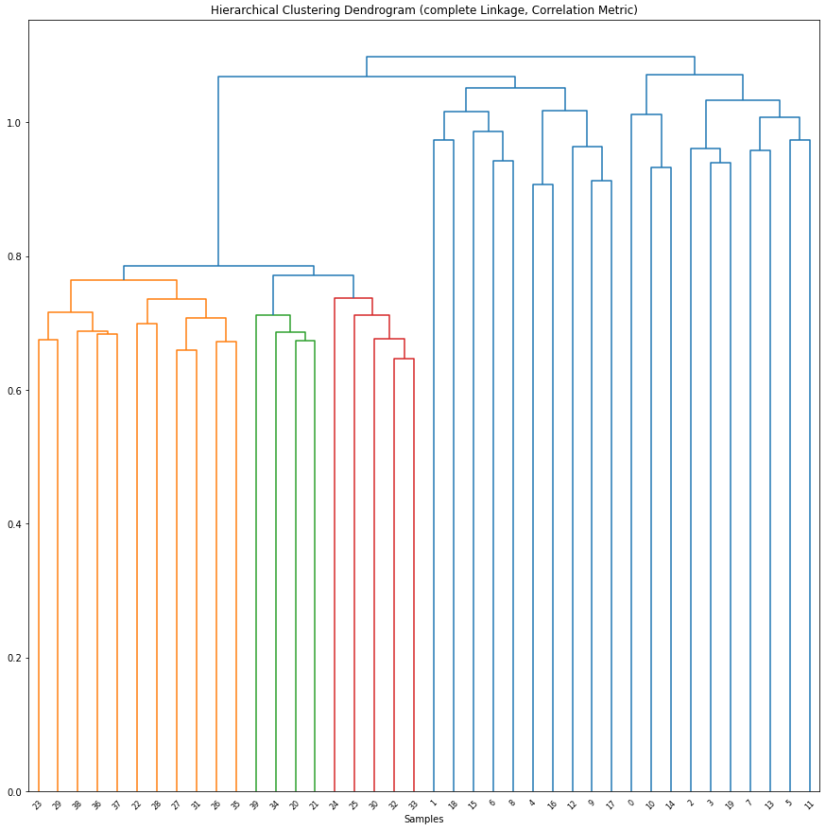


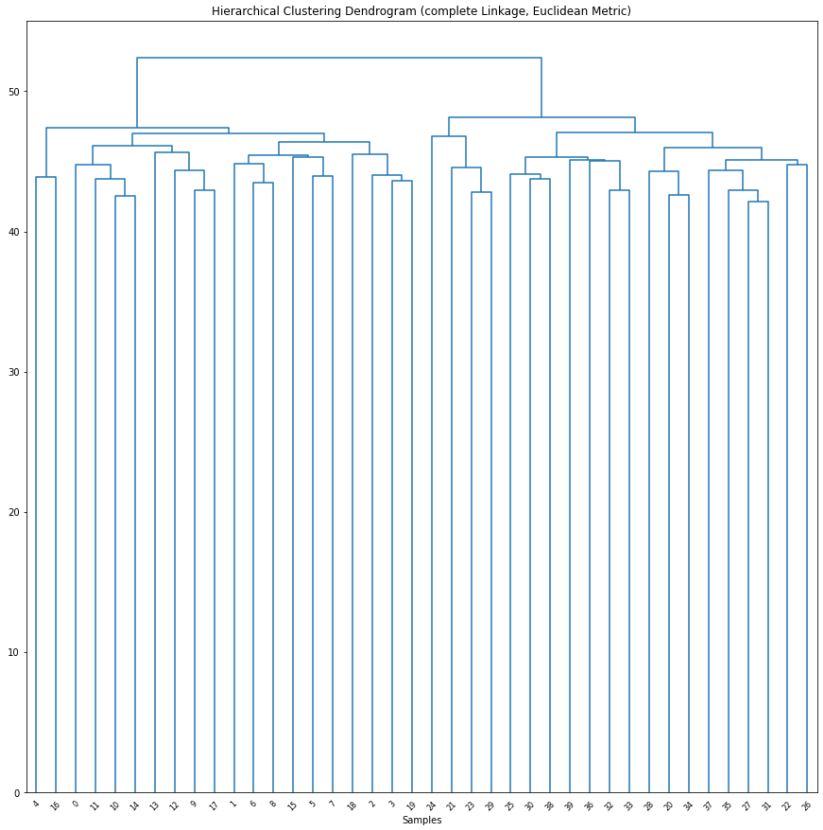
d) من برای خوشه‌بندی از کتابخانه‌ها استفاده کردم. استفاده از معیارهای شباهت متفاوت و همچنین رویکردهای متفاوت linkage متفاوت منجر به خروجی‌های متفاوت می‌شود. به طور دقیق‌تر برای معیار شباهت اقلیدوسی داده‌ها به درستی به دو خوشه شکسته می‌شود. برای معیار شباهت correlation وقتی که از single link استفاده کنیم یک داده در یک خوشه و مابقی در خوشه دیگر قرار می‌گیرند و وقتی که complete

link استفاده کنیم داده‌ها به دو خوشه شکسته می‌شود ولی حدود نیمی از داده‌های کلاس سالم در کنار داده‌های کلاس بیمار قرار می‌گیرد. در ادامه نمودار دندوگرام مربوط به این چهار حالت را می‌توانید مشاهده کنید:




Hierarchical Clustering Dendrogram (single Linkage, Euclidean Metric)
0
10
20
30
40
Samples


Hierarchical Clustering Dendrogram (complete Linkage, Correlation Metric)
0.0
0.2
0.4
0.6
0.8
1.0
Samples



e) یک راه ساده آن است که ابتدا داده را نرمال کنیم (مثلا میانگین را صفر و واریانس داده را یک کنیم.) سپس داده‌ها را به دو کلاس تقسیم کنیم و میانگین داده‌ها را حساب کنیم. ژن‌هایی که در دو میانگین بیشترین اختلاف را داشته باشند ژن‌هایی هستند که داده‌های دو کلاس را متفاوت می‌کند. مثلا برای ژن ۵۰۱ میانگین دو کلاس نزدیک به ۱/۷ احتلاف دارند. بعد از ژن ۵۰۱ ژن‌های ۵۸۸، ۵۹۹، ۵۸۹ و ۵۶۴ به ترتیب بیشترین اختلاف را دارند. البته توجه کنید که اگر قصدمان انتخاب یک مجموعه پنج تایی از بهترین ویژگی‌ها باشد این راه مناسب نیست و این راه ویژگی‌ها را به تنهایی بررسی می‌کند. جالب است بدانید باتوجه به ژن ۵۰۱ تمام داده‌های کلاس سالم به جز یک داده دارای مقدار کمتر از صفر و تمام داده‌های کلاس بیمار به جز یک داده دارای مقدار بالای صفر خواهد بود که دقت مناسبی را باتوجه به داشتن تنها یک ویژگی ممکن می‌سازد.

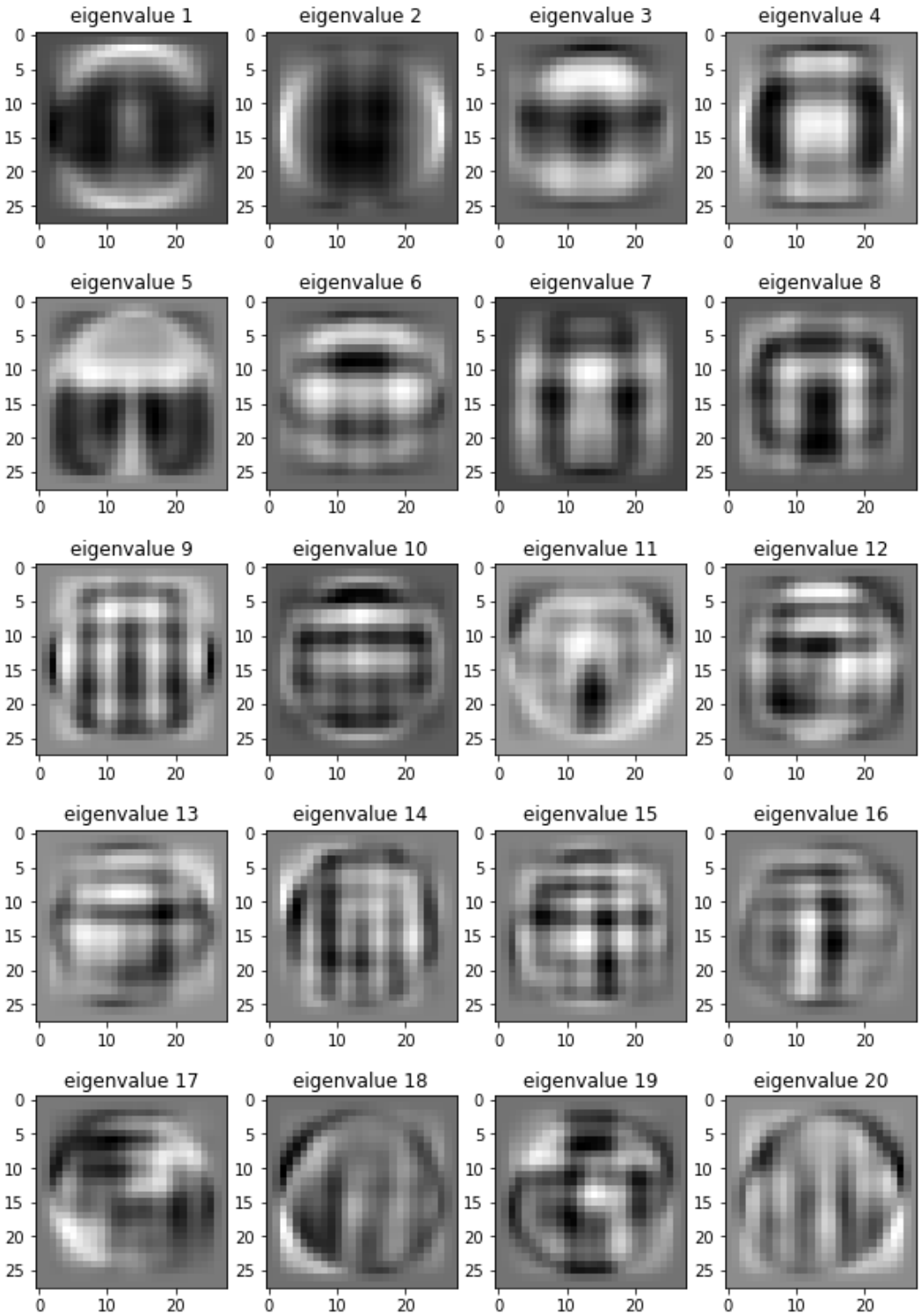
یک راه دیگر هم استفاده از LDA است. با استفاده از LDA می‌توانیم مناسب‌ترین جهت برای جداکردن دو کلاس را پیدا کنیم (فقط بعد اول). در این جهت مشخص است که کدام ویژگی‌ها به همراه یکدیگر می‌توانند بیشترین تاثیر را بگذارند (اعداد متناسب با ویژگی با مقادیر بالا) و کدام کمتر(اعداد متناسب با ویژگی نزدیک به صفر)

f) در این سوال باتوجه به جنس داده ورودی از single link برای الگوریتم سلسله مراتبی استفاده کردم. نتایج دو الگوریتم برای دو معیار در جدول زیر آمده است. بدیهی است که الگوریتم سلسله مراتبی به خوبی توانسته است داده‌ها را خوشه‌بندی کند ولی الگوریتم kmeans اصلا!

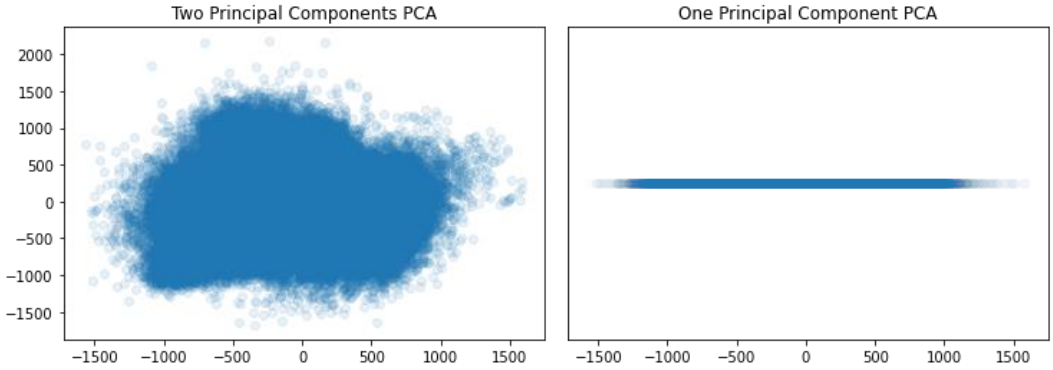
	Purity	NMI
K-means	۰/۵۹	۰/۰۲۳
Single-linkage	۱	۱

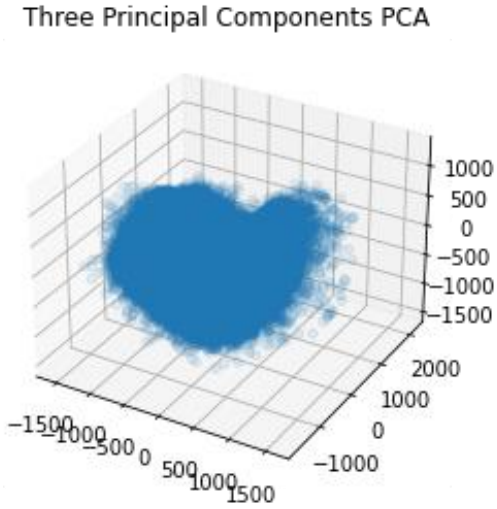
سوال ۴

a) بردارهای مربوط به مهم‌ترین مولفه‌ها عبارت‌اند از:

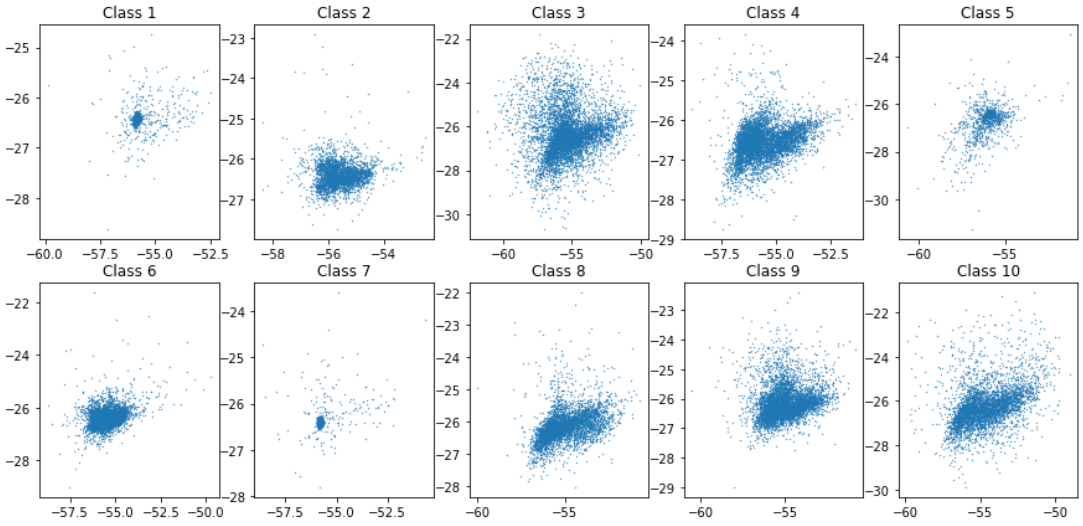


خروجی PCA به ازای یک، دو و سه مولفه اصلی:

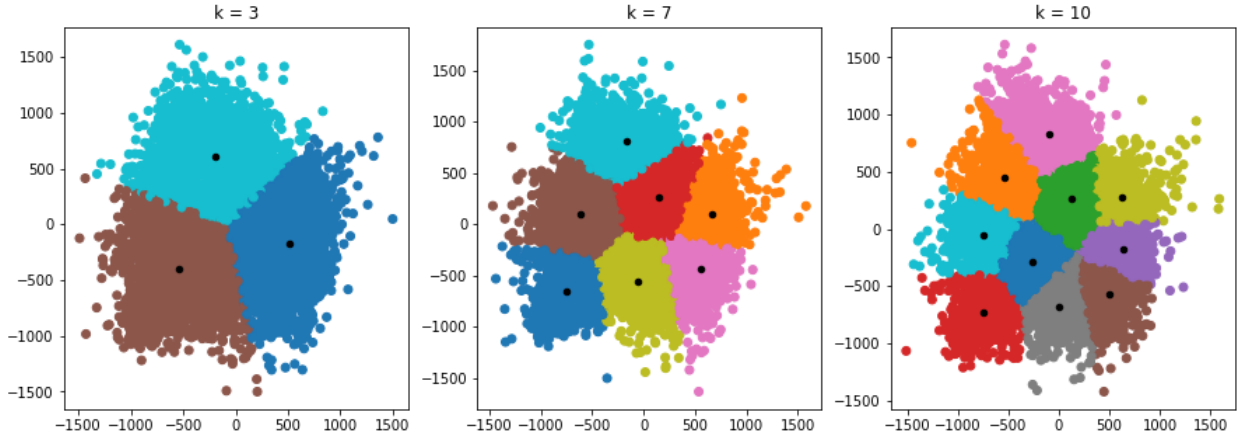




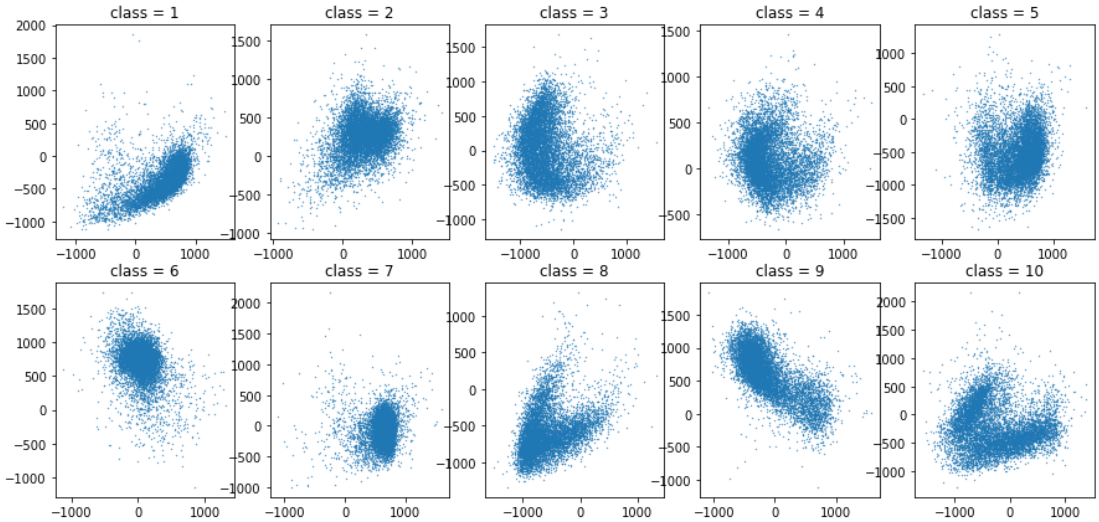
b) باتوجه به اینکه LDA به برچسب کلاس هم توجه می‌کند، در نمودار زیر هر کلاس در یک زیرنمودار جداگانه آورده شده است:



c) قبل از هرچیز توجه کنید که برای قسمت‌های مربوط به خوشه‌بندی در این سوال و برای اجرای سریع‌تر تنها ۱۰ درصد داده‌های آموزش درنظر گرفته شده است. چراکه همین تعداد هم برای کسب نتیجه کافی است. نتایج خوشه‌بندی به شرح زیر است:

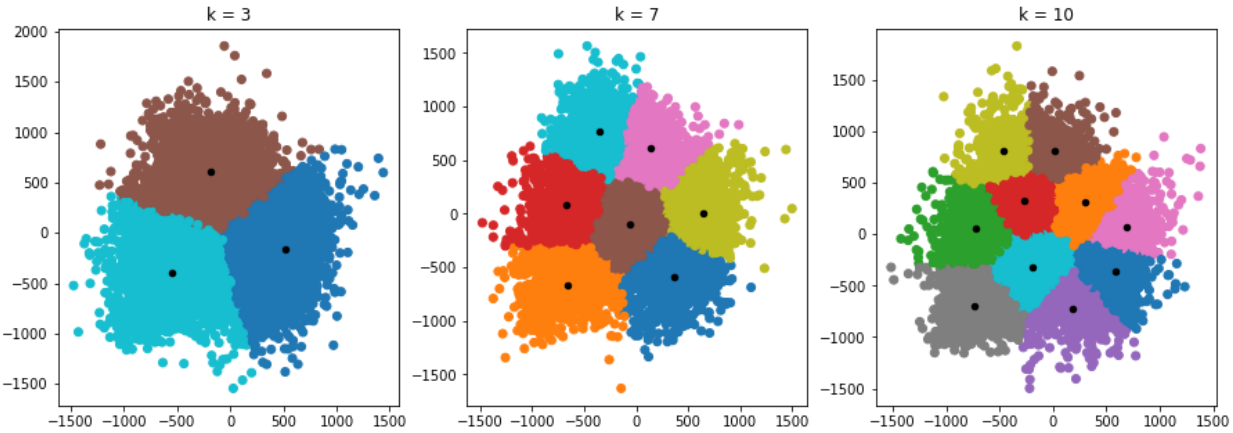


اگرچه خروجی PCA ذاتا دارای اجزای جدا از هم نبوده ولی خوشه‌بندی روی خروجی بدست آمده از PCA، خوشه‌بندی معقول است و فضا به به خوشه‌های تقریبا مساوی شکسته شده است. اما یک نکته‌ای که باید به آن توجه کرد این است که از آنجا که PCA توجهی به کلاس‌های هر داده نداشته است، لزومی هم وجود ندارد که هر خوشه بتواند نماینده یک یا چند کلاس مشخص باشد و حتی این امکان وجود دارد که PCA تفاوتی که در کلاس‌ها وجود داشته است را از بین برده باشد. لذا برای تحلیل بهتر نمونه‌های کلاس‌های مختلف را ترسیم می‌کنیم:



اگرچه برخی از کلاس‌ها تا حدی از سایر کلاس‌ها جدا شده است ولی نسبت به حالت LDA داده‌های کلاس‌ها بیشتر در هم فرو رفته‌اند و نتایج خوشه‌بندی برای جداکردن داده‌ها مطلوب به نظر نمی‌رسد.

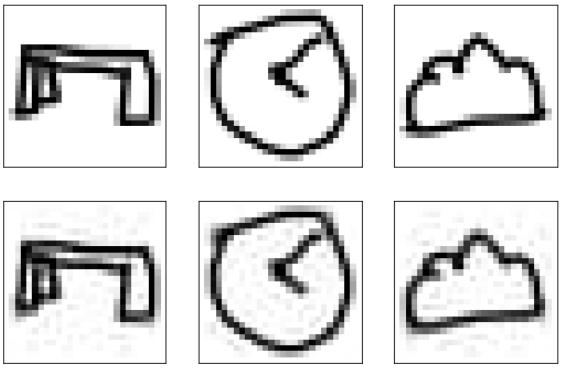
d) نتایج زیر حاصل می‌شود:



نتایج این قسمت با قسمت قبل چندان تفاوت ندارد و این بار هم یک خوشه نمی‌تواند نماینده داده‌های یک کلاس باشد. علت اینکه نقطه شروع متفاوت در اینجا کم تاثیر است احتمالا بدلیل آن است که داده‌ها خروجی PCA هستند و در یک فضا به صورت نسبتا یک دست پخش شده‌اند.

e) اهمیت و واریانس داده‌ها برای هر مولفه متناسب با مقدار ویژه متناسب با آن مولفه است. چنانچه مجموعه‌ای از k مولفه داشته باشیم. حاصل تقسیم مجموع مقادیر ویژه آن‌ها بر مجموع مقادیر ویژه کل مولفه‌ها نشان می‌دهد که چه میزان از واریانس حفظ شده است. با بررسی مقادیر ویژه دریافتم که باید حداقل ۲۷۱ مولفه اول را حفظ کرد تا این مقدار از واریانس باقی بماند.

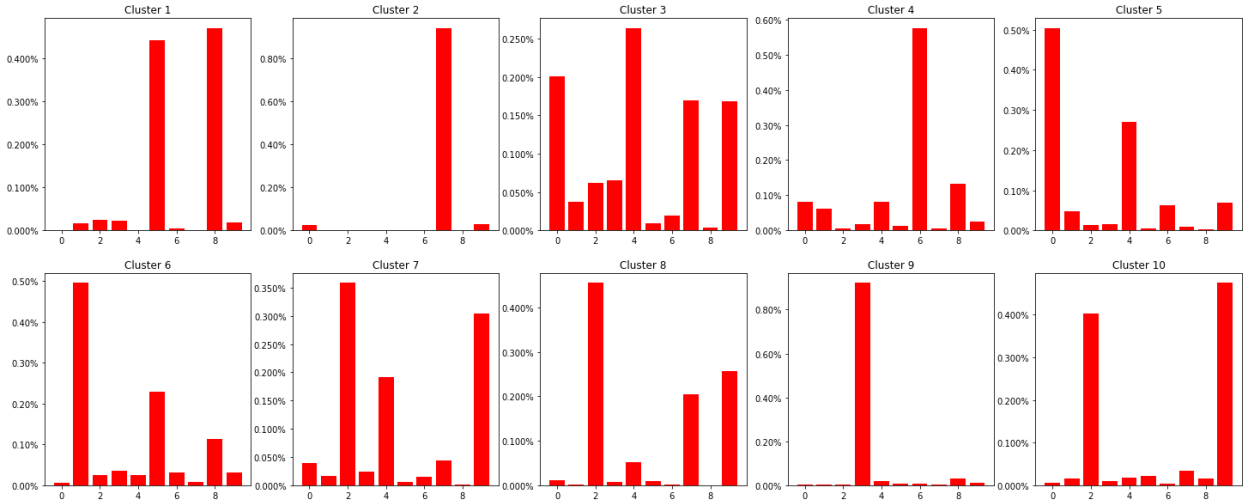
در تصویر زیر سه نمونه اصلی (سطر بالا) به همراه خروجی حاصل شده از ۲۷۱ مولفه اول (سطر پایین) را می‌توانید مشاهده کنید. همانطور که مشخص است خروجی دو تصویر شباهت بسیار بالایی به یکدیگر دارند و تنها تعدادی از پیکسل‌های پس‌زمینه دارای نویز شده است.



f) برای خوشه‌بندی داده‌ها را در فضای ۲۷۱ مولفه‌ای برده‌ام. نمونه‌های تصادفی زیر بدست آمدند (هر سطر مربوط به نمونه‌های یک خوشه هستند):

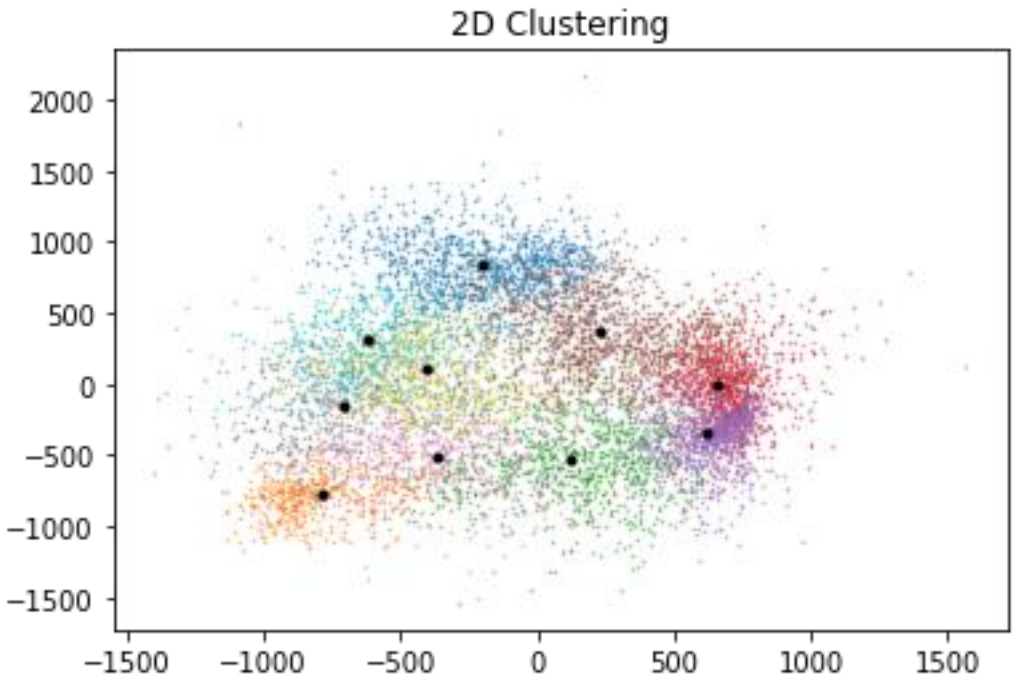
با بررسی نتایج فوق به نظر می‌رسد برخی از خوشه‌ها مانند خوشه مربوط به ابرها دارای خلوص مناسبی هستند ولی برخی دیگر مانند اولین خوشه مربوط به چند کلاس شده‌اند.

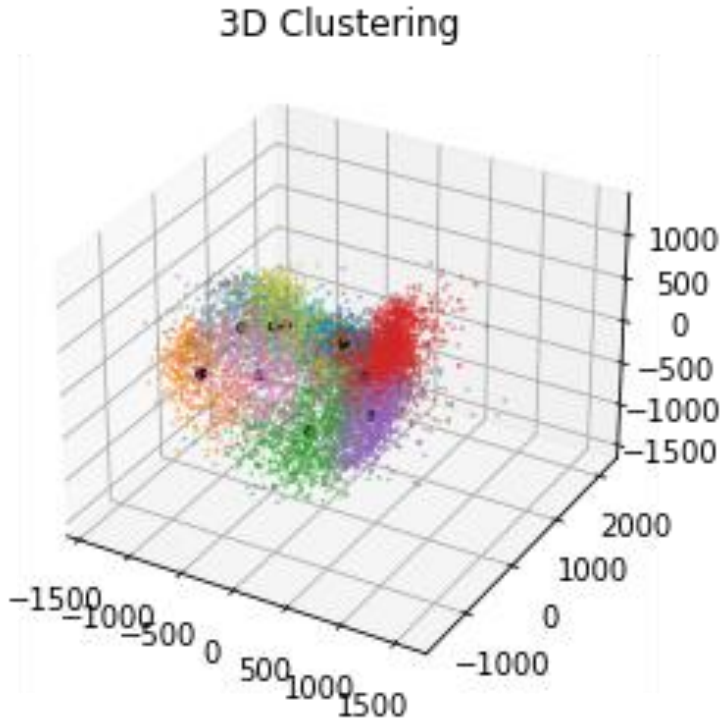
g) نتایج زیر حاصل شد:



برخی از خوشه‌ها (۲ و ۹) دارای خلوص بسیار بالایی هستند. برخی از خوشه‌ها (۱ و ۱۰) متعلق به دو کلاس هستند و برخی از خوشه‌ها (مانند ۳) عملا بین چندین کلاس پخش شده‌اند.

h) برای خوشه‌بندی دو و سه مولفه اول را حفظ می‌کنیم. برای حالت دوبعدی عملا در قسمت c برای حالت ده خوشه‌ای نمودار ترسیم شده است ولی مجدد برای این قسمت نیز ترسیم می‌شود:


2D Clustering
2000
1500
1000
500
0
−500
−1000
−1500
−1500
−1000
−500
0
500
1000
1500


3D Clustering
1000
500
0
−500
−1000
−1500
2000
1000
0
−1000
−1500
−1000
−500
0
500
1000
1500

سوال ۵

a) در مسائل خوشه‌بندی به عنوان یک دسته از مسائل بدون نظارت، عملا باتوجه به رابطه بین داده‌ها می‌توان خوشه‌بندی را انجام داد؛ یعنی داده‌هایی که با یکدیگر فاصله کمی دارند ولی با سایر داده‌ها فاصله زیادی دارند در یک خوشه قرار می‌گیرند. در این حالت می‌توان نقش پررنگ نمونه‌ها در خوشه‌بندی را مشاهده کرد. در برخی از الگوریتم‌ها برای آنکه یک داده جدید خوشه‌اش مشخص شود باید دید به کدام داده نزدیک است و به آن خوشه برود. این چیزی است که در الگوریتمی مانند k-NN هم دیده می‌شود. برای تعیین برچسب یک داده در K-NN هم به سراغ کلاس داده‌های آموزشی می‌رویم.

b) الگوریتم k-means هیچ راهکاری برای حذف داده‌های نویز ندارد و لذا هر داده نویز باید در یک خوشه قرار گیرد. در k-means برای تعیین مرکز خوشه از میانگین تمام داده‌های خوشه استفاده شود. طبیعی است که در این شرایط یک داه پرت می‌تواند مرکز خوشه را به جای نامناسبی بکشاند. مشکل دیگری که شاید پیش بیاید بالا رفتن معیار خطا مانند SSE است و بدین ترتیب این احتمال وجود دارد که به طور اشتباه تعداد خوشه‌ها را زیادتر از چیزی که باید باشد پیشنهاد دهیم.

c) خیر امکان ندارد؛ هر وضعیت در k-means دارای یک خطای SSE است. در الگوریتم k-means دو گام وجود دارد. یک گام بروز کردن میانگین است. در این گام مقدار SSE کم‌تر یا مساوی می‌شود. گام دیگر جابجایی داده‌ها به خوشه‌ای است که به مرکز آن نزدیک‌تر است. طبیعتا در این حالت هم مقدار SSE کم‌تر یا مساوی می‌شود. در نتیجه در طول الگوریتم k-means امکان ندارد مقدار SSE زیادتر شود. پس اگر یک وضعیت دوباره مشاهده شود به این معناست که مقدار SSE تغییری نکرده است. همچنین امکان ندارد که از یک وضعیت با یک مقدار SSE به وضعیت دیگر با همان مقدار SSE برویم چون در صورتی SSE ثابت می‌ماند که وضعیت عوض نشود. پس امکان ندارد یک وضعیت را دوباره ببینیم.

با این شرایط اگر وضعیت ثابت بماند که به وضوح به همگرایی رسیده‌ایم ولی اگر قرار باشد وضعیت تغییر کند یعنی SSE دائما در حال کاهش است و یک کمینه مطلق یعنی

۰ وجود دارد. طبیعتا در این شرایط امکان ندارد تا ابد SSE کاهش یابد پس یکجایی به همگرایی خواهیم رسید.

d) امکان ندارد تعداد خوشه‌ها بیشتر از k شود! ما همواره k مرکز داریم و هر داده به یکی از این k مرکز نزدیک‌تر است. پس نمی‌توانی حالتی را یافت که تعداد خوشه‌ها بیشتر شود.

امکان دارد تعداد خوشه‌ها کم‌تر از k شود. اگر n<k باشد قطعا خوشه‌ای وجود خواهد داشت که هیچ داده‌ای ندارد و لذا با حذف خوشه‌های خالی تعداد خوشه بازگشتی کمتر از k خواهد شد. این تنها حالتی نیست که تعداد خوشه‌ها کمتر از k شده است. وقتی چند داده دقیقا در یک محل قرار می‌گیرند حتما در یک خوشه هم خواهند بود. پس اگر تعداد محل‌های یکتا کمتر از k باشد فارغ از آنکه تعداد کل داده بیشتر از k یا خیر باز خوشه‌های غیرخالی بیشتر از یک می‌شود. بسته به پیاده‌سازی یک احتمال دیگر هم برای تعداد خوشه کمتر از k وجود دارد؛ اگر در مقداردهی اولیه به خوشه‌ها برخی از مراکز دقیقا روی هم بیافتد و در پیاده‌سازی زمانی که یک داده به چند مرکز به یک میزان نزدیک است، داده به صورت غیرتصادفی به یک خوشه تخصیص داده شود (مثلا خوشه با آیدی پایین‌تر) و اگر دو مرکز در یک محل باشند و در ابتدا در وضعیت همگرایی قرار داشته باشیم، طبیعتا داده‌هایی که به این دو مرکز نزدیک هستند تنها به یکی از این مرکزها تعلق خواهد گرفت و یک مرکز بدون داده خواهد ماند!

e) الگوریتم k-means نگاه دایره‌ای به خوشه‌ها دارد درحالی که الگوریتم‌های سلسله‌مراتبی می‌توانند متفاوت باشد. موارد برتری دو الگوریتم نسبت به یکدیگر می‌تواند خیلی زیاد باشد که در اینجا چند حالت معروف بررسی می‌شود. در شرایطی که خوشه‌های واقعی کشیده هستند یک الگوریتم سلسله‌مراتبی single link می‌تواند این خوشه‌ها را تشخیص دهد ولی k-means ممکن است موفق نشود. به عنوان مثالی دیگر اگر خوشه‌ها به شکل دو حلقه باشند که یکی در دیگری قرار دارد، باز الگوریتم single link می‌تواند آن را حل کند ولی k-means خیر.

در حالت عکس اگر داده‌ها دو خوشه دایره‌ای باشند که این دو دایره در کنار هم باشند و به هم چسبیده باشند، الگوریتم k-means می‌تواند هر کدام را در یک خوشه قرار دهد

ولی الگوریتم single link شاید مرز اشتراک دو خوشه را از ابتدا در یک خوشه قرار دهد و به مشکل بخورد.

تا به اینجا راجع به دقت صحبت کردم. در الگوریتم k-means باید تعداد خوشه‌ها را از ابتدا بدانیم و اگرنه مجبور می‌شویم بارها و بارها الگوریتم را به ازای k های مختلف اجرا کنیم و ببینیم کدام یک مناسب‌تر است ولی یک الگوریتم سلسله‌مراتبی با یک باز اجرا شدن به ازای تعداد خوشه مختلف جواب را در خود خواهد داشت. به علاوه الگوریتم k-means دارای یک وضعیت اولیه تصادفی است و به ازای شروع‌های مختلف به جواب‌های مختلف ختم می‌شود. اما اگر مقدار k را بدانیم الگوریتم k-means می‌تواند سرعت خوبی را برای پردازش داده‌های حجیم داشته باشد. فارغ از محسبات نسبتا کم آن در مقایسه با سایر الگوریتم‌های خوشه‌بندی، می‌توان باتوجه به حجم داده‌ها تعداد گام کمتری درنظر گرفت که به هر حال ما را به یک جواب می‌رساند ولی در سلسله‌مراتبی چنین تنظیمی وجود ندارد.