**به نام خدا**

**داده‌کاوی**

**تمرین پنجم**

**خانم دکتر شاکری**

**محمد امین عرب خراسانی**

**۸۱۰۱۰۲۲۰۵**

**بهار ۱۴۰۳**

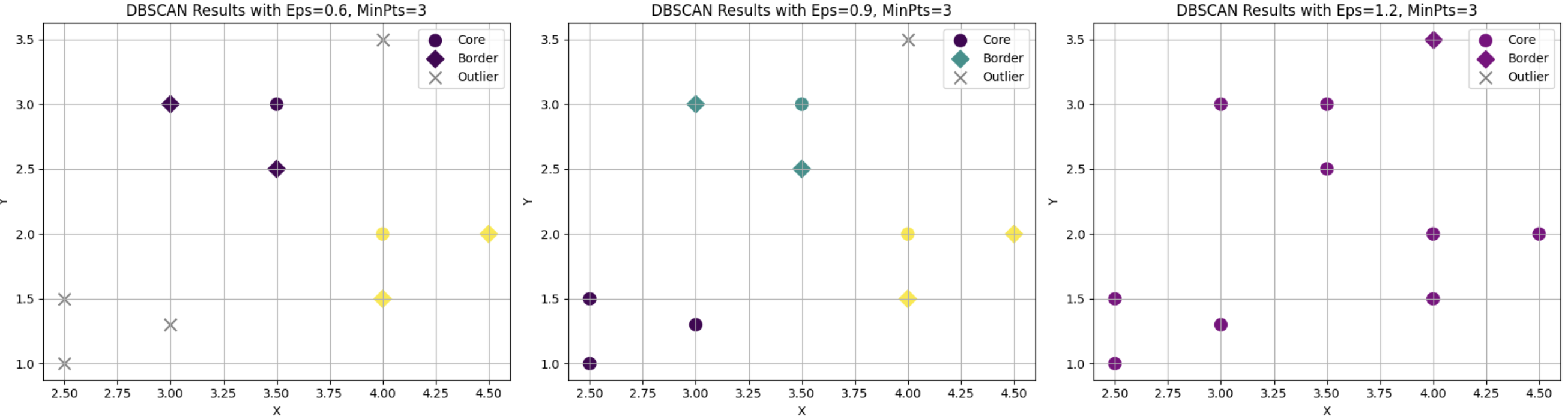
بخش تشریحی

سوال اول)

الف) برای اعمال الگوریتم DBSCAN با استفاده از فاصله‌ی جفت نقاط از یکدیگر بر اساس معیار منهتن، برای هر کدام از شرایط ذکر شده به این صورت است که اگر فاصله‌ای هر نقطه با نقاط دیگر از بیشترین شعاع همسایگی (Eps) داده شده در هر شرط کوچک‌تر باشد جزو نقاط با فاصله‌ی همسایگی مشخص شده در نظر گرفته می‌شود. اگر این تعداد نقاطی که این شرط را برقرار کنند بیشتر یا برابر با MinPts باشد، این نقطه به عنوان core در نظر گرفته می‌شود. همچنین در صورتی که نقطه‌ای که شرط Eps را ارضا می‌کند اما شرط MinPts برای آن برقرار نمی‌باشد به عنوان نقطه‌ی border در نظر گرفته خواهد شد. نقطه‌ای که نه border و نه core باشد نقطه‌ی outlier خواهد بود. جدول زیر نتیجه‌ی اعمال الگوریتم ذکر شده برای نقاط داده شده می‌باشد.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Eps = 1.2, MinPts = 3 | | Eps = 0.9, MinPts = 3 | | Eps = 0.6, MinPts = 3 | | Points |
| Core | P2, P3, P8 | Core | P2, P3 | Outlier | - | P1 |
| Core | P2, P3 | Core | P1, P3 | Outlier | P3 | P2 |
| Core | P1, P2 | Core | P1, P2 | Outlier | P2 | P3 |
| Core | P5, P6 | Border | P5 | Border | P5 | P4 |
| Core | P4, P6, P9 | Core | P4, P6 | Core | P4, P6 | P5 |
| Core | P4, P5, P10 | Border | P5 | Border | P5 | P6 |
| Core | P8, P10 | Border | P10 | Border | P10 | P7 |
| Core | P1, P7, P10 | Border | P10 | Border | P10 | P8 |
| Border | P5 | Outlier | - | Outlier | - | P9 |
| Core | P6, P7, P8 | Core | P7, P8 | Core | P7, P8 | P10 |

ب) همانطور که مشخص است، در هر سه شرط داده شده MinPts ثابت بوده و Eps افزایش یافته است. طبق نتایج حاصل از قسمت الف، با افزایش Eps تعداد نقاط core افزایش یافته است و این بدان معنا است که این به معنای گسترش خوشه در کنار کاهش نقاط outlier می‌باشد. این امر کمک می‌کند که نقاطی که به اندازه‌ی کافی به یکدیگر نزدیک هستند، در یک خوشه قرار بگیرند. از سمت دیگر ثابت بودن MinPts در این شرایط نشان می‌دهد که نقش اصلی در خوشه‌بندی را پارامتر Eps دارد. از طرفی مشخص است که با افزایش MinPts، تعداد نقاط core ممکن است کاهش یابد و کاهش نقاط core می‌تواند منجر به افزایش نقاط outlier شود. برای درک بهتر از تغییر پارامترها در پایتون کدی پیاده‌سازی شده است که برای هر کدام از شرایط ذکر شده، خوشه‌های ایجاد شده را با نقاط core، outlier و border نشان می‌دهد. برای این منظور مختصات نقاط با استفاده از شکل اول تمرین که نقاط را در مختصات کارتزین نشان می‌دهد استخراج شده است و برای چک کردن درست بودن مختصات جدول فاصله‌ی منهتن آن تشکیل شده تا درستی آن چک شود. در ادامه به کمک کتابخانه‌ی scikit-learn الگوریتم DBSCAN روی آن اعمال شده و نتایج حاصل از خوشه‌بندی در هر سه شرط مشخص می‌شود. شکل‌های زیر حاصل اجرای کد پیاده‌سازی شده می‌باشند.



همانطور که از شکل‌های بالا برمی‌آید، با کم بودن Eps ممکن است اطلاعات معناداری از داده‌ها از دست برود. نقاط P1، P2 و P3 نقاط معناداری هستند که به دلیل کم بودن Eps به عنوان outlier در نظر گرفته شده‌اند. از طرفی با زیاد بودن این پارامتر، ممکن است خوشه‌هایی که واقعا می‌بایست از هم جدا باشند در یک خوشه، خوشه‌بندی شوند که باعث می‌شود تاثیر الگوریتم و خوشه‌ها کمتر شود. همانطور که مشاهده می‌شود، تمامی نقاط در حالت سوم در یک خوشه دسته‌بندی شده‌اند.

ج) برای آن که این حالت مورد بررسی قرار گیرد، ابتدا باید در نظر داشت که نقطه‌ای که نسبت به نقطه‌ی دیگر directly density-reachable است، در شرط Eps صدق می‌کند و یکی از نقاط شرط MinPts را نیز دارد. این به آن معناست که یکی از نقاط باید core باشد. از طرفی طبق الگوریتم DBSCAN، نقاطی که به عنوان core در نظر گرفته می‌شوند با نقاطی که directly density-reachable هستند یک خوشه را تشکیل می‌دهند. در واقع، تمام نقاط directly density-reachable که توسط یک نقطه core متصل می‌شوند، بخشی از یک خوشه خواهند بود، یا اگر شرط MinPts را ارضا نکند حداقل نقطه‌ای که به آن می‌رسد، یک نقطه Border خواهد بود اما همچنان از طریق یک نقطه core به خوشه متصل است. بنابراین چنین حالتی امکان‌پذیر نمی‌باشد.

بله. برای بخش دوم این سوال، ابتدا باید در نظر داشت که نقاطی با یکدیگر density-connected هستند که به واسطه‌ی زنجیره‌ای از نقاط در یک خوشه قرار گرفته‌اند اما با یکدیگر directly density-reachable نمی‌باشند. همانطور که از شکل قسمت قبل مشخص است، نقاط بنفش همگی core هستند و در یک خوشه قرار گرفته‌اند که این شرط برای آن‌ها برقرار نمی‌باشد اما برای دو خوشه‌ی دیگر می‌توان نقاطی پیدا کرد که با یکدیگر density-connected هستند. به طور مثال، نقطه‌ی P5 به عنوان یک نقطه‌ی core و P4 با یکدیگر directly density-reachable می‌باشند. از طرفی این شرایط برای P5 و P6 نیز برقرار می‌باشد. اما نقاط P4 و P6، این شرط را ندارند. بنابراین نقاط P4 و P6 به واسطه‌ی P5 با یکدیگر density-connected می‌باشند. همین حالت برای نقاط P7 و P8 نیز برقرار است.

د) برای خوشه‌بندی دیتا به روش KMeans ابتدا فاصله‌ی تمامی نقاط از هر یک از سه نقطه‌ای که به عنوان centroid در نظر کرفته شده‌اند، محاسبه‌ می‌شود. در ادامه برای تعیین خوشه‌ی هر نقطه، بر اساس کمترین فاصله از هر centroid، خوشه‌ی نقطه مشخص می‌شود. جدول زیر این فواصل را برای هر نقطه نشان می‌دهد.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Points | Coordinates | D to P6 | D to P8 | D to P10 |
| P1 | (3, 1.3) | 1.7 | 1.2 | 1.7 |
| P2 | (2.5, 1) | 2.5 | 2 | 2.5 |
| P3 | (2.5, 1.5) | 2 | 1.5 | 2 |
| P4 | (3, 3) | 0.5 | 2.5 | 2 |
| P5 | (3.5, 3) | 0.5 | 2 | 1.5 |
| P6 | (3.5, 2.5) | 0 | 1.5 | 1 |
| P7 | (4.5, 2) | 1.5 | 1 | 0.5 |
| P8 | (4, 1.5) | 1.5 | 0 | 0.5 |
| P9 | (4, 3.5) | 1.5 | 2 | 1.5 |
| P10 | (4, 2) | 1 | 0.5 | 0 |

در این مرحله نقاط مربوط به هر خوشه مشخص می‌شود.

Cluster 1 (centroid: P6): P4, P5, P6, P9

Cluster 2 (centroid: P8): P1, P2, P3, P8

Cluster 3 (centroid: P10): P7, P10

از آنجایی که فاصله‌ی نقطه‌ی P9 از خوشه‌های اول و سوم به یک اندازه است می‌توانند در هر دو دسته قرار بگیرند. به دلیل نزدیک بودن بقیه‌ی نقاط خوشه‌ی ۱ به P9، این نقطه در خوشه‌ی ۱ قرار می‌گیرد.

برای پیدا کردن نقاط centroid جدید نیاز است به کمک نقاطی که در یک خوشه قرار دارند نقاط جدید پیدا شوند. برای پیدا کردن مختصات نقاط centroid جدید از میانگین نقاط هر خوشه استفاده شده است.

Cluster 1:

Centroid = mean(P4, P5, P6, P9) = (((4+3+3.5+3.5)/4) , ((3.5+3+3+2.5)/4)) = (3.5, 3)

Cluster 2:

Centroid = mean(P1, P2, P3, P8) = (((3+2.5+2.5+4)/4) , ((1.3+1+1.5+1.5)/4)) = (3, 1.325)

Cluster 3:

Centroid = mean(P7, P10) = (4.25, 2)

حال بر اساس centroidهای جدید فاصله‌ی هر نقطه تا این نقاط اندازه گرفته می‌شود. جدول زیر نشان دهنده‌ی فاصله‌های ذکر شده می‌باشد.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Points | Coordinates | D to (3.5, 3) | D to (3, 1.325) | D to (4.25, 2) |
| P1 | (3, 1.3) | 2.2 | 0.025 | 1.95 |
| P2 | (2.5, 1) | 3 | 0.825 | 2.25 |
| P3 | (2.5, 1.5) | 2.5 | 0.675 | 2.25 |
| P4 | (3, 3) | 0.5 | 1.675 | 2.25 |
| P5 | (3.5, 3) | 0 | 2.175 | 1.75 |
| P6 | (3.5, 2.5) | 0.5 | 1.675 | 1.25 |
| P7 | (4.5, 2) | 2 | 2.175 | 0.25 |
| P8 | (4, 1.5) | 2 | 1.175 | 0.75 |
| P9 | (4, 3.5) | 1 | 3.175 | 1.75 |
| P10 | (4, 2) | 1.5 | 1.675 | 0.25 |

همانطور که از جدول بالا برمی‌آید، فقط نقطه‌ی P8 به خوشه‌ی جدید assign شد. بنابراین یک مرحله‌ی دیگر این الگوریتم ادامه می‌یابد تا زمانی که همه‌ی نقاط در همان خوشه‌ی خود باقی بمانند.

Cluster 1:

Centroid = mean(P4, P5, P6, P9) = (((4+3+3.5+3.5)/4) , ((3.5+3+3+2.5)/4)) = (3.5, 3)

Cluster 2:

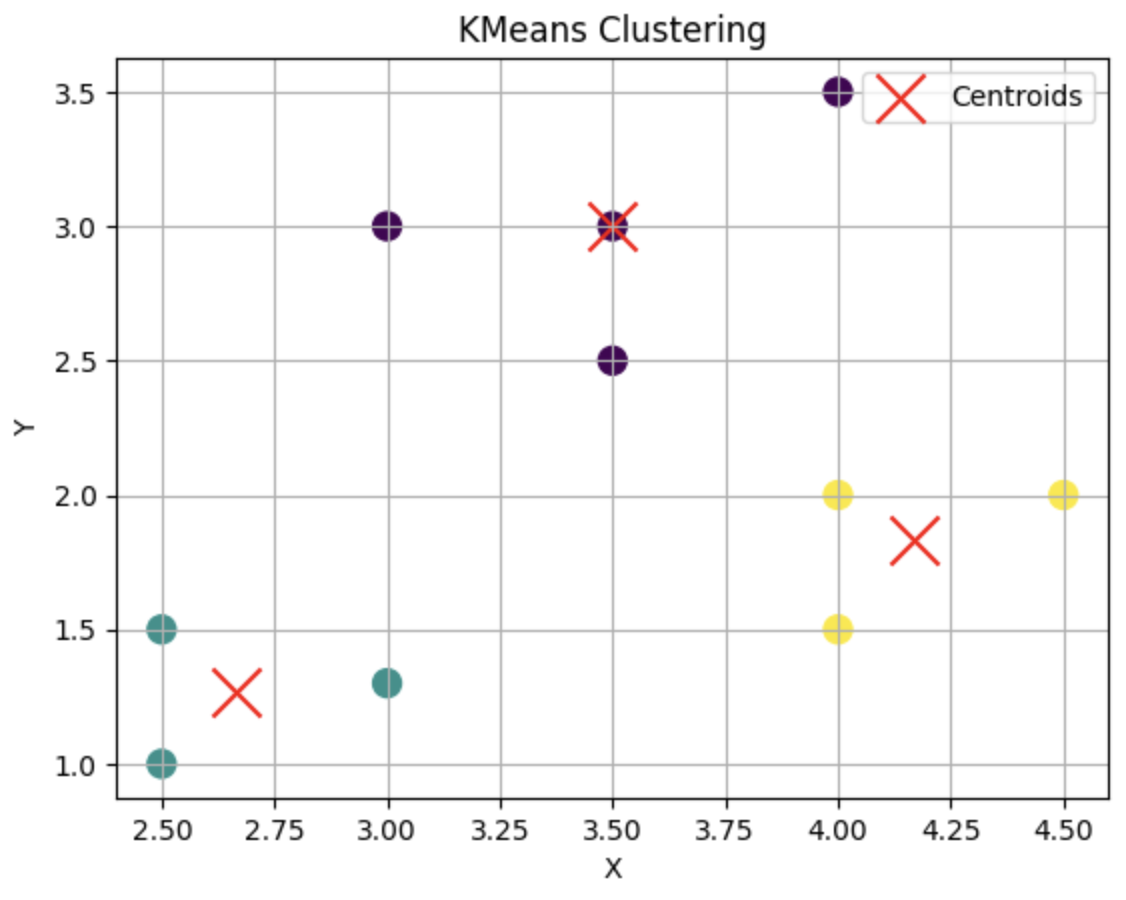
Centroid = mean(P1, P2, P3) = (((3+2.5+2.5)/3) , ((1.3+1+1.5)/3)) = (2.67, 1.27)

Cluster 3:

Centroid = mean(P7, P8, P10) = (((4.5, 4, 4)/3) , ((2, 1.5, 2)/3)) = (4.17, 1.83)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Points | Coordinates | D to (3.5, 3) | D to (2.7, 1.3) | D to (4.2, 1.8) |
| P1 | (3, 1.3) | 2.2 | 0.36 | 1.7 |
| P2 | (2.5, 1) | 3 | 0.44 | 2.5 |
| P3 | (2.5, 1.5) | 2.5 | 0.4 | 2 |
| P4 | (3, 3) | 0.5 | 2.06 | 2.34 |
| P5 | (3.5, 3) | 0 | 2.56 | 1.84 |
| P6 | (3.5, 2.5) | 0.5 | 2.06 | 1.34 |
| P7 | (4.5, 2) | 2 | 2.56 | 0.5 |
| P8 | (4, 1.5) | 2 | 1.56 | 0.5 |
| P9 | (4, 3.5) | 1 | 3.56 | 1.84 |
| P10 | (4, 2) | 1.5 | 2.06 | 0.4 |

طبق نتایج حاصل، خوشه‌ی هیج نقطه‌ای دچار تغییر نشده است بنابراین الگوریتم در این مرحله متوقف می‌شود. نقاط centroid نهایی به ترتیب برای خوشه‌های ۱ تا ۳، (۳ ،۳.۵)، (۱.۳ ،۲.۷) و (۱.۸ ،۴.۲) می‌باشند. همچنین به کمک کتابخانه‌ی scikit-learn این الگوریتم پیاده‌سازی شده و نتیجه‌ی آن در شکل زیر آورده شده است.



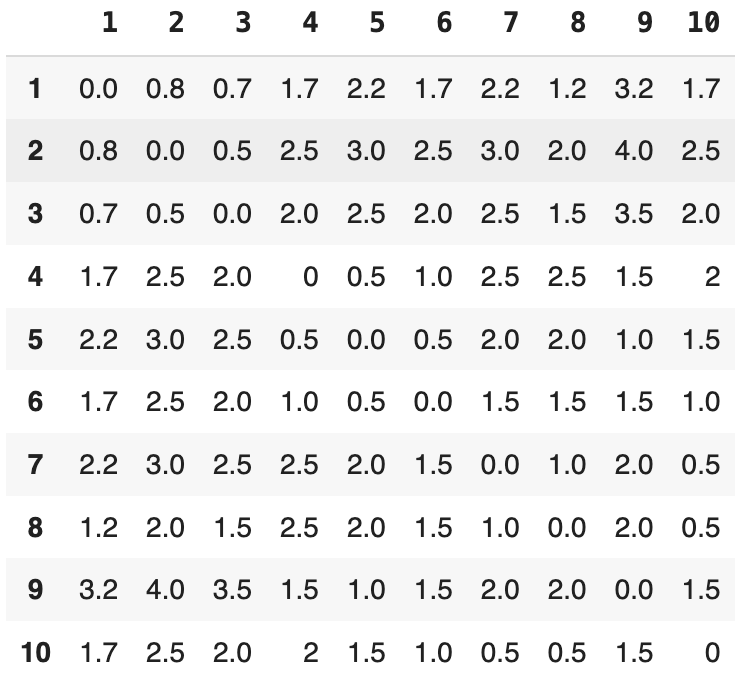
ه) در الگوریتم DBSCAN بر اساس نزدیکی و تراکم داده‌ها، خوشه‌بندی انجام شده است. از طرفی در الگوریتم KMeans با اختلاف فاصله و در واقع کمینه کردن واریانس در یک خوشه، خوشه‌بندی انجام شده است.

همانطور که از نتایج قسمت‌های قبلی مشخص است، تفاوت‌هایی در نتیجه‌ی خوشه‌بندی این دو الگوریتم وجود دارد. الگوریتم K-Means به نقاط outlier حساس است زیرا می‌توانند به طور قابل توجهی بر میانگین یک خوشه تأثیر بگذارند. یک نقطه دور از مرکز سایر نقاط داده می‌تواند مرکز را به سمت خود بکشد. این امر باعث می‌شود خوشه‌بندی نهایی بهینه نباشد. از طرفی برای الگوریتم DBSCAN، با توجه به منطق این الگوریتم، نقاط outlier را به خوبی شناسایی می‌کند و این باعث می‌شود در نهایت خوشه‌بندی مناسب‌تری شکل بگیرد. علت این که الگوریتم DBSCAN در تشخیص outlierها بهتر عمل می‌کند استفاده از تراکم داده‌ها با دو پارامتر Eps و MinPts است که در بخش ب تاثیر این پارامترها بررسی شد.

برای الگوریتم DBSCAN تنها کاری که می‌تواند به تشخیص درست نقاط outlier کمک کند، تنظیم پارامترهای ذکر شده به بهترین نحو است. اگر پارامتر Eps خیلی کم باشد منجر به آن می‌شود که نقاط outlier بیشتری تشخیص داده شود. از طرفی افزایش آن نیز باعث می‌شود نقاط outlier واقعی شناسایی نشوند. بنابراین برای این الگوریتم بهترین روش تنظیم بهینه‌ی پارامترها می‌باشد.

برای الگوریتم KMeans با توجه به ضعفی که وجود دارد، در مرحله‌ی اول، قبل از اعمال الگوریتم می‌توان پیش‌پردازش بهتری انجام داد تا الگوریتم با outlierهای کمتری مواجه باشد. از طرفی نقاط انتخابی اولیه نیز می‌تواند تاثیر داشته باشد. روش دیگر آن است که به جای استفاده از mean می‌توان از medoid و median استفاده کرد. همچنین انتخاب بهترین K نیز، تاثیر خوبی می‌تواند داشته باشد.

و) ابتدا با روش single link مسئله حل می‌شود. جدول فاصله‌ی منهتن داده شده در زیر آورده شده است.



بر اساس جدول بالا کمترین مقدار به دست می‌آید. نقاط P2, P3 و P4, P5 و P7, P10 کمترین فاصله را از یکدیگر دارند که برابر ۰.۵ می‌باشد.

با توجه به نقطه‌های انتخابی، جدول اندازه‌ها تغییر می‌کند.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P9 | P8 | P7, P10 | P6 | P4, P5 | P2, P3 | P1 | Points |
| 3.2 | 1.2 | 1.7 | 1.7 | 1.7 | 0.7 | 0 | P1 |
| 3.5 | 1.5 | 2 | 2 | 2 | 0 |  | P2, P3 |
| 1 | 2 | 1.5 | 0.5 | 0 |  |  | P4, P5 |
| 1.5 | 1.5 | 1 | 0 |  |  |  | P6 |
| 1.5 | 0.5 | 0 |  |  |  |  | P7, P10 |
| 2 | 0 |  |  |  |  |  | P8 |
| 0 |  |  |  |  |  |  | P9 |

در ادامه با توجه به جدول بالا، کمترین مقدار بین نقاط برای مرحله‌ی بعد به دست می‌آید. P4, P5 با P6 و P7, P10 با P8 کمترین مقادیر می‌باشند. نتیجه‌ی جدول در مرج بعدی در زیر آورده شده است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P9 | P7, P10, P8 | P4, P5, P6 | P2, P3 | P1 | Points |
| 3.2 | 1.2 | 1.7 | 0.7 | 0 | P1 |
| 3.5 | 1.5 | 2 | 0 |  | P2, P3 |
| 1 | 1 | 0 |  |  | P4, P5, P6 |
| 1.5 | 0 |  |  |  | P7, P10, P8 |
| 0 |  |  |  |  | P9 |

در مرج بعدی P1 با P2,P3 مرج می‌شود زیرا ۰.۷ کمترین مقدار می‌باشد.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P9 | P7, P10, P8 | P4, P5, P6 | P1, P2, P3 | Points |
| 3.2 | 1.2 | 1.7 | 0 | P1, P2, P3 |
| 1 | 1 | 0 |  | P4, P5, P6 |
| 1.5 | 0 |  |  | P7, P10, P8 |
| 0 |  |  |  | P9 |

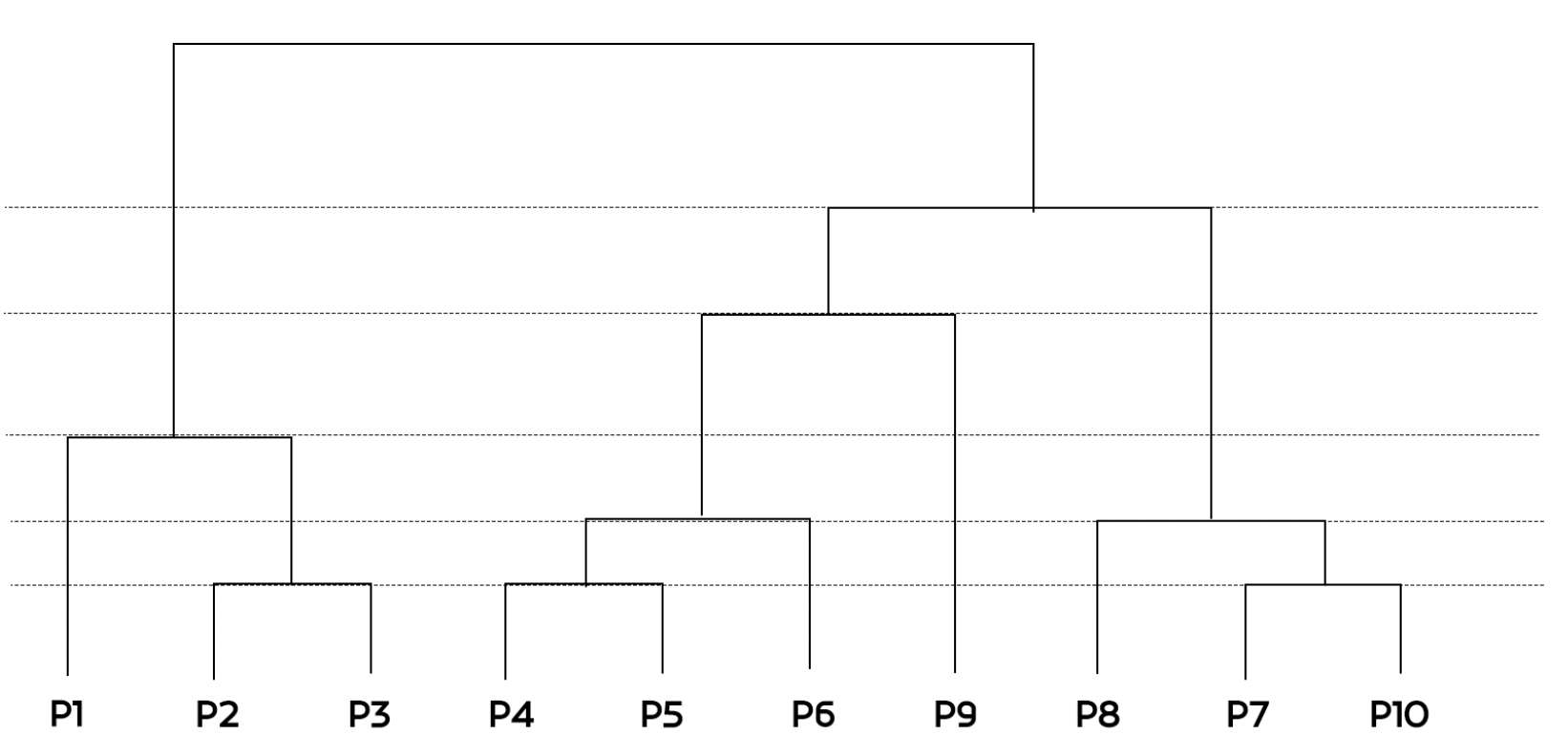
در مرج بعدی P4, P5, P6 با P9 مرج می‌شود زیرا ۱ کمترین مقدار می‌باشد.

|  |  |  |  |
| --- | --- | --- | --- |
| P7, P10, P8 | P4, P5, P6, P9 | P1, P2, P3 | Points |
| 1.2 | 1.7 | 0 | P1, P2, P3 |
| 1 | 0 |  | P4, P5, P6, P9 |
| 0 |  |  | P7, P10, P8 |

در مرج بعدی P4, P5, P6, P9 با P7, P10, P8 مرج می‌شود زیرا ۱ کمترین مقدار می‌باشد.

|  |  |  |
| --- | --- | --- |
| P4, P5, P6, P9, P7, P10, P8 | P1, P2, P3 | Points |
| 1.2 | 0 | P1, P2, P3 |
| 0 |  | P4, P5, P6, P9, P7, P10, P8 |

Dendrogram با روش Single-link برای مجموعه نقاط داده شده در زیر آورده شده است.



حال با روش complete link مسئله حل می‌شود. این روش مانند روش قبلی است با این تفاوت که پس از مرج شدن، بیشترین فاصله‌ی ممکن معیار می‌باشد.

بنابراین برای مرج اول همانند روش قبل از نقاط ذکر شده استفاده می‌شود.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P9 | P8 | P7, P10 | P6 | P4, P5 | P2, P3 | P1 | Points |
| 3.2 | 1.2 | 2.2 | 1.7 | 2.2 | 0.8 | 0 | P1 |
| 4 | 2 | 3 | 2.5 | 3 | 0 |  | P2, P3 |
| 1.5 | 2.5 | 2.5 | 2.5 | 0 |  |  | P4, P5 |
| 1.5 | 1.5 | 1.5 | 0 |  |  |  | P6 |
| 2 | 1 | 0 |  |  |  |  | P7, P10 |
| 2 | 0 |  |  |  |  |  | P8 |
| 0 |  |  |  |  |  |  | P9 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P9 | P8 | P7, P10 | P6 | P4, P5 | P1, P2, P3 | Points |
| 4 | 2 | 3 | 2.5 | 3 | 0 | P1, P2, P3 |
| 1.5 | 2.5 | 2.5 | 2.5 | 0 |  | P4, P5 |
| 1.5 | 1.5 | 1.5 | 0 |  |  | P6 |
| 2 | 1 | 0 |  |  |  | P7, P10 |
| 2 | 0 |  |  |  |  | P8 |
| 0 |  |  |  |  |  | P9 |

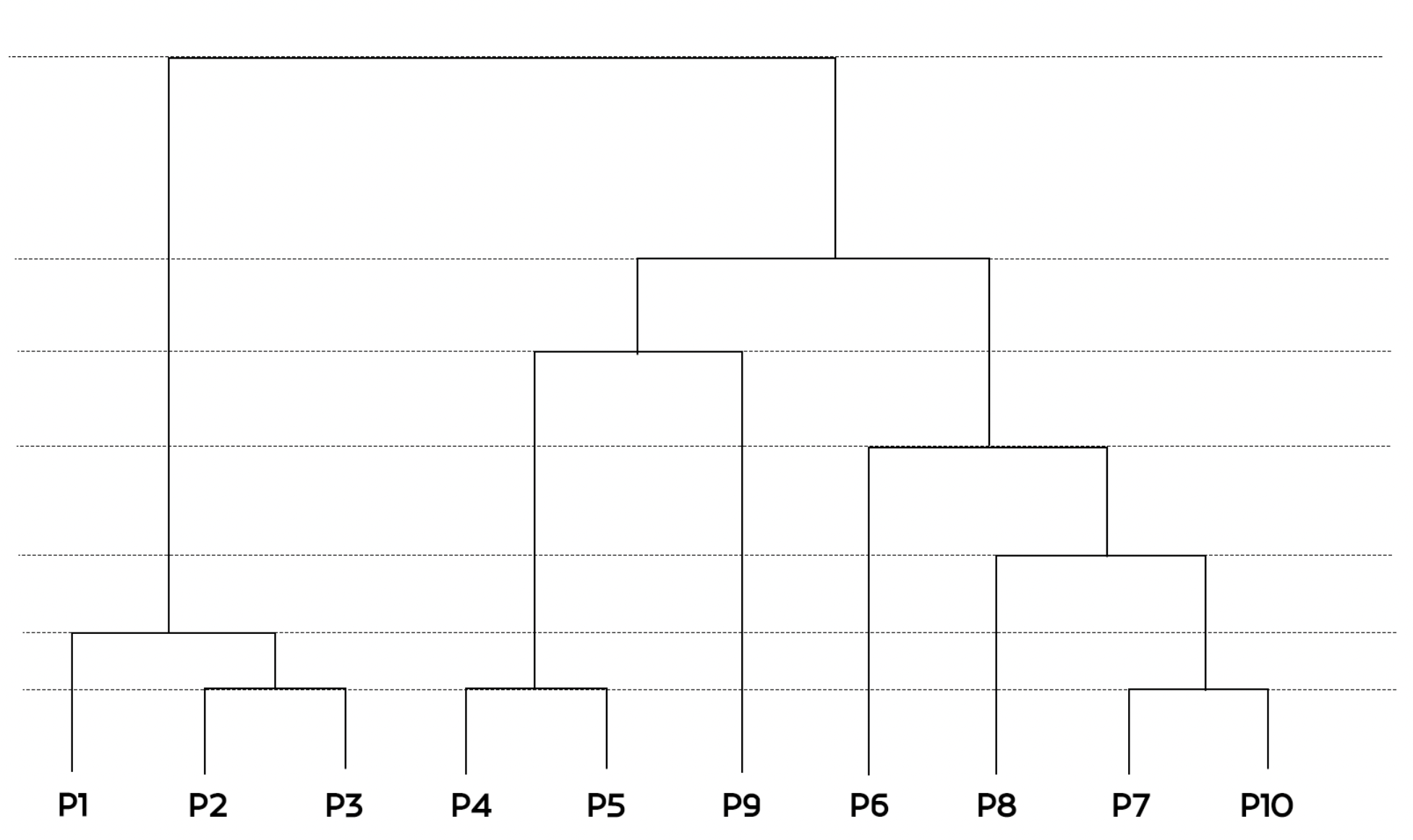
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P9 | P7, P10, P8 | P6 | P4, P5 | P1, P2, P3 | Points |
| 4 | 3 | 2.5 | 3 | 0 | P1, P2, P3 |
| 1.5 | 2.5 | 2.5 | 0 |  | P4, P5 |
| 1.5 | 1.5 | 0 |  |  | P6 |
| 2 | 0 |  |  |  | P7, P10, P8 |
| 0 |  |  |  |  | P9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P9 | P6, P7, P10, P8 | P4, P5 | P1, P2, P3 | Points |
| 4 | 3 | 3 | 0 | P1, P2, P3 |
| 1.5 | 2.5 | 0 |  | P4, P5 |
| 2 | 0 |  |  | P6, P7, P10, P8 |
| 0 |  |  |  | P9 |

|  |  |  |  |
| --- | --- | --- | --- |
| P6, P7, P10, P8 | P4, P5, P9 | P1, P2, P3 | Points |
| 3 | 4 | 0 | P1, P2, P3 |
| 2.5 | 0 |  | P4, P5, P9 |
| 0 |  |  | P6, P7, P10, P8 |

|  |  |  |
| --- | --- | --- |
| P4, P5, P9, P6, P7, P10, P8 | P1, P2, P3 | Points |
| 4 | 0 | P1, P2, P3 |
| 0 |  | P4, P5, P9, P6, P7, P10, P8 |

با توجه به مراحل بالا Dendrogram به روش complete link در زیر آورده شده است.



و) مورد اول آن که این الگوریتم زمان محاسباتی بالایی دارد و زمانی که حجم دیتا بیشتر شود نیز این هزینه‌ی محاسباتی بیشتر خواهد بود. مورد دیگر آن که نسبت به نقاط outlier حساس است و این نقاط به راحتی می‌توانند مسیر رشد خوشه‌‌بندی را تغییر دهند. مورد آخر آن که مقدار مشخص و تعیین کننده‌ای نمی‌توان برای تعداد خوشه‌ها برای آن مشخص کرد و شرط اتمام می‌تواند سطح خوشه‌بندی باشد که تعداد خوشه‌ها در آن مشخص نیست.

سوال دوم)

الف) به منظور محاسبه‌ی انتروپی شرطی برای هر کدام از خوشه‌ها، از رابطه‌ی زیر استفاده می‌شود.

در نتیجه داریم:

انتروپی شرطی، میزان عدم قطعیت را اندازه‌گیری می‌کند. هر چه انتروپی شرطی کمتر باشد قطعیت دسته بالاتر است. به طور مثال برای خوشه‌ی اول با توجه به کمترین مقدار محاسبه شده نسبت به باقی خوشه‌ها می‌توان اشاره کرد که قطعیت بالاتری دارد. در حالی که خوشه‌ی دوم برعکس خوشه‌ی اول می‌باشد.

ب) با توجه به آن که اندازه‌ی همه‌ی خوشه‌ها با یکدیگر برابر است *در فرمول زیر برابر با ۰.۳۳ می‌باشد.*

بنابراین برای محاسبه‌ی پارامتر خواسته شده در این سوال داریم:

ج) mutual information بین دسته‌ها و خوشه‌ها معیاری است که میزان اطلاعاتی را که دسته‌ها در مورد خوشه‌ها ارائه می‌کند، کمی می‌کند. اگر مقدار mutual information بین دسته‌ها و خوشه‌ها بالا باشد، نشان‌دهنده وابستگی یا ارتباط قوی بین خوشه‌بندی و دسته‌هاست و این به معنای آن است که خوشه‌بندی در گروه‌بندی موارد مشابه بر اساس دسته‌های حقیقی موثر است. از طرفی اگر مقدار آن برابر ۰ شود، به معنای آن است که دسته و خوشه از یکدیگر مستقل هستند و ممکن است خوشه‌بندی برای دسته‌ها معنایی نداشته باشند.

د) مقدار purity کل از رابطه‌ی زیر محاسبه می‌شود.

ابتدا می‌بایست رایج‌ترین دسته در هر خوشه مشخص شود.

برای خوشه‌ی اول تا سوم به ترتیب impulse buyers، discount seekers و loyal costumers رایج‌ترین‌ها می‌باشند. برای purity هر کدام از خوشه‌ها داریم:

Cluster 1: purity =

Cluster 2: purity =

Cluster 3: purity =

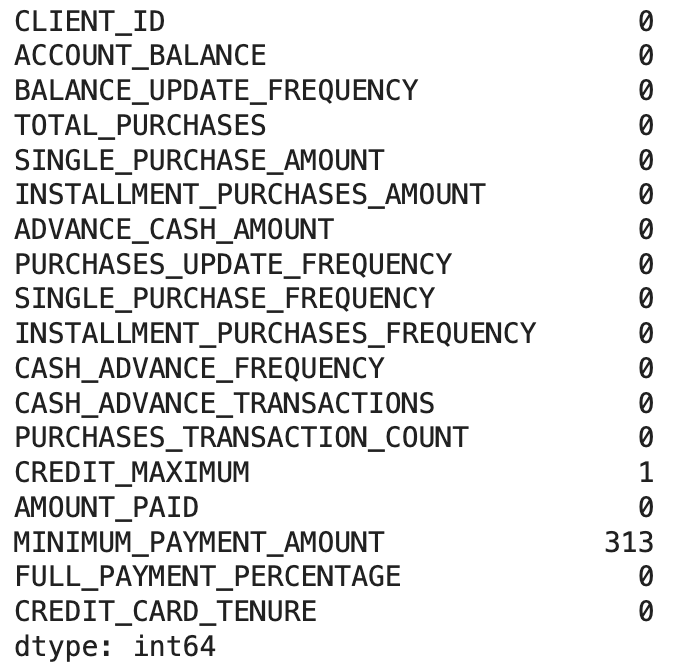
برای محاسبه‌ی purity کل باید مجموع کل دسته‌های رایج‌تر بر کل داده‌ها تقسیم شود.

بخش عملی)

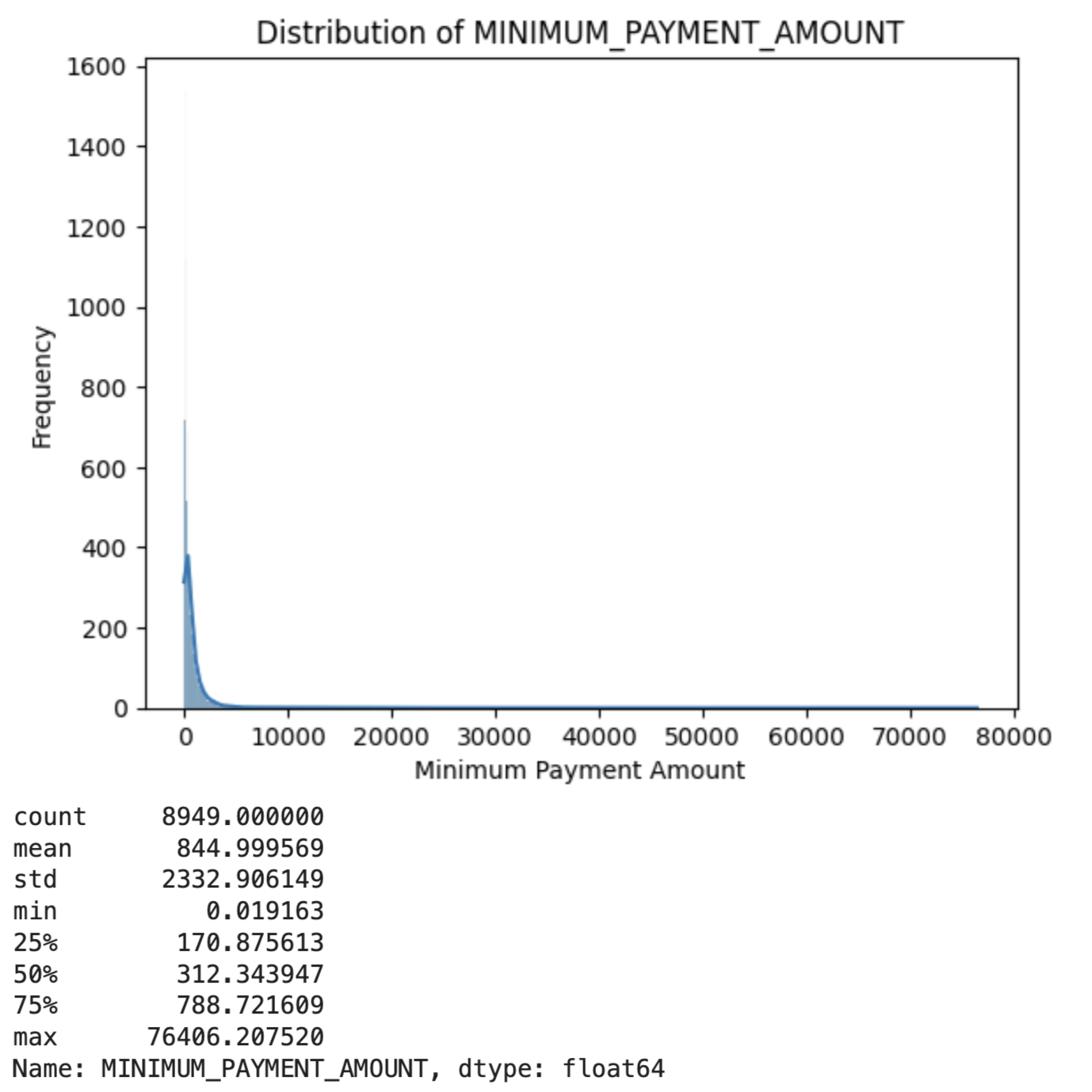
آشنایی با داده

ابتدا دیتاست در google colab آپلود می‌شود. سپس با دستور pd.read\_csv() دیتاست آماده‌ی پیش‌پردازش می‌باشد.

الف) برای به دست آوردن تعداد ردیف‌های دیتاست از shape[0] روی دیتافریم ایجاد شده استفاده می‌شود. تعداد ردیف‌های این دیتاست برابر ۸۹۵۰ می‌باشد. همچنین برای مشخص‌سازی تعداد missing valueها در هر ستون از دستور data.isnull().sum() استفاده می‌شود. نتیجه‌ی اجرای این دستور در زیر آورده شده است.

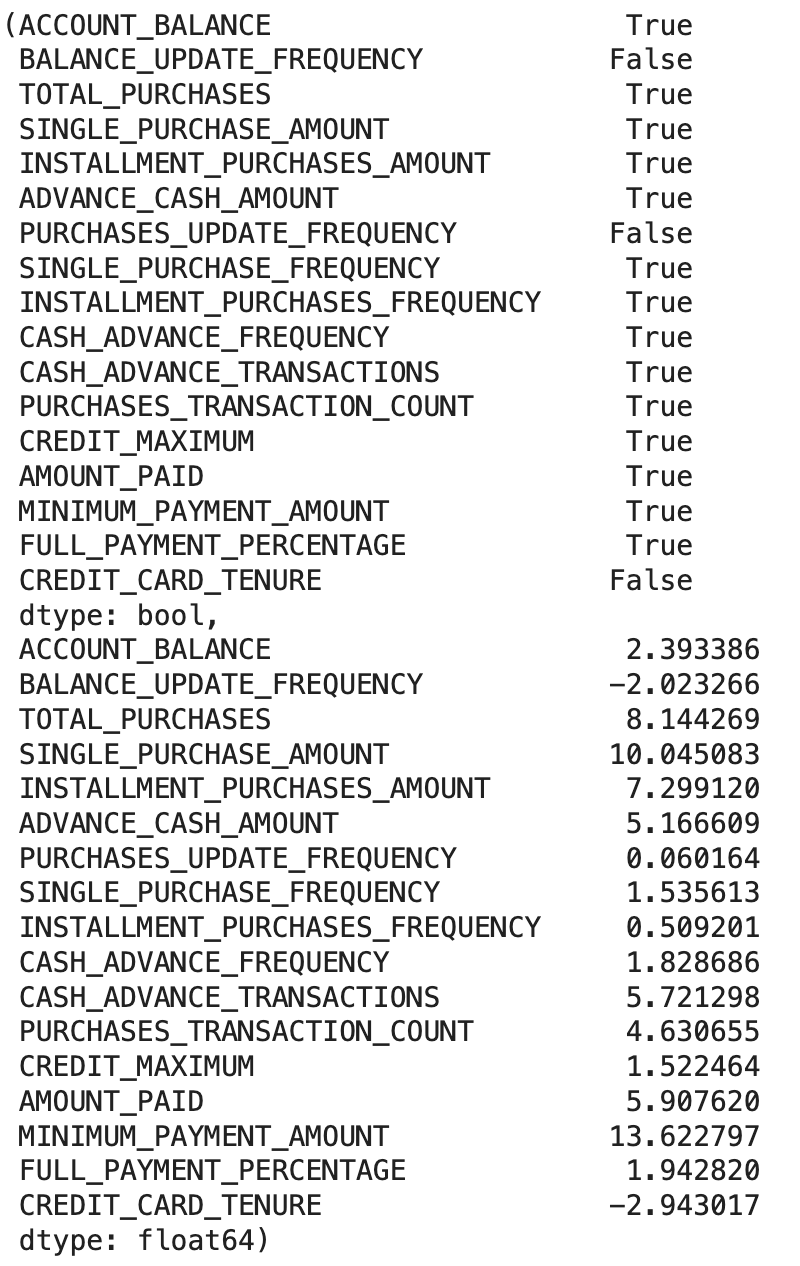


همانطور که مشخص است فقط دو ستون حاوی missing values هستند. برای ویژگی CREDIT\_MAXIMIUM، با توجه به آن که فقط یک ردیف شامل آن می‌شود، drop اتفاق می‌افتد و کل آن ردیف از دیتاست حذف می‌شود. برای MINIMUM\_PAYMENT\_AMOUNT با توجه به آن که تعداد زیادی ردیف را شامل می‌شود، drop کردن آن باعث از دست رفتن بخش قابل توجهی از دیتاست می‌شود. بنابراین بهترین روش استفاده از میانه یا میانگین می‌باشد. برای آن که انتخاب بهتر انجام شود، ابتدا skew داده‌های مربوط به این ویژگی بررسی می‌شود. پیاده‌سازی کد این بخش با کمک کتابخانه‌ی seaborn انجام شده است. همچنین با استفاده از تابع describe() پارامترهای آماری مربوط به این داده‌ها استخراج می‌شود. نتیجه‌ی اجرای این کد در زیر آورده شده است.



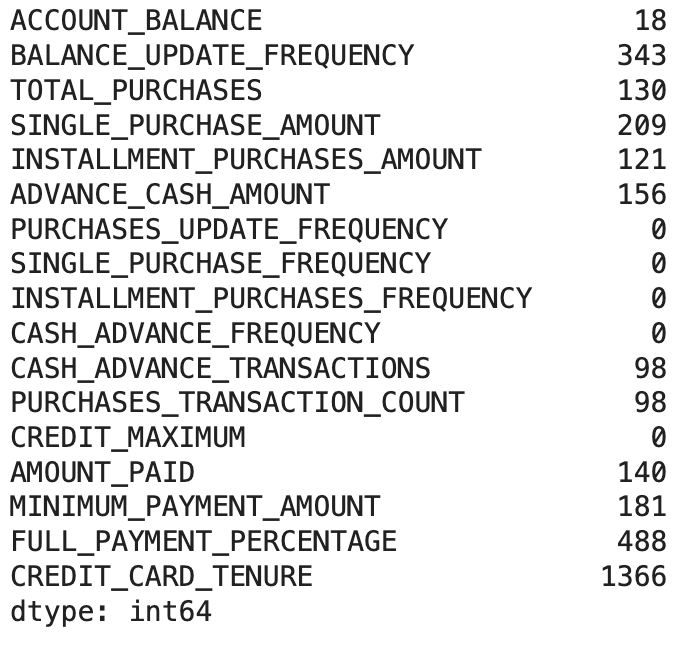
همانطور که از شکل و نتیجه‌ی بالا بر می‌آید، مقدار mean از median بیشتر می‌باشد. بنابراین دیتا در این ویژگی چولگی به راست دارند. از آنجایی که median در مقایسه با mean در توزیع‌هایی که چولگی دارند کمتر تحت تاثیر outlierها قرار می‌گیرند، بنابراین در این ویژگی برای پر کردن missing values استفاده از median انتخاب بهتری می‌باشد.

ب) ابتدا باید در نظر داشت که برای بررسی موارد ذکر شده می‌بایست ستون‌هایی که عددی هستند جدا شوند. در این مرحله به منظور بررسی چولگی داده‌ها از skew() استفاده شده است. این تابع عدد منفی و مثبت برمی‌گرداند. هر چه عدد مثبت‌تر باشد، چولگی دیتا به راست بیشتر است و برعکس. از روش دیگری نیز می‌توان چوله به راست بودن یا چپ بودن هر ویژگی را تشخیص داد. مقادیر mean و median از هر ویژگی به دست می‌آیند. در نهایت با مقایسه‌ی بزرگتر بودن mean نسبت به median مشخص می‌شود که چولگی به سمت راست می‌باشد. نتیجه‌ی اجرای این دستورات در زیر آورده شده است.



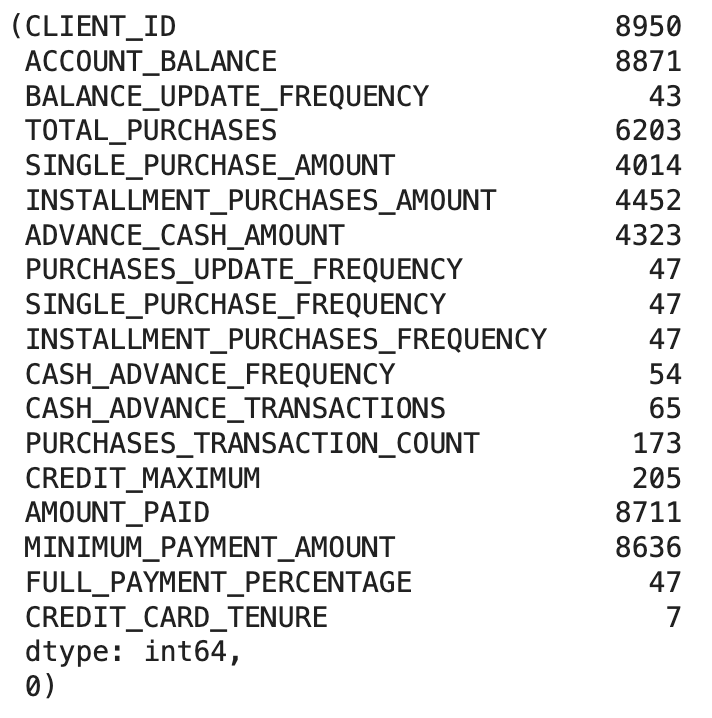
برای حل مشکل چولگی می‌توان از استاندارد کردن دیتا قبل از اعمال الگوریتم‌های مدنظر سوال در ادامه شد. استفاده از StandardScaler() از کتابخانه‌ی scikit-learn می‌تواند کمک زیادی بکند زیرا این تابع میانگین داده را روی صفر و std یک تنظیم می‌کند.

برای پیدا کردن outlierها از روش‌های مختلفی استفاده می‌شود. یکی از این روش‌ها حذف ۵ درصد از اول و آخر دیتاست برای هر ویژگی است. روش دیگر استفاده از IQR است. استفاده از ضریب ۱.۵ باعث می‌شود که بخش زیادی از داده به عنوان outlier در نظر گرفته شود. اگر منظور سوال استفاده از همین روش باشد ضریب IQR برابر ۱.۵ در نظر گرفته می‌شود اما در این حالت برای این که دیتای مفید حذف نشود، از ضریب 7 استفاده شده است. نتیجه‌ی اجرای کد به شکل زیر خواهد بود.

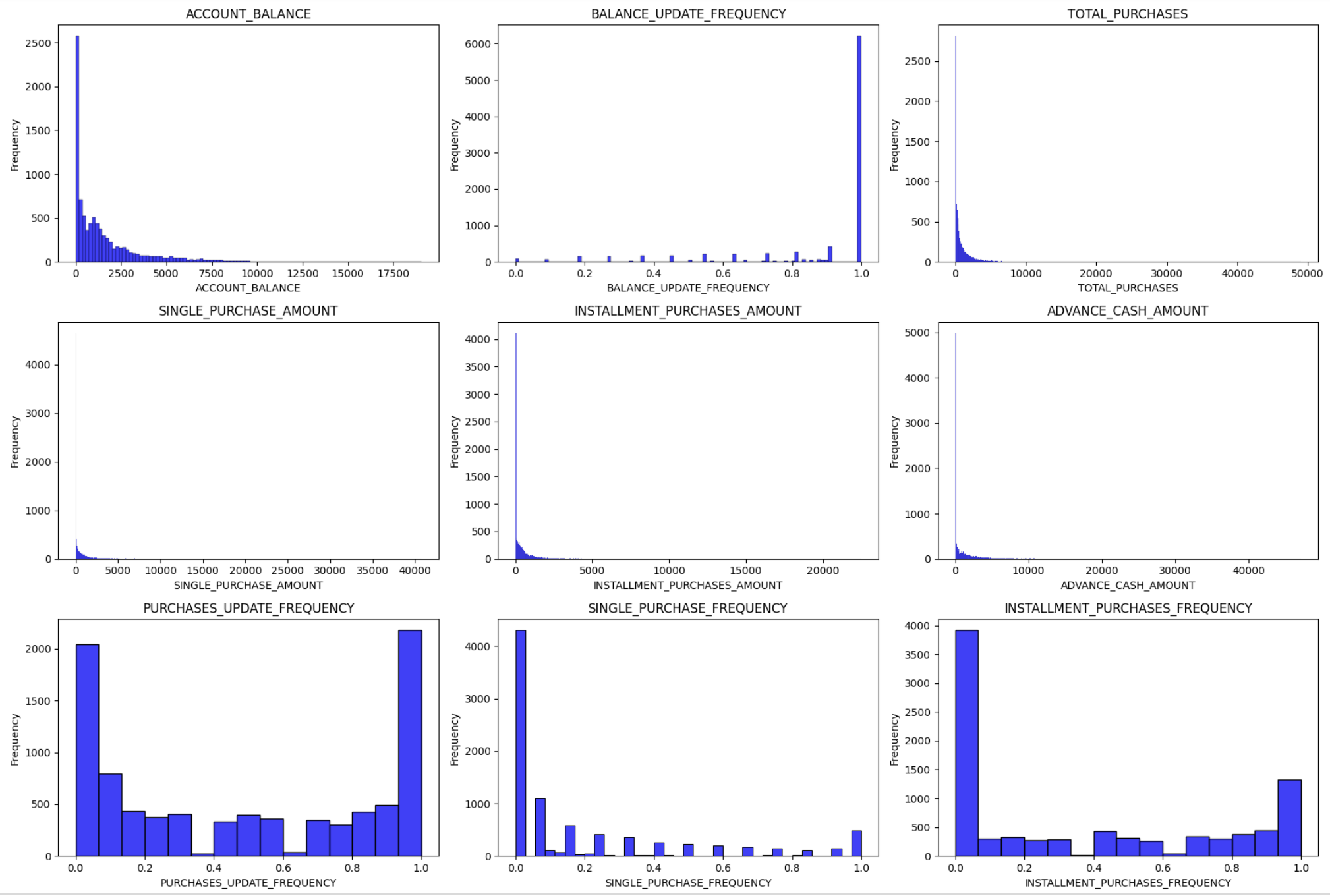


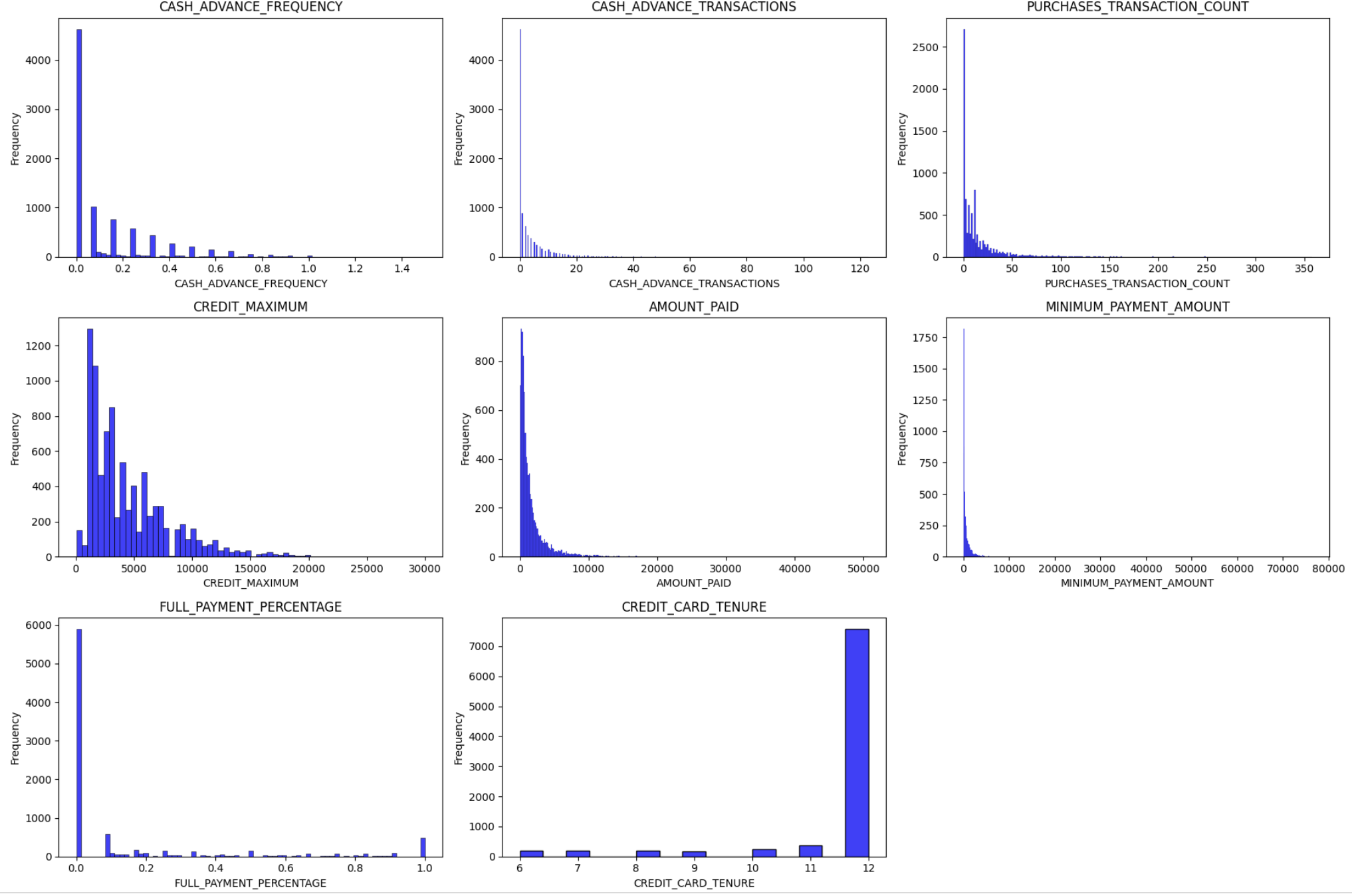
برای رسیدن به ضریب ۶، bar chart و scatter chart برای هر کدام از ویژگی‌ها رسم شده است تا بتوان ضریبی که بهترین نتیجه را می‌دهد انتخاب شود. کد‌های مربوط به این بخش‌ها در فایل ارسالی ضمیمه شده است.

ج) با استفاده از تابع unique() تعداد مقادیر یونیک هر ویژگی استخراج می‌شود. همچنین برای به دست آوردن تعداد سطرهای تکراری، از تابع duplicate() در کنار sum() استفاده می‌شود. نتیجه‌ی اجرای کد به شکل زیر خواهد بود.



د) ابتدا برای تمامی ویژگی‌ها histogram رسم می‌شود. شکل‌های زیر histogram برای هر کدام از ویژگی‌ها می‌باشد.

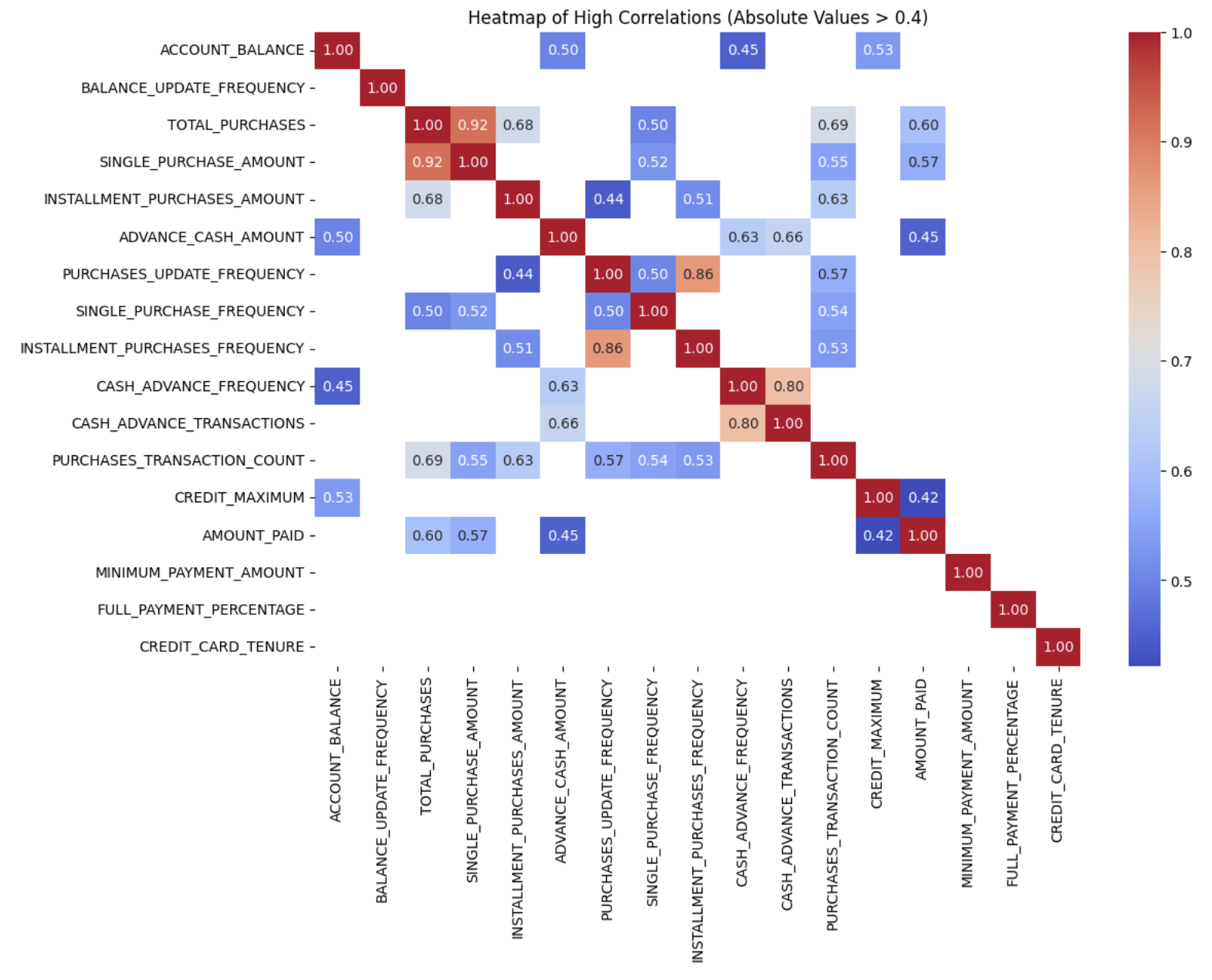




با توجه به نمودارهای بالا می‌توان دریافت که:

* با توجه به نمودار AMOUNT\_PAID می‌توان متوجه الگوهای پرداخت شد، همانطور که از سطوح پرداخت مشخص است، در سطوح پایین‌تر پرداخت می‌توان متوجه شد که بسیاری از مشتریان فقط حداقل پرداخت ها را انجام می‌دهند.
* نمودار ADVANCE\_CASH\_AMOUNT نشان می‌دهد که پیش پرداخت‌های نقدی اغلب توسط اکثریت استفاده نمی‌شود، اما بخش کوچکی به شدت از پیش‌پرداخت‌های نقدی استفاده می‌کنند.
* نمودار ACCOUNT\_BALANCE توزیع با چولگی به سمت راست را دارد. این به آن معناست که اکثر مشتریان موجودی کمی دارند اما تعداد کمی از آن‌ها به میزان قابل توجهی موجودی زیادی دارند.
* از مقایسه‌ی سه نمودار TOTAL\_PURCHASES، SINGLE\_PURCHASE\_AMOUNT و INSTALLMENT\_PURCHASES\_AMOUNT می‌توان رفتار و مدل مشتریان در خرج کردن را استخراج کرد. این مقایسه نشان می‌دهد که آیا مشتریان خریدهای تکی بزرگ را ترجیح می‌دهند یا خریدهای اقساطی کوچکتر و مکرر.
* نمودار CASH\_ADVANCE\_FREQUENCY می‌تواند به شناسایی تعداد دفعاتی که مشتریان در برابر محدودیت اعتبار خود پول نقد استفاده می‌کنند کمک کند. تمرکز نقاط داده نزدیک به صفر نشان دهنده استفاده کمتر است، در حالی که توزیع یکنواخت تر نشان دهنده استفاده منظم است. آشنایی با این پترن‌ها می‌تواند در تصمیمات اقتصادی تاثیر بسازیی داشته باشد.
* نمودارهای PURCHASES\_TRANSACTION\_COUNT و CASH\_ADVANCE\_TRANSACTIONS اطلاعاتی در مورد فعالیت معاملاتی مشتریان ارائه می‌دهد. فرکانس‌های بالا در این نمودارها نشان می‌دهد که مشتریان به طور فعال از کارت‌های اعتباری خود برای خرید و پیش پرداخت نقدی استفاده می‌کنند که نشان‌دهنده‌ی تعامل بیشتر با کارت اعتباری است.
* نمودار CREDIT\_CARD\_TENURE می تواند توزیع مدت زمان استفاده مشتریان از بانک یا کارت اعتباری را نشان دهد. توزیع منحرف به چپ، حساب‌های جدیدتر زیادی را نشان می‌دهد، که می‌تواند نشان‌دهنده‌ی استراتژی‌های اخیر جذب مشتری موفق باشد. برعکس، یک توزیع درست یا یکنواخت ممکن است نشان دهنده وفاداری طولانی مدت مشتری باشد.

ه) ابتدا به کمک کتابخانه‌های لازم، heatmap فقط برای ویژگی‌هایی که هم‌بستگی آن‌ها بیشتر از ۰.۴ است رسم می‌شود. شکل زیر heatmap اشاره شده را نشان می‌دهد.

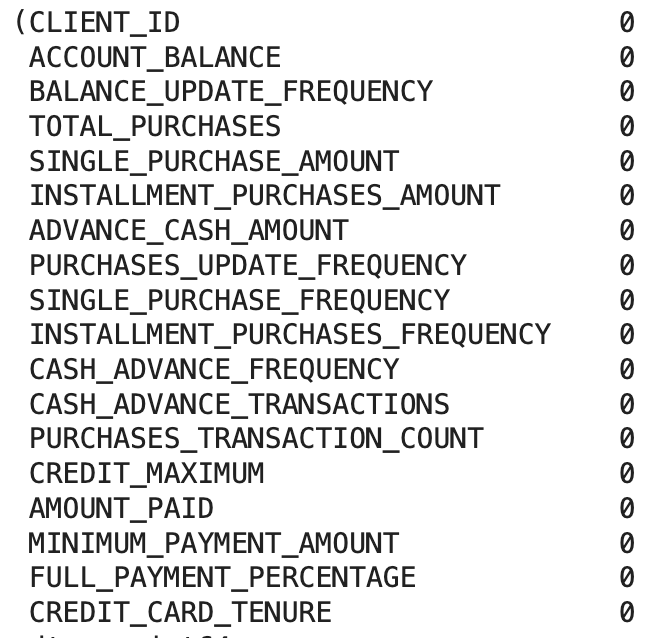


یکی ویژگی‌هایی که هم‌بستگی زیادی با یکدیگر دارند می‌تواند حذف شود. زیرا ویژگی‌هایی که با هم هم‌بستگی زیادی دارند حاوی اطلاعات تکراری هستند. به طور مثال از بین SINGLE\_PURCHASE\_AMOUNT و TOTAL\_PURCHASE یکی می‌تواند حذف شود.

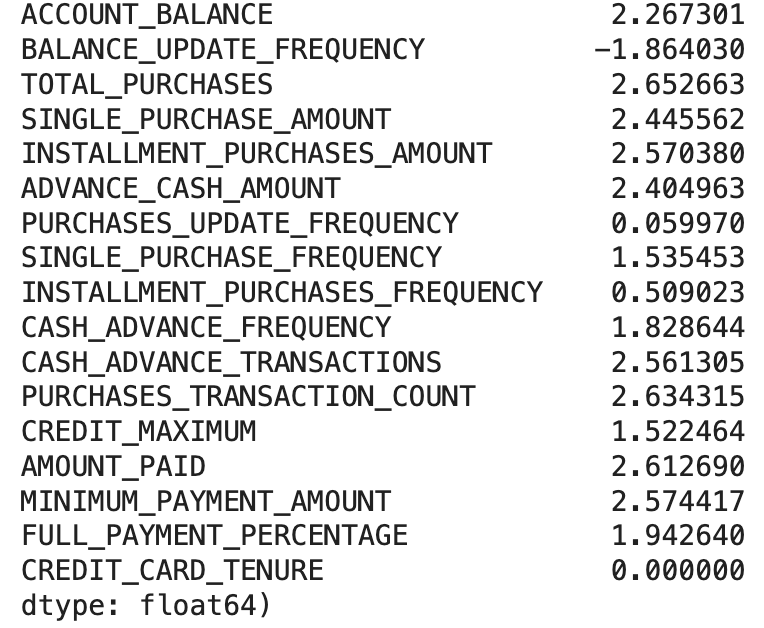
آماده‌سازی داده)

الف) همانطور که در قسمت‌های قبل توضیح داده شد، کد مربوط به این بخش پیاده شده است.

ب) همانطور که از شکل زیر مشخص است، تعداد missing values در ویژگی‌ها صفر است.



در ادامه با توجه به استراتژی مطرح شده برای چولگی، نتیجه‌ی زیر حاصل می‌شود که تا حد خوبی این مشکل را برطرف کرده است.



همچنین outlierها حذف شده‌اند و اندازه‌ی دیتاست برابر ۷۱۰۲ می‌باشد.

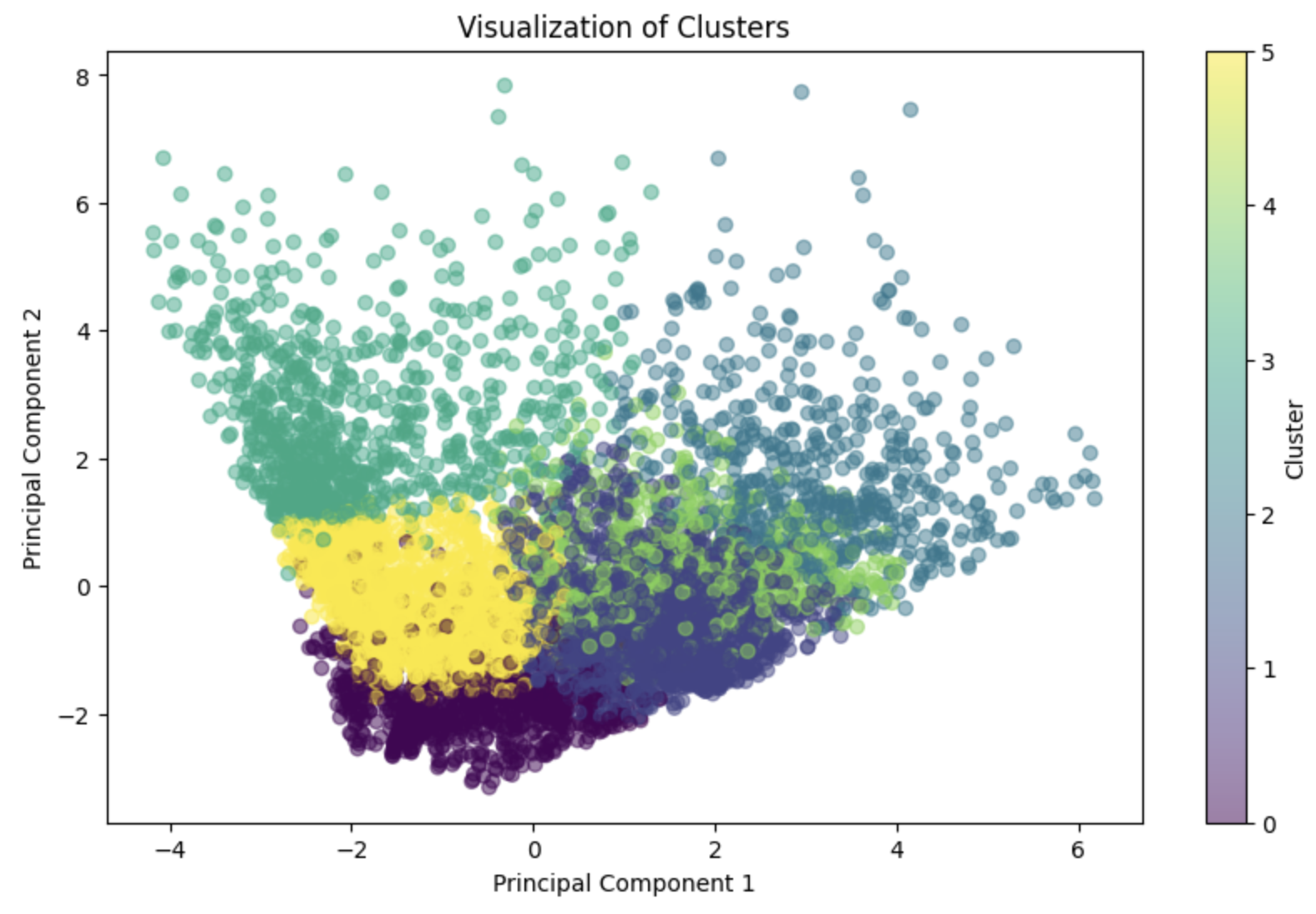
الگوریتم‌های خوشه‌بندی)

الف) برای پیدا کردن تعداد خوشه‌های بهینه از silhouette score استفاده می‌شود. silhouette score ارزیابی می‌کند که هر نقطه در یک خوشه چقدر به نقاط خوشه‌های همسایه نزدیک است و معیاری از متمایز بودن خوشه‌ها ارائه می‌کند.

شکل زیر نتیجه‌ی اجرای کد مربوط به این بخش است.

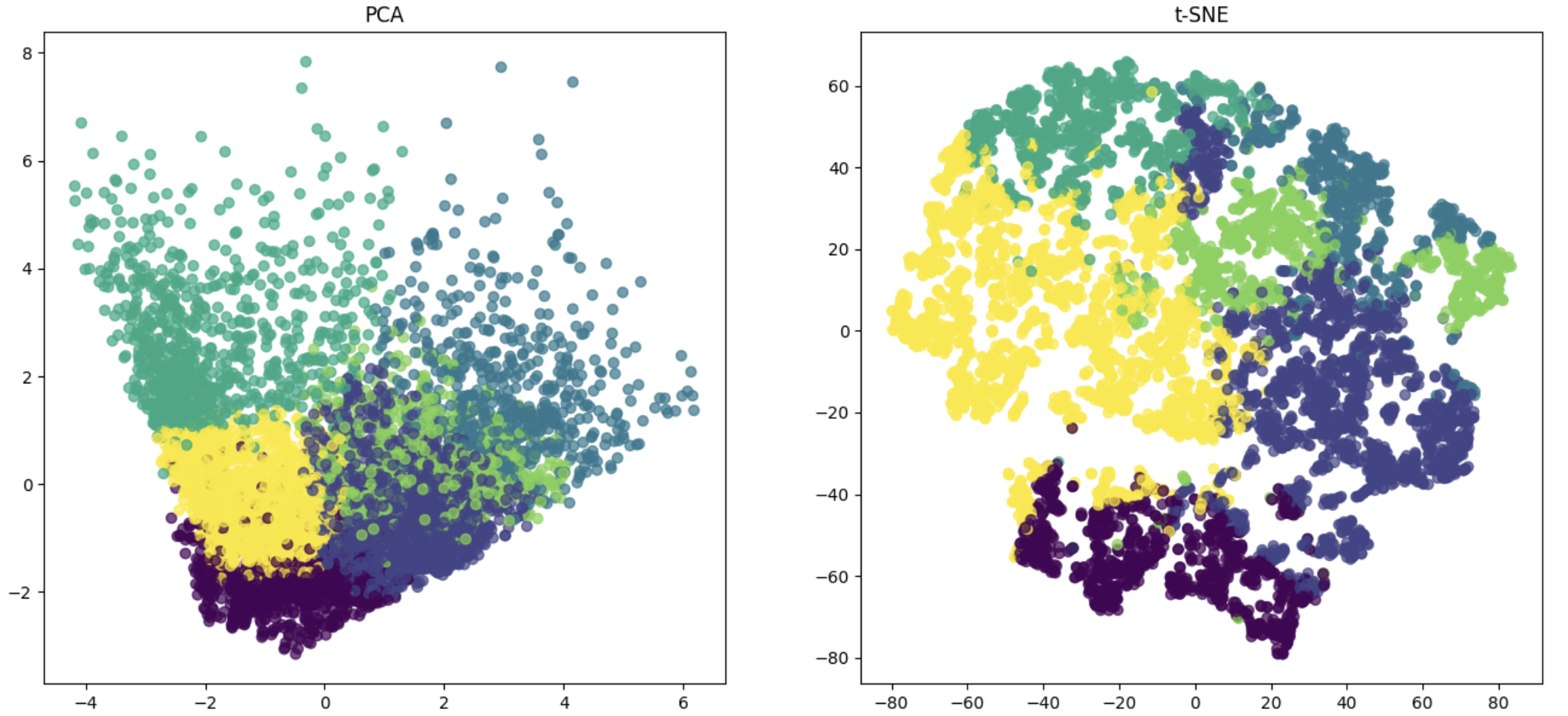


ب) الگوریتم KMean به خوبی در قسمت‌های تشریحی توضیح داده شده‌است. با توجه به قسمت قبل برای ۶ خوشه، خوشه‌‌بندی انجام می‌شود. همچنین نقاط centroid برای هر خوشه ثبت می‌شود. در ادامه به دیتافریم اصلی، لیبل هر خوشه زده می‌شود. نتیجه‌ی پیاده‌سازی کد، در زیر آورده شده است.



ج) رسم نتیجه‌ی خوشه‌بندی با استفاده از روش PCA در قسمت قبل انجام شده است.

برای مقایسه‌ی دو روش PCA و t-SNE ابتدا باید KMean روی الگوریتم‌ها اعمال شود. سپس با استفاده از این دو روش، خروجی‌ها مقایسه شوند. شکل زیر خوشه‌بندی با تنظیم ۶ خوشه، برای دو روش ذکر شده را نشان می‌دهد.



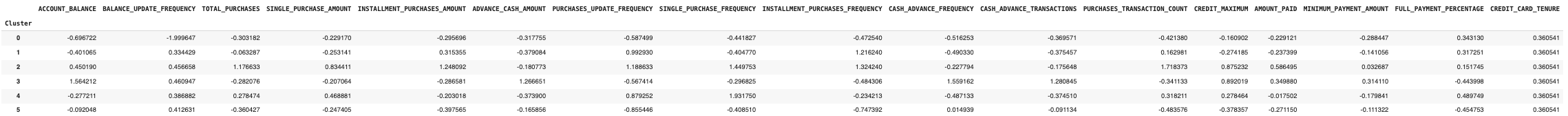
همانطور که از مقایسه‌ی دو شکل بالا برای خوشه‌بندی دیتاست یکسان برمی‌آید، استفاده از روش t-SNE جداسازی واضح‌تری نسبت به PCA ارائه می‌دهد زیرا local relationship را حفظ می کند. البته از طرفی با توجه به این مورد در t-SNE، فواصل بین خوشه‌ها در نتیجه‌ی این روش قابل تفسیر نیست زیرا global relationship در این حالت حفظ نمی‌شود.

زمان اجرای روش t-SNE از PCA بیشتر است. به همین دلیل برای دیتاست‌ها بزرگ استفاده از روش PCA انتخاب مناسبی نمی‌باشد.

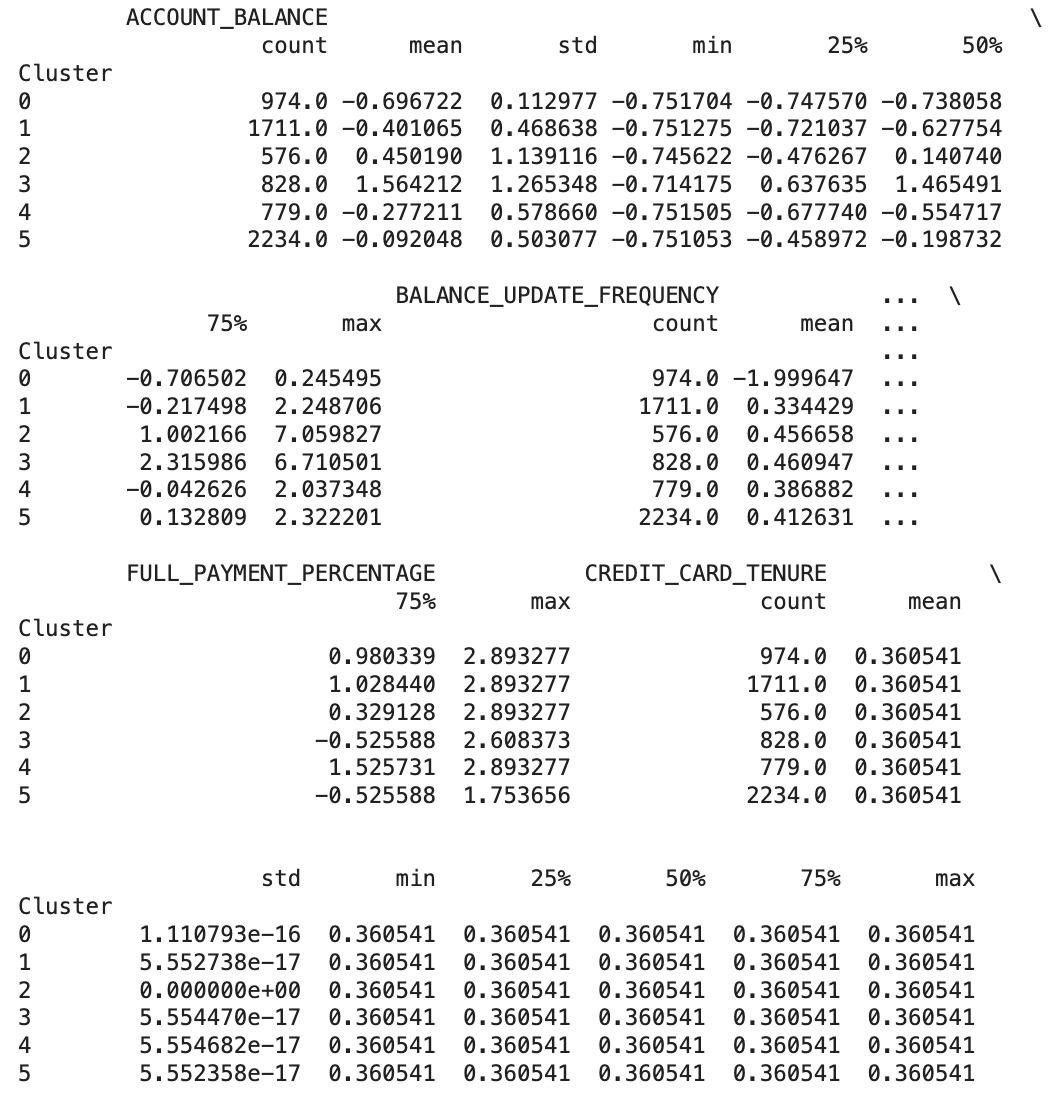
همچنین با توجه به شکل بالا با استفاده از روش PCA خوشه‌ها ممکن است با یکدیگر هم‌پوشانی داشته باشند و به خوبی نمایش داده نشوند اما نتیجه‌ی روش t-SNE این خوشه‌ها را تا حد خوبی جدا از هم نشان می‌دهد.

به طور کلی، برای ترسیم خوشه‌بندی، t-SNE خروجی بهتری را با جداسازی واضح خوشه‌ها ارائه می‌دهد، به‌ویژه اگر داده‌ها حاوی الگوهای پیچیده‌ای باشند که به‌صورت خطی قابل تفکیک نباشند. اما اگر هدف از ترسیم بررسی فاصله‌ی بین خوشه‌ها باشد یا دیتاست بزرگی داشته باشیم، PCA انتخاب بهتری می‌باشد.

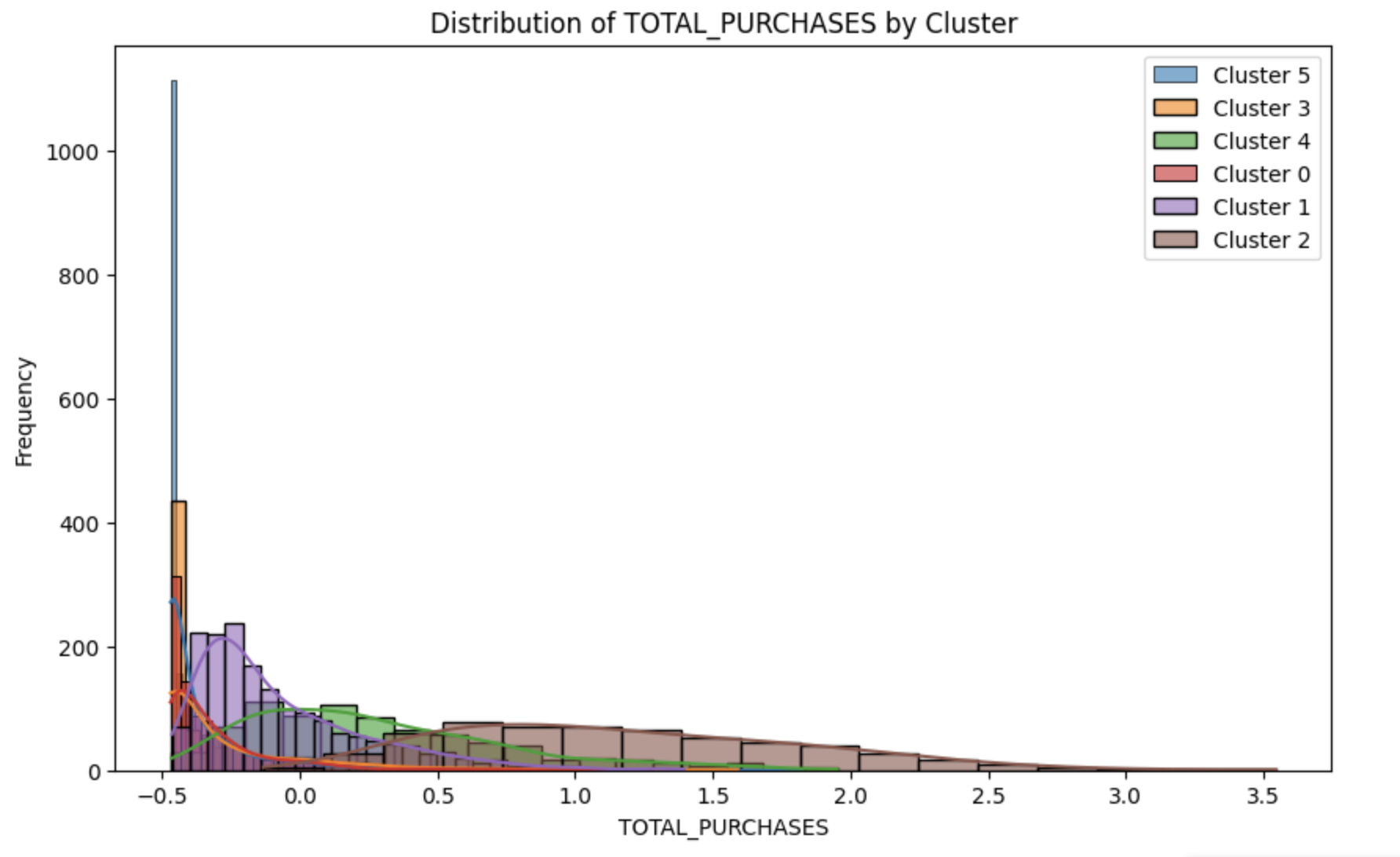
د) جدول زیر میانگین هر کدام از ویژگی‌ها را برای هر خوشه نشان می‌دهد. کد مربوط به این قسمت در فایل کد پیاده‌سازی شده است.



برای آن که بتوان برای هر کدام از خوشه‌ها توصیفی ارائه داد، بهتر است از ویژگی‌های آماری آن‌ها استفاده کرد. به همین منظور برای هر خوشه پارامترهای آماری استخراج می‌شود.



همچنین می‌توان با استفاده از هیستوگرام هر ویژگی توصیف خوبی از آن‌ها داشت به طور مثال برای TOTAL\_PURCHASE داریم:

**