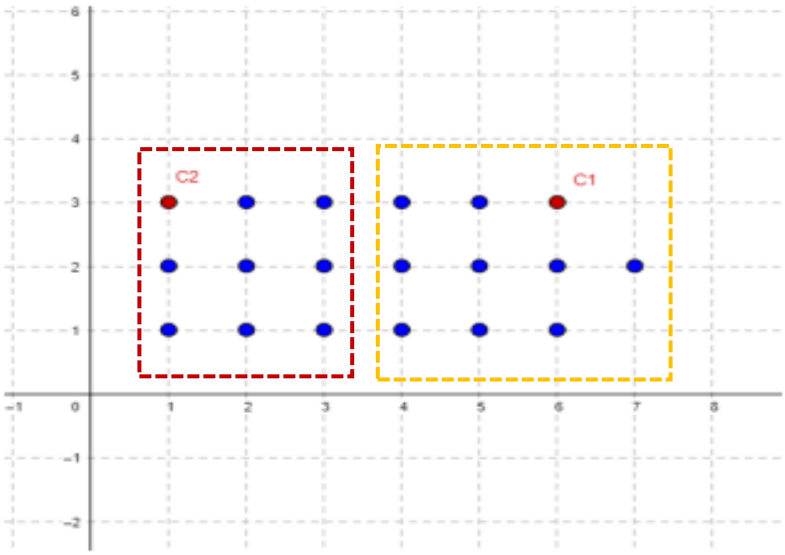
۱ ) ابتدا دو centroid داده شده است. حال باید فاصله تمامی نقاط را با دو نقطه c1 و c2 حساب کنیم و نسبت به هر کدام که نزدیک تر بود نقطه را به آن خوشه باید assign کنیم. خوشه بندی ها طبق شکل به صورت زیر انجام میشود:




نقطه های موجود در مستطیل قرمز رنگ در یک خوشه و نقاط موجود در مستطیل زرد رنگ در خوشه دیگر قرار میگیرند. پس خوشه های مربوط به هر نقطه مشخص شد حال باید مراکز خوشه هارا آپدیت کنیم، داریم:

خوشه ۲ :

$$X = (1 * 3 + 2 * 3 + 3 * 3) / 9 = 2$$

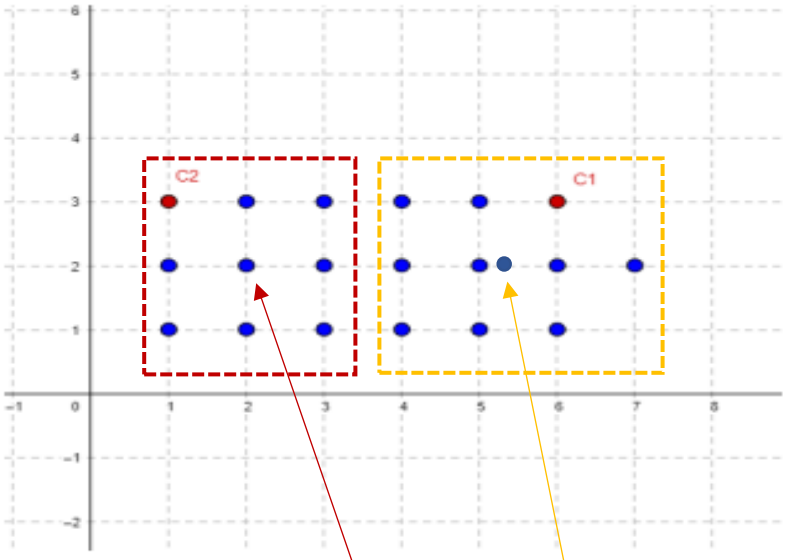
$$Y = (1 * 3 + 2 * 3 + 3 * 3) / 9 = 2$$

خوشه ۱ :

$$X = (4 * 3 + 5 * 3 + 6 * 3 + 7 * 1) / 10 = 5.2$$

$$Y = (1 * 3 + 2 * 4 + 3 * 3) / 10 = 2$$

حال مراکز آپدیت شدند. بار دیگر که فواصل نقاط با این مراکز را حساب بکنیم خوشه بندی ها عوض نمیشوند پس درنتیجه خوشه بندی نهایی به صورت زیر است:




C2 (2, 2)

C1(5.2, 2)

۲ )

برای دسته بندی داده های زیر از الگوریتم DBSCAN استفاده میکنیم. این الگوریتم ویژگی های زیر را دارد

- خوشه هایی که تشکیل میدهد میتوانند از نظر شکل و اندازه با یکدیگر متفاوت باشند
- تعداد خوشه ها لازم نیست از قبل مشخص شود، درواقع تعداد خوشه ها دیگر مثل الگوریتم Kmeans یک Hyperparamter نیست.
- داده های نویز و outlier ها را برخلاف الگوریتم Kmeans به خوبی هندل میکند و نمیگذارد این داده ها مثل الگوریتم Kmeans باعث ایجاد خوشه های نامناسب بشوند.
- مناطق چگال رو از مناطقی که نقاط چگالی کمتری دارند متمایز میکند.

طبق ویژگی های گفته شده برای داده های زیر از الگوریتم DBSCAN استفاده میکنیم زیرا طبق شکل میتوان متصور شد دو منطقه چگال داریم که الگوریتم DBSCAN به خوبی آنهارا تشخیص میدهد.

الگوریتم DBSCAN دارای دو هایپرپارامتر با نام های minPoints و eps میباشد. minPoints حداقل تعداد نقاطی است که باید در همسایگی یک نقطه در شعاع همسایگی eps باشند تا بتوانیم یک خوشه ایجاد کنیم. متغیر eps نیز میزان شعاع همسایگی را مشخص میکند.

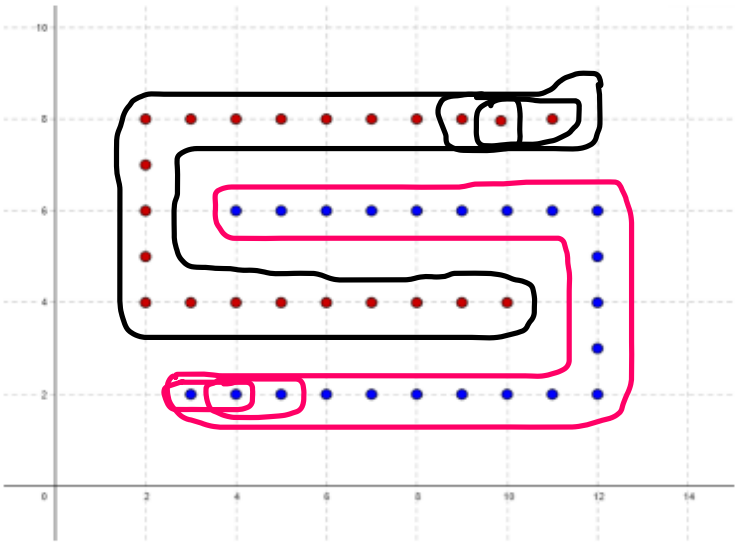
الگوریتم DBSCAN بدینصورت عمل میکند که ابتدا یک نقطه به صورت رندوم انتخاب میکنیم و تعداد همسایه های آن نقطه را بدست می‌آوریم. اگر تعداد همسایه ها از minPoints کمتر بود آنگاه آن نقطه را outlier تشخیص میدهیم. درغیر اینصورت آن نقطه هسته است و میتوانیم خوشه را ایجاد کنیم. بدین ترتیب تمام نقاط را بررسی میکنیم تا همه نقاط بررسی شوند. همچنین نقاطی که در داخل خوشه قرار میگیرند ولی خود هسته نیستند border نامیده میشوند و داده هایی که در داخل هیچ خوشه ای جای ندارند outlier نامیده میشوند.

برای این مسئله هایپرپارامتر هارا به صورت زیر درنظر میگیریم:

minPoints = 2

eps = 1

پس الگوریتم به صورت زیر ایجاد میشود:



مطابق شکل مشاهده میکنیم تعداد خوشه ها پس از اجرای الگوریتم DBSCAN برابر ۲ میشود و خوشه به درستی بر اساس تراکم نقاط تعیین شدند درحالی که اگر از الگوریتم Kmeans استفاده میکردیم خوشه بندی ها با هر نقاط ابتدایی به این صورت نمیشدند و خوشه بندی با الگوریتم Kmeans کیفیت کمتری داشت.

۳ )

Closed frequent itemset ها زیر مجموعه frequent itemset ها هستند. همچنین مجموعه ای بسته است و مقدار support آن بزرگتر و مساوی minsup میباشد. همچنین itemset ای بسته است که هیچ superset ای از این مجوعه وجود نداشته باشد که مقدار support یکسانی با آن داشته باشد. دراقع رابطه $\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$ برقرار است.

همچنین کنار هر نود میزان support آن را مشخص میکنیم و بدین صورت closed frequent itemset هارا بدست می‌آوریم. طبق رابطه ذکر شده حال با داشتن closed frequent itemset ها میتوانیم مقدار support را برای frequent itemset ها بدست آوریم.

( ۴

support = ۳۳٪

confidence = ۶۰٪

1-itemsets:

| | count | support | |
|---|---|---|---|
| سیب | ۴ | $\frac{۴}{۶} \geq \frac{۱}{۳}$ | ✓ |
| پرتقال | ۳ | $\frac{۳}{۶} \geq \frac{۱}{۳}$ | ✓ |
| موز | ۳ | $\frac{۳}{۶} \geq \frac{۱}{۳}$ | ✓ |
| انار | ۳ | $\frac{۳}{۶} \geq \frac{۱}{۳}$ | ✓ |
| نارنگی | ۲ | $\frac{۲}{۶} \geq \frac{۱}{۳}$ | ✓ |

← هیچ itemset ای دیگه نمی‌شود

2-itemsets:

| | count | support | |
|---|---|---|---|
| سیب و پرتقال | ۲ | $\frac{۲}{۶} \geq \frac{۱}{۳}$ | ✓ |
| سیب و موز | ۲ | $\frac{۲}{۶} \geq \frac{۱}{۳}$ | ✓ |
| سیب و انار | ۱ | $\frac{۱}{۶} < \frac{۱}{۳}$ | ✗ |
| سیب و نارنگی | ۲ | $\frac{۲}{۶} \geq \frac{۱}{۳}$ | ✓ |
| پرتقال و موز | ۲ | $\frac{۲}{۶} \geq \frac{۱}{۳}$ | ✓ |
| پرتقال و انار | ۱ | $\frac{۱}{۶} < \frac{۱}{۳}$ | ✗ |
| پرتقال و نارنگی | ۰ | $\frac{۰}{۶} < \frac{۱}{۳}$ | ✗ |
| موز و انار | ۱ | $\frac{۱}{۶} < \frac{۱}{۳}$ | ✗ |
| موز و نارنگی | ۰ | $\frac{۰}{۶} < \frac{۱}{۳}$ | ✗ |
| انار و نارنگی | ۱ | $\frac{۱}{۶} < \frac{۱}{۳}$ | ✗ |

→

| | count | |
|---|---|---|
| سیب و پرتقال و موز | ۲ | ✓ |
| سیب و پرتقال و نارنگی | ۰ | ✗ |
| سیب و موز و نارنگی | ۰ | ✗ |
| پرتقال و موز و نارنگی | ۰ | ✗ |

پس آیتم‌های دارای پشتیبانی بیشتر از ۲ سیب و پرتقال و موز است و حال قوانین انجمنی به صورت زیر نوشته می‌شوند:

{پرتقال و موز} ⟵ {سیب} $\quad confidence = \frac{\text{count}(\text{سیب و پرتقال و موز})}{\text{count}(\text{سیب})} = \frac{۲}{۴} < ۰/۶$ ✗

{پرتقال} ⟵ {سیب و موز} $\quad confidence = \frac{۲}{۳} > ۰/۶$ ✓

{موز} ⟵ {سیب و پرتقال} $\quad confidence = \frac{۲}{۳} > ۰/۶$ ✓

{سیب و پرتقال} ⟵ {موز} $\quad confidence = \frac{۲}{۲} > ۰/۶$ ✓

{سیب و موز} ⟵ {پرتقال} $\quad confidence = \frac{۲}{۲} > ۰/۶$ ✓

{موز و پرتقال} ⟵ {سیب} $\quad confidence = \frac{۲}{۲} > ۰/۶$ ✓

پس روابط به صورت زیر هستند:

(۱) {موز و پرتقال} ⟵ {سیب} $\quad c = ۱$

(۲) {سیب و موز} ⟵ {پرتقال} $\quad c = ۱$

(۳) {سیب و پرتقال} ⟵ {موز} $\quad c = ۱$

(۴) {سیب و پرتقال} ⟵ {موز} $\quad c = ۰/۶۶$

(۵) {پرتقال} ⟵ {سیب و موز} $\quad c = ۰/۶۶$

( ۵

(52.

single link و در single link با پیدا کردن دو خوشه که کمترین فاصله را دارند انتخاب کرده و سطر و ستون ماتریس فواصل را بروز کرده . (خط اصلی شکل است که آنها را در نظر گرفته ایم .

| | $P_1P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|
| $P_1P_2$ | 0 | 0.41 | 0.47 | 0.345 |
| $P_3$ | 0.41 | 0 | 0.44 | 0.85 |
| $P_4$ | 0.47 | 0.44 | 0 | 0.76 |
| $P_5$ | 0.345 | 0.85 | 0.76 | 0 |

① کمترین فاصله در ماتریس $P_1P_2 = 0.1$ است پس داریم : انتخاب اولین خوشه بصورت $P_1P_2$ .

② حال باید ماتریس را بروز کنیم داریم : $d(P_1P_2, P_3) = \min(d(P_1,P_3), d(P_2,P_3)) = 0.41$

$d(P_1P_2, P_4) = \min(d(P_1,P_4), d(P_2,P_4)) = 0.47$

$d(P_1P_2, P_5) = \min(d(P_1,P_5), d(P_2,P_5)) = 0.345$

| | $P_1P_2P_5$ | $P_3$ | $P_4$ |
|---|---|---|---|
| $P_1P_2P_5$ | 0 | 0.41 | 0.47 |
| $P_3$ | 0.41 | 0 | 0.44 |
| $P_4$ | 0.47 | 0.44 | 0 |

③ کمترین فاصله $P_1P_2$ با $P_5$ برابر $0.345$ است پس گروهبندی اینگونه باید انتخاب شود .

④ حال ماتریس را بروز رسانی می کنیم :

$d(P_1P_2P_5, P_3) = \min(d(P_1P_2,P_3), d(P_5,P_3)) = 0.41$

$d(P_1P_2P_5, P_4) = \min(d(P_1P_2,P_4), d(P_5,P_4)) = 0.44$

| | $P_1P_2P_5P_3$ | $P_4$ |
|---|---|---|
| $P_1P_2P_5P_3$ | 0 | 0.44 |
| $P_4$ | 0.44 | 0 |

⑤ کمترین فاصله برابر $0.41$ است پس دایره حول $P_3$ انتخاب می شود .

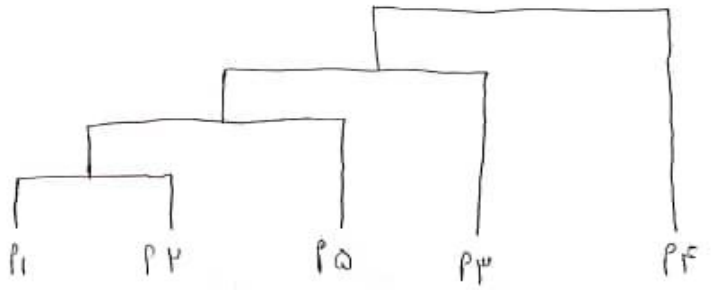
⑥ ماتریس را بروز می کنیم :

$d(P_1P_2P_5P_3, P_4) = \min(d(P_1P_2P_5,P_4), d(P_3,P_4)) = 0.44$

⑦ در نهایت $P_4$ خوشه می شود پس داریم :

$P_1P_2P_5P_3P_4$

دندروگرام نهایی بصورت زیر است :



complete link : برکس single link هنگام آپدیت کردن ماتریس به جای min از max استفاده می‌کنیم.

(۱) کمترین فاصله برای $P_1P_2$ است با مقدار ۰/۱

(۲) بروز کردن ماتریس :

| | $P_1P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|
| $P_1P_2$ | ۰ | ۰/۶۴ | ۰/۵۵ | ۰/۹۸ |
| $P_3$ | ۰/۶۴ | ۰ | ۰/۴۴ | ۰/۸۵ |
| $P_4$ | ۰/۵۵ | ۰/۴۴ | ۰ | ۰/۸۶ |
| $P_5$ | ۰/۹۸ | ۰/۸۵ | ۰/۸۶ | ۰ |

(۳) کمترین مقدار $P_3P_4$ با مقدار ۰/۴۴ است

(۴) بروز کردن ماتریس :

| | $P_1P_2$ | $P_3P_4$ | $P_5$ |
|---|---|---|---|
| $P_1P_2$ | ۰ | ۰/۶۴ | ۰/۹۸ |
| $P_3P_4$ | ۰/۶۴ | ۰ | ۰/۸۵ |
| $P_5$ | ۰/۹۸ | ۰/۸۵ | ۰ |

(۵) کمترین فاصله $P_3P_4$-$P_1P_2$ است با ۰/۶۴

(۶) بروز کردن ماتریس

| | $P_1P_2P_3P_4$ | $P_5$ |
|---|---|---|
| $P_1P_2P_3P_4$ | ۰ | ۰/۹۸ |
| $P_5$ | ۰/۹۸ | ۰ |

(۷) در آخر $P_5$ انتخاب می‌شود داریم:
$P_1P_2P_3P_4P_5$