

Bem-Vindx à Comunidade DS

Live #037

DS em Clusterização
Ciclo 08 -
Como os modelos funcionam?

Agenda:

- **Live #037 - DS em Clusterização - Ciclo 08 - Explicação dos modelos de clusterização - Parte II**
 - Hierarchical Clustering
 - DBScan
- **Live #038 - DS em Clusterização - Ciclo 09 - Aplicação dos modelos de clusterização**
 - Aplicação dos modelos.
 - Avaliação das performance.

Live #037

Ciclo 08 -

Como os modelos funcionam?

Conteúdo da Live #037

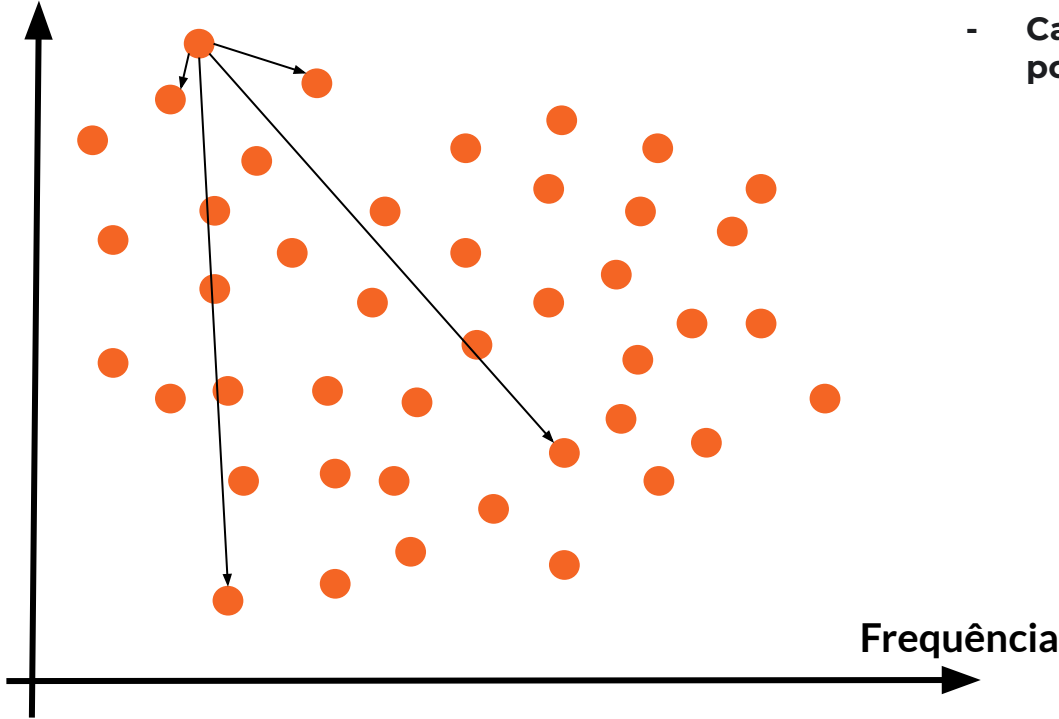
1. Como funcionam os modelos:

- a. Hierarchical Clustering
- b. DBSCAN (Density-Based Spatial Clustering Application Noise)

O funcionamento do Hierarchical Clustering

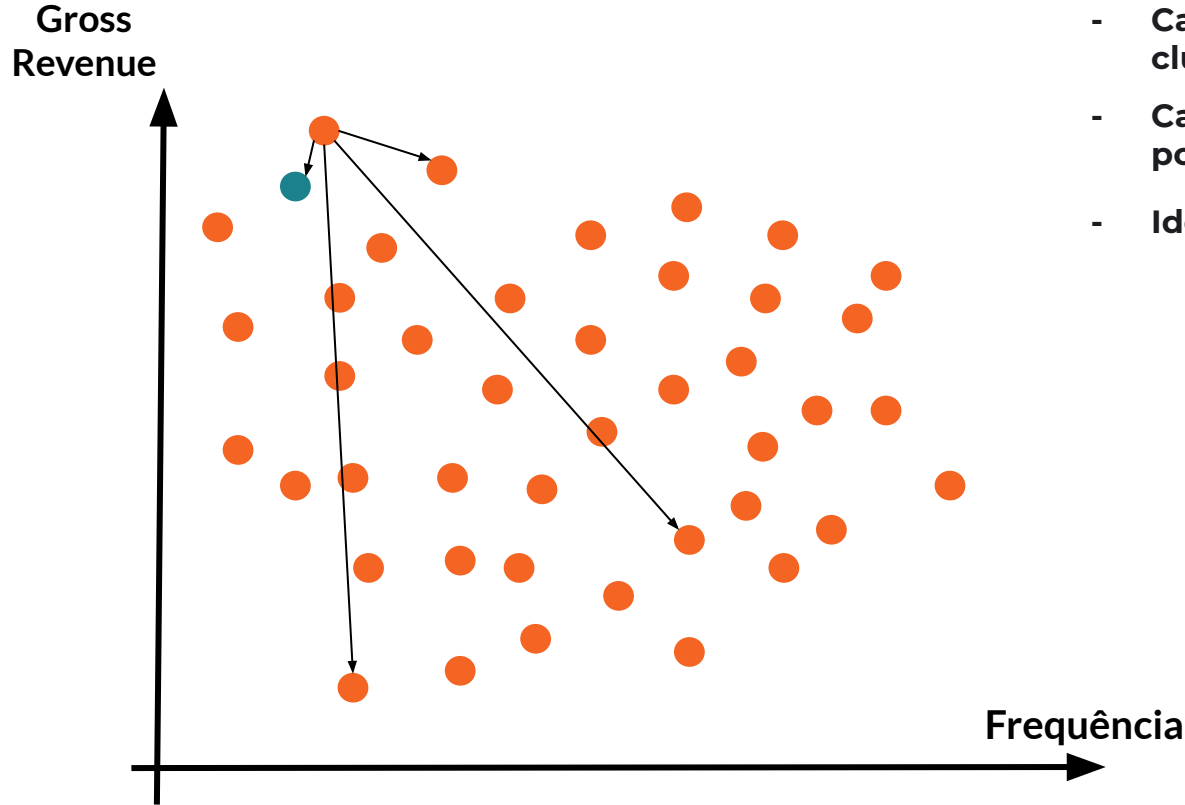
O funcionamento do H-Clustering

Gross
Revenue



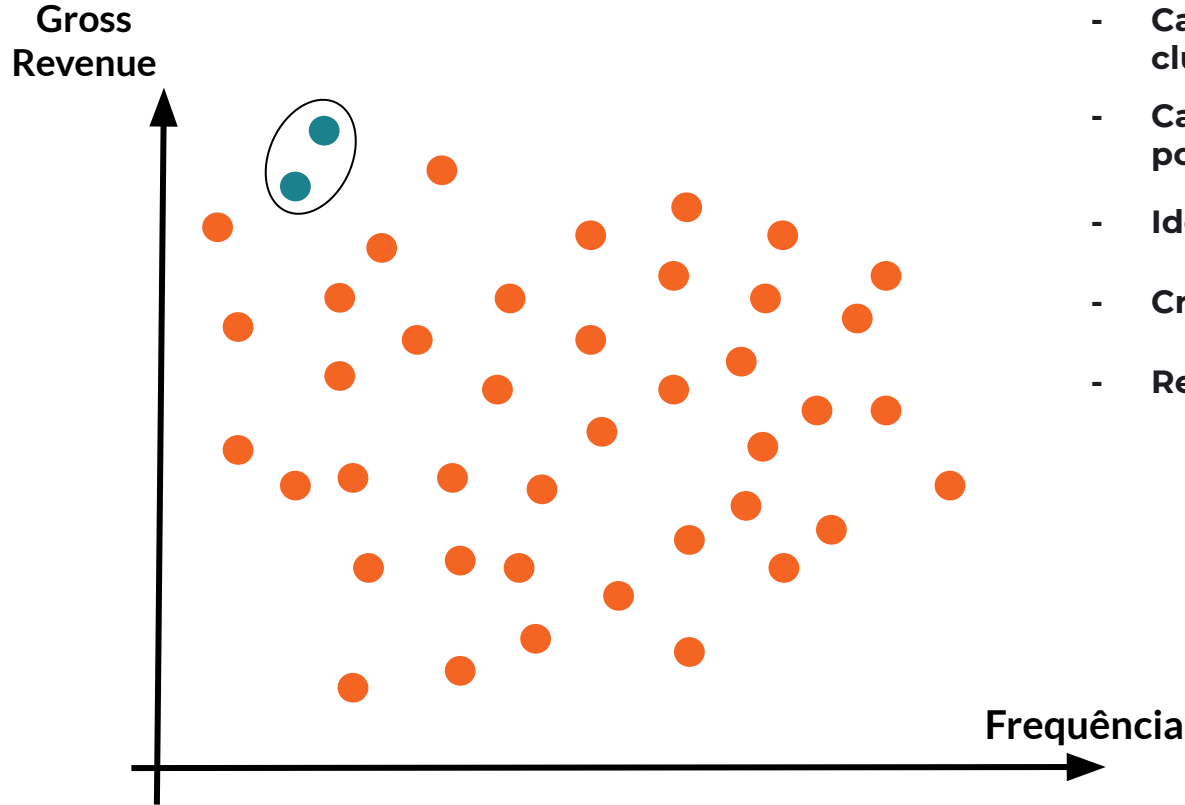
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.

O funcionamento do H-Clustering



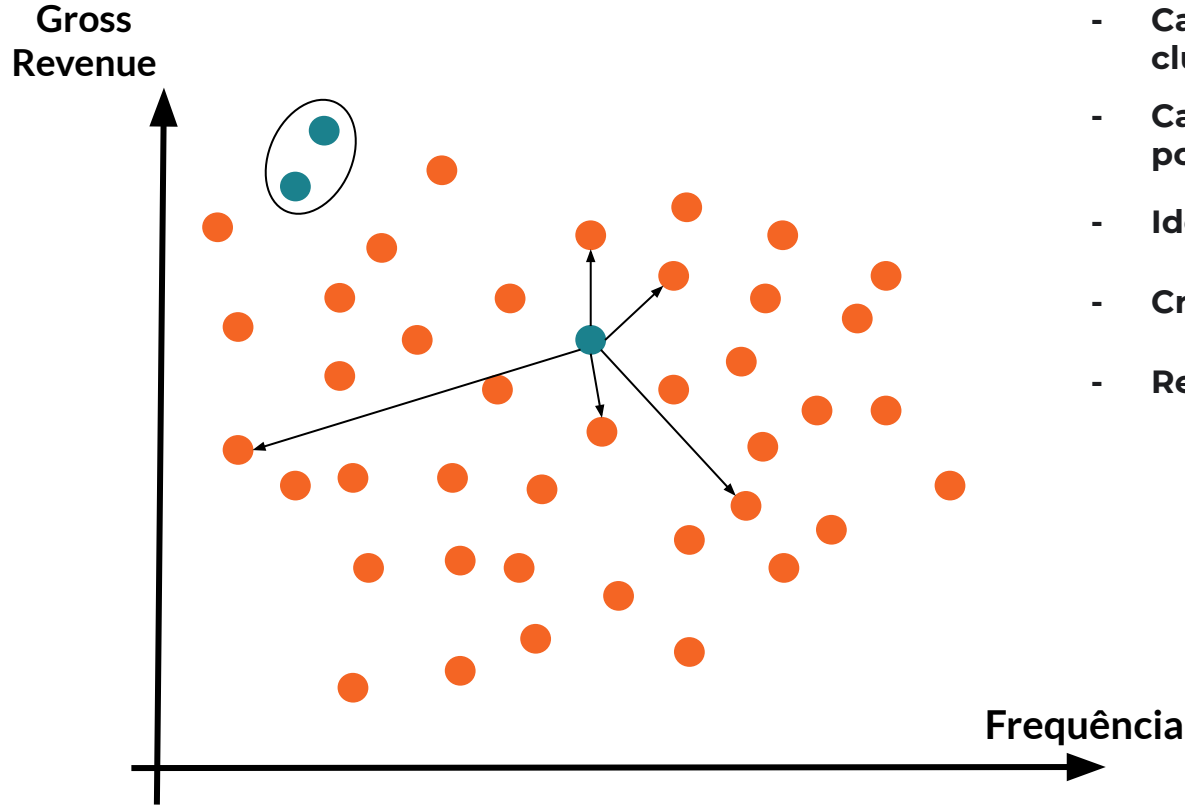
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.

O funcionamento do H-Clustering



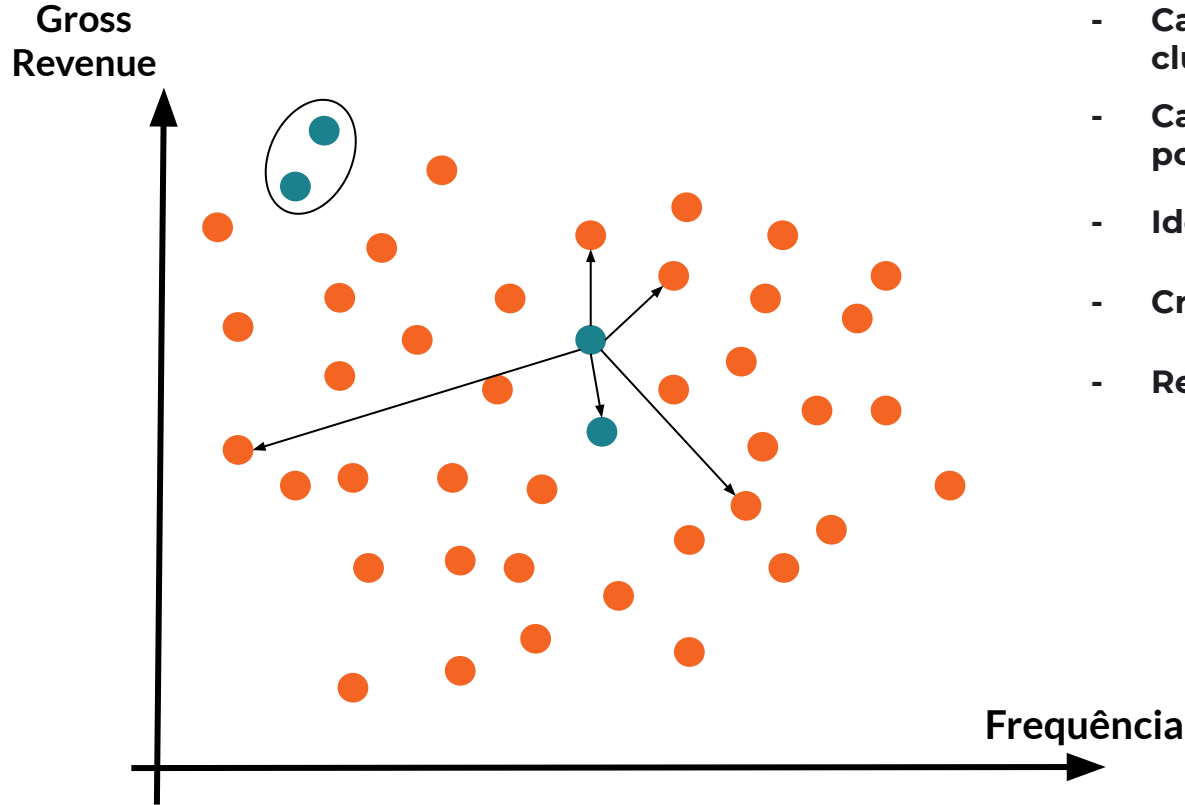
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



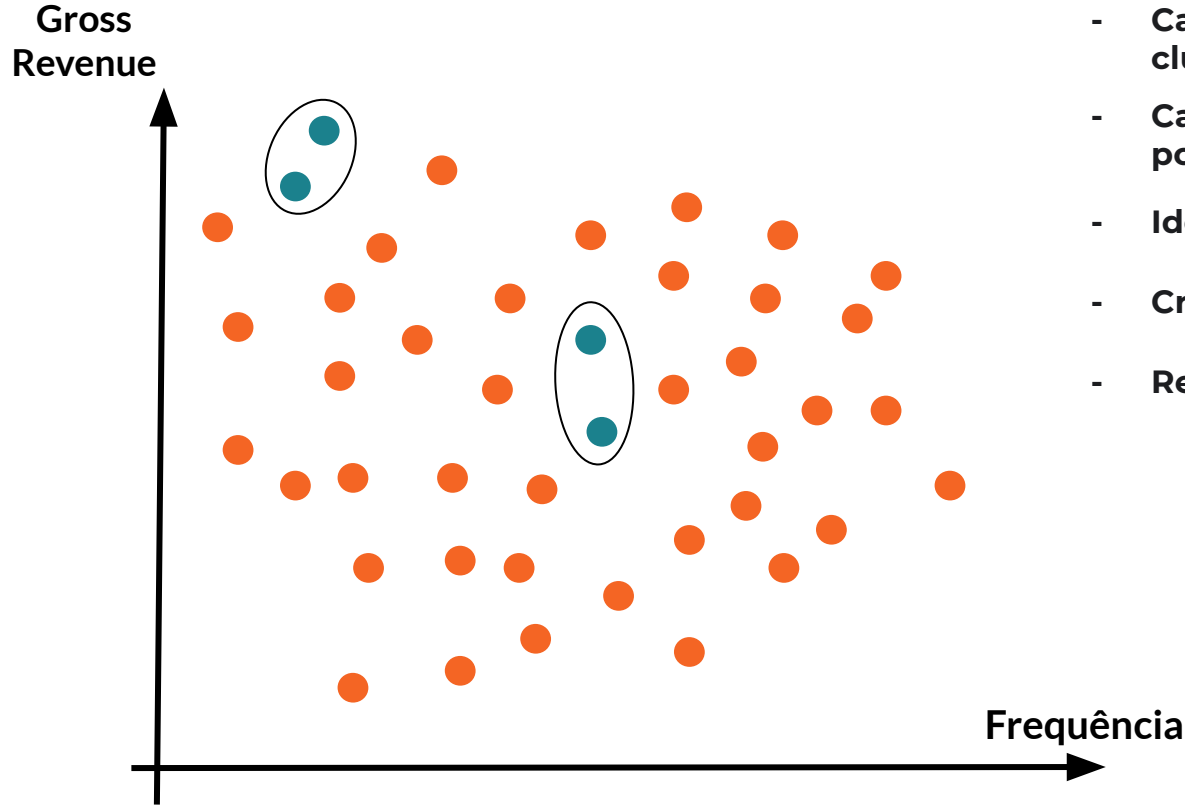
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



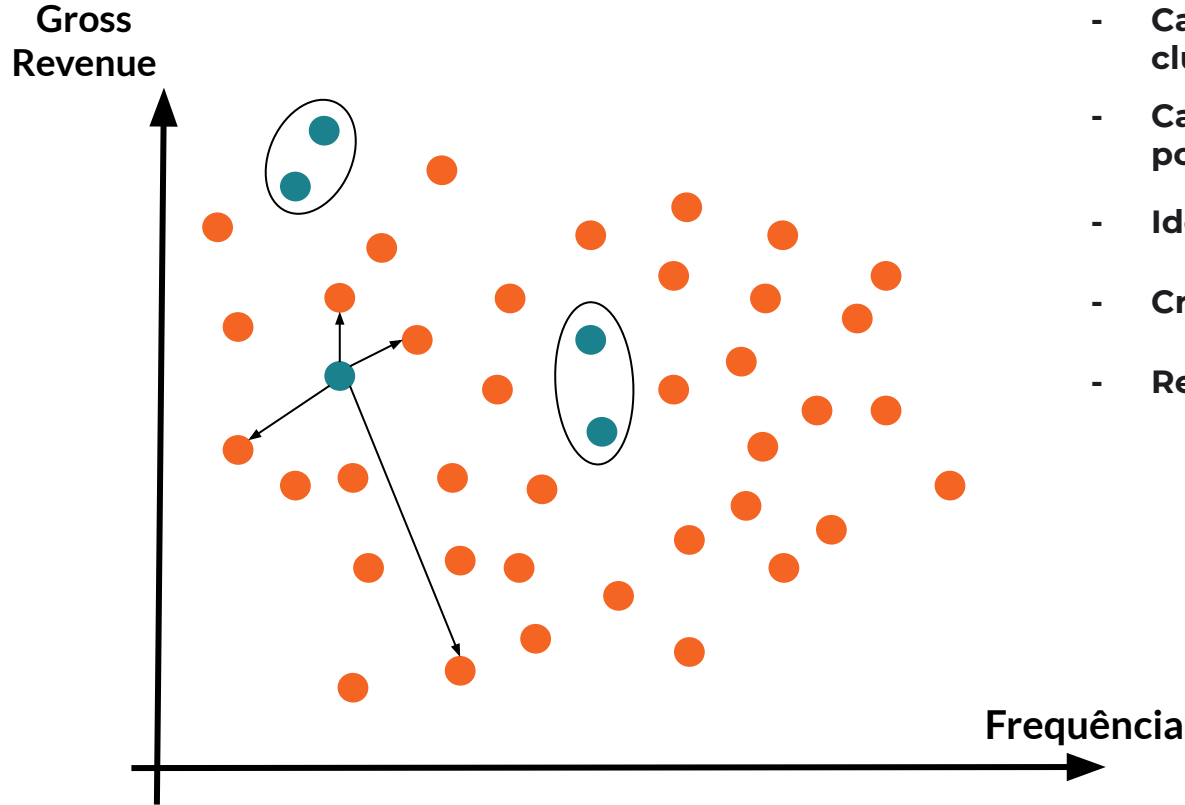
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



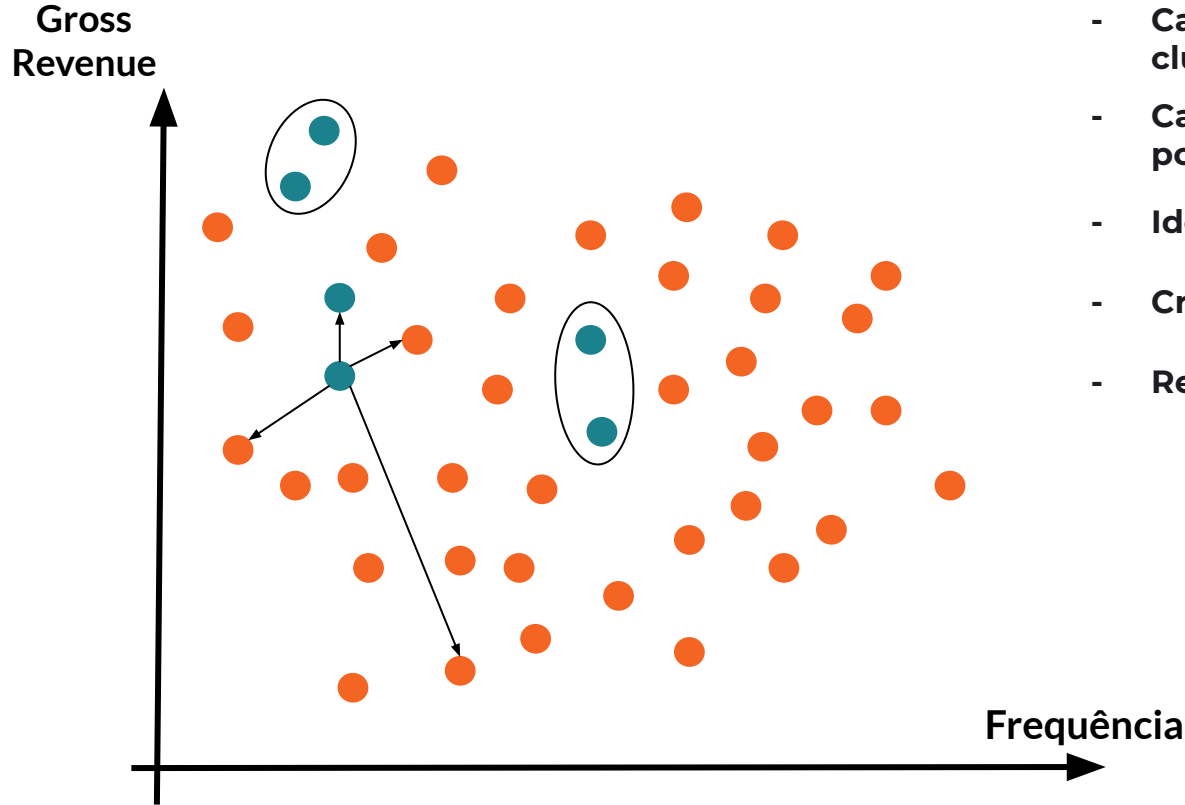
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



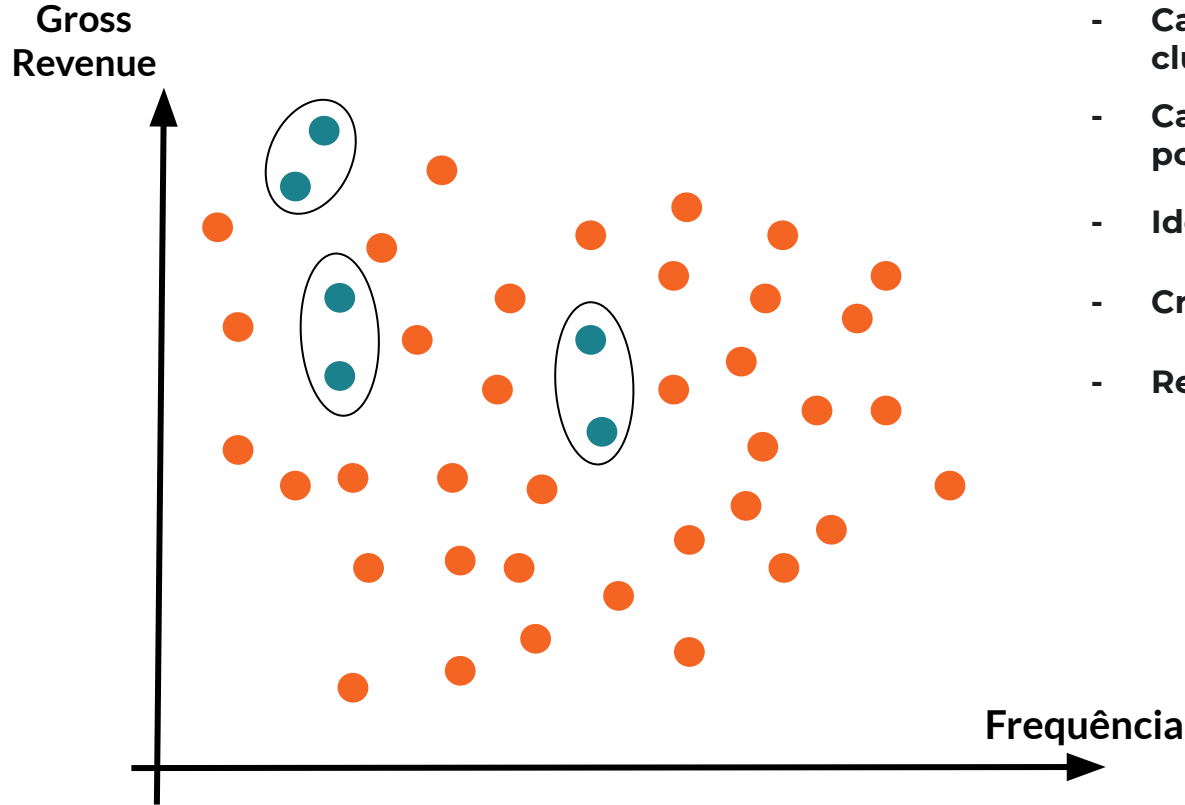
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



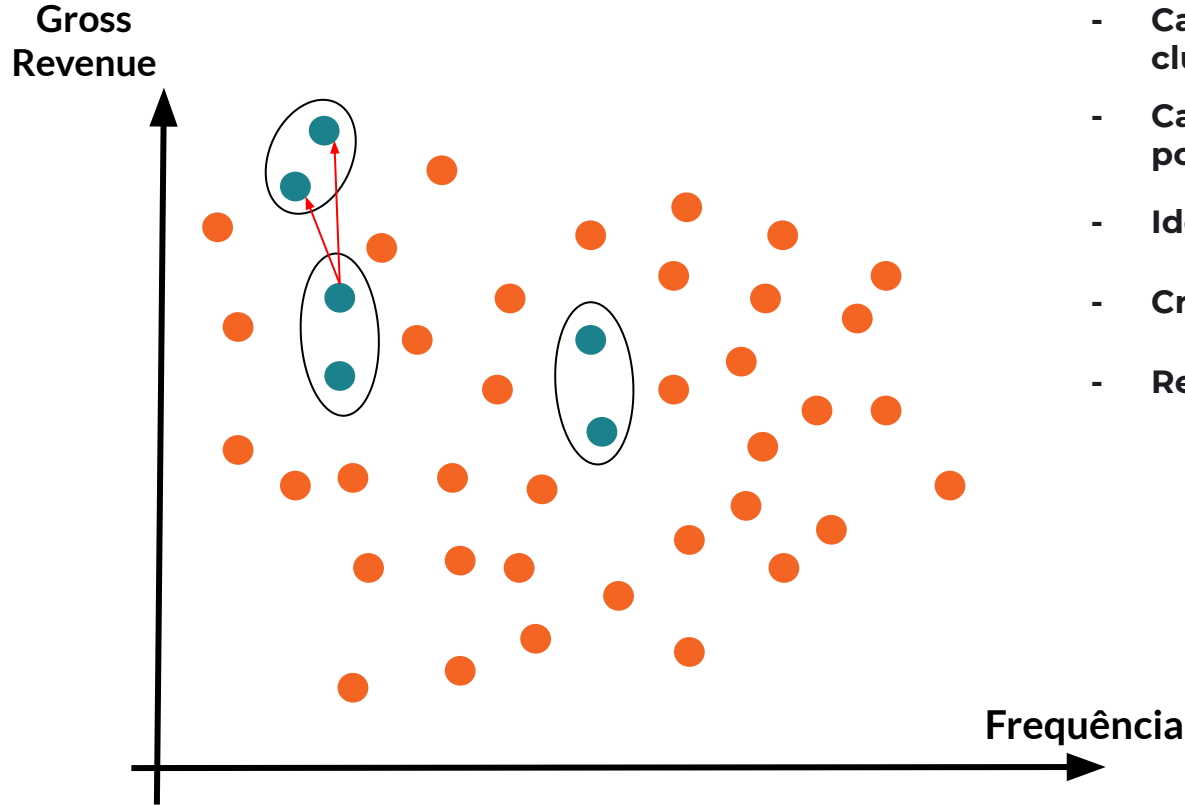
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering

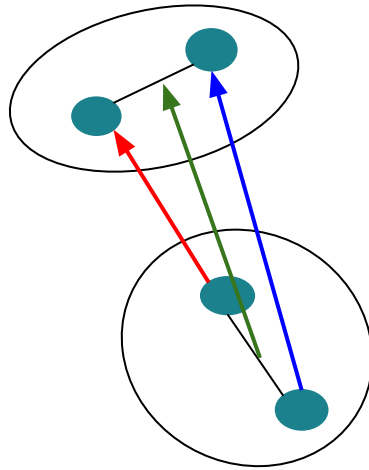


- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

Critério de ligação (Linkage Criteria)

O funcionamento do H-Clustering

Gross
Revenue



Entre os pontos mais próximos de diferentes clusters
(Single-Linkage)

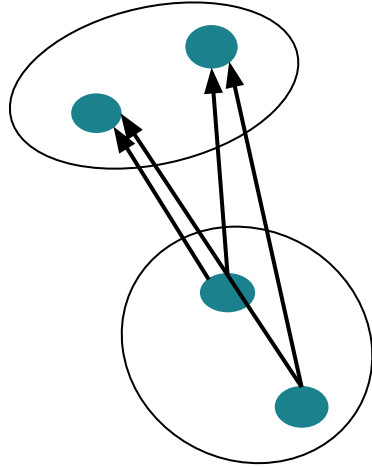
Entre os pontos mais distantes de diferentes clusters
(Complete-Linkage)

Entre os pontos médios de diferentes clusters
(Average-Linkage)

Frequência

O funcionamento do H-Clustering

Gross
Revenue



Entre os pontos mais próximos de diferentes clusters
(Single-Linkage)

Entre os pontos mais distantes de diferentes clusters
(Complete-Linkage)

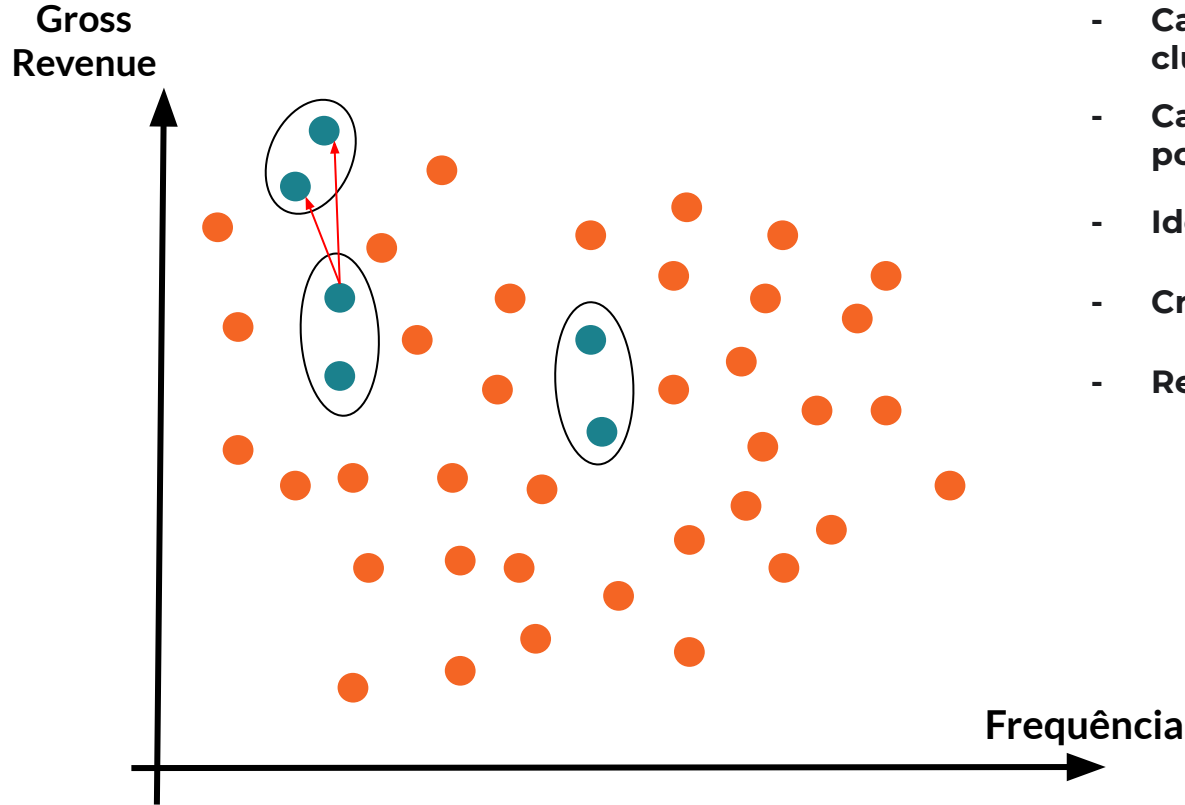
Entre os pontos médios de diferentes clusters
(Average-Linkage)

Distância Média entre todos pontos de diferentes clusters
(Group-Linkage)

Distância Quadrática entre todos pontos de diferentes clusters
(Ward's Method)

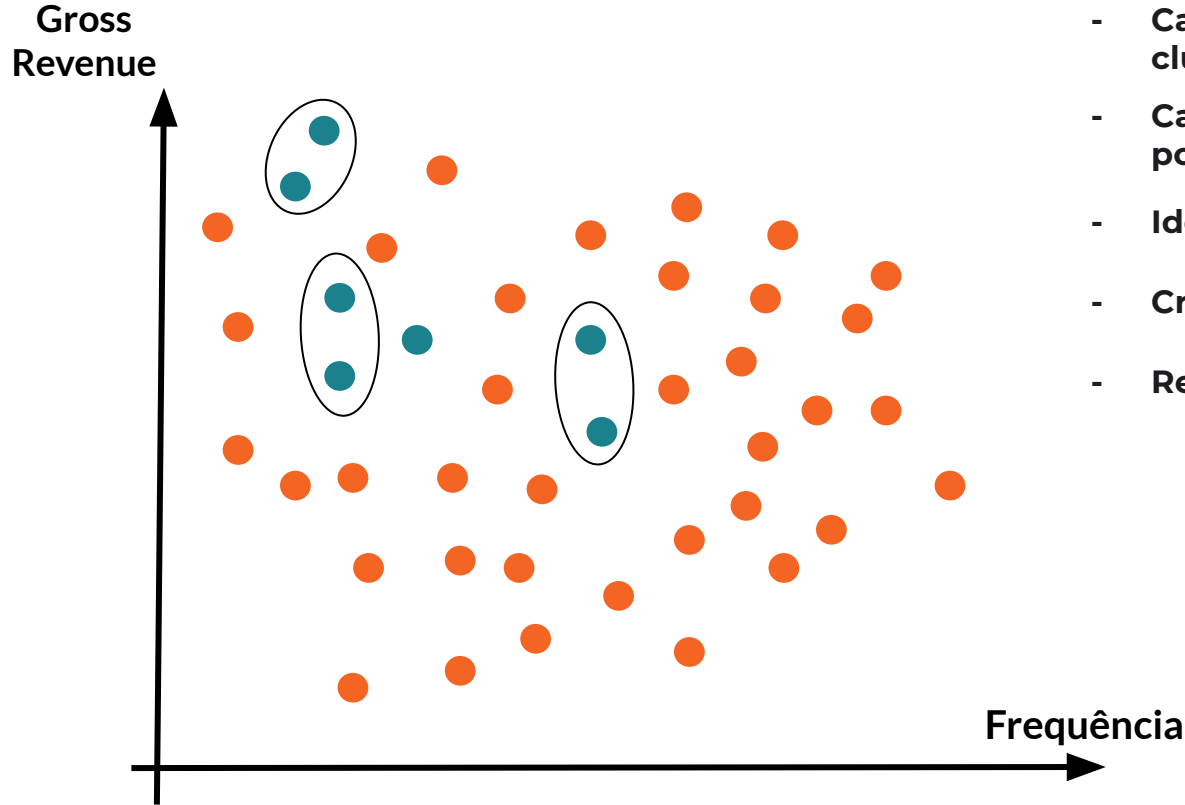
Frequência

O funcionamento do H-Clustering



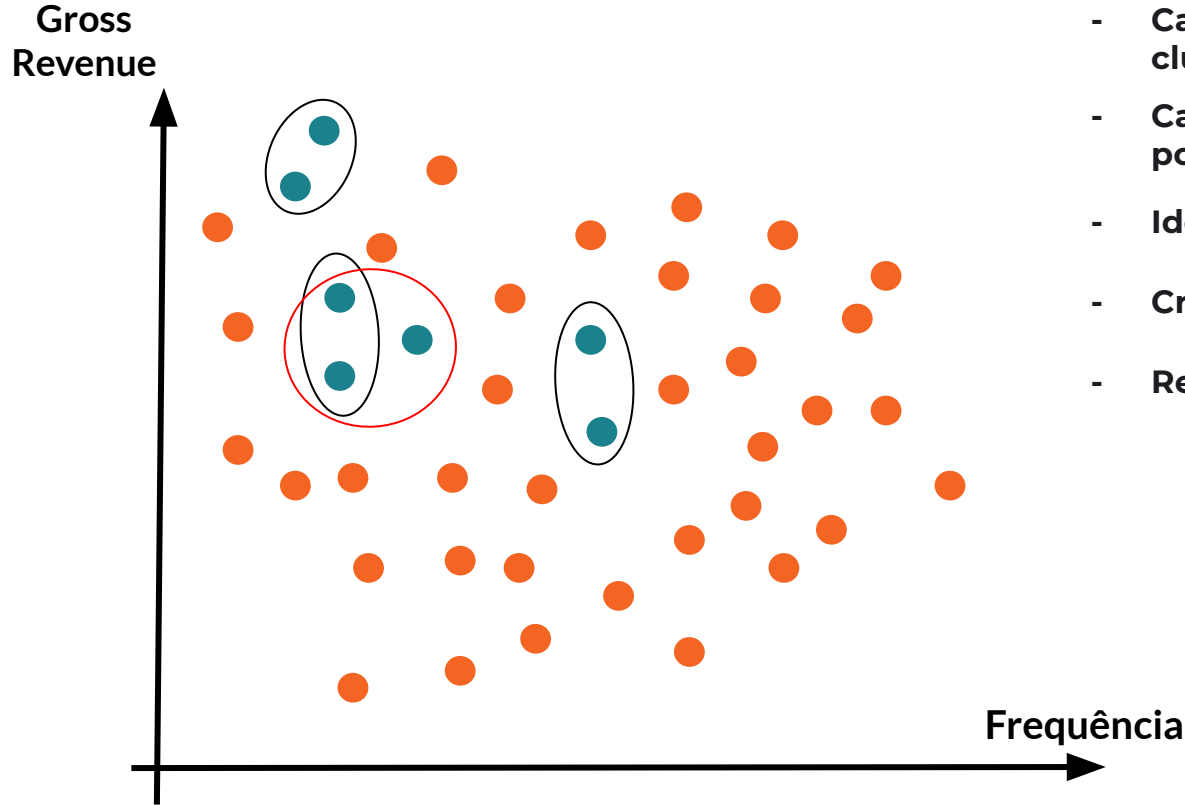
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



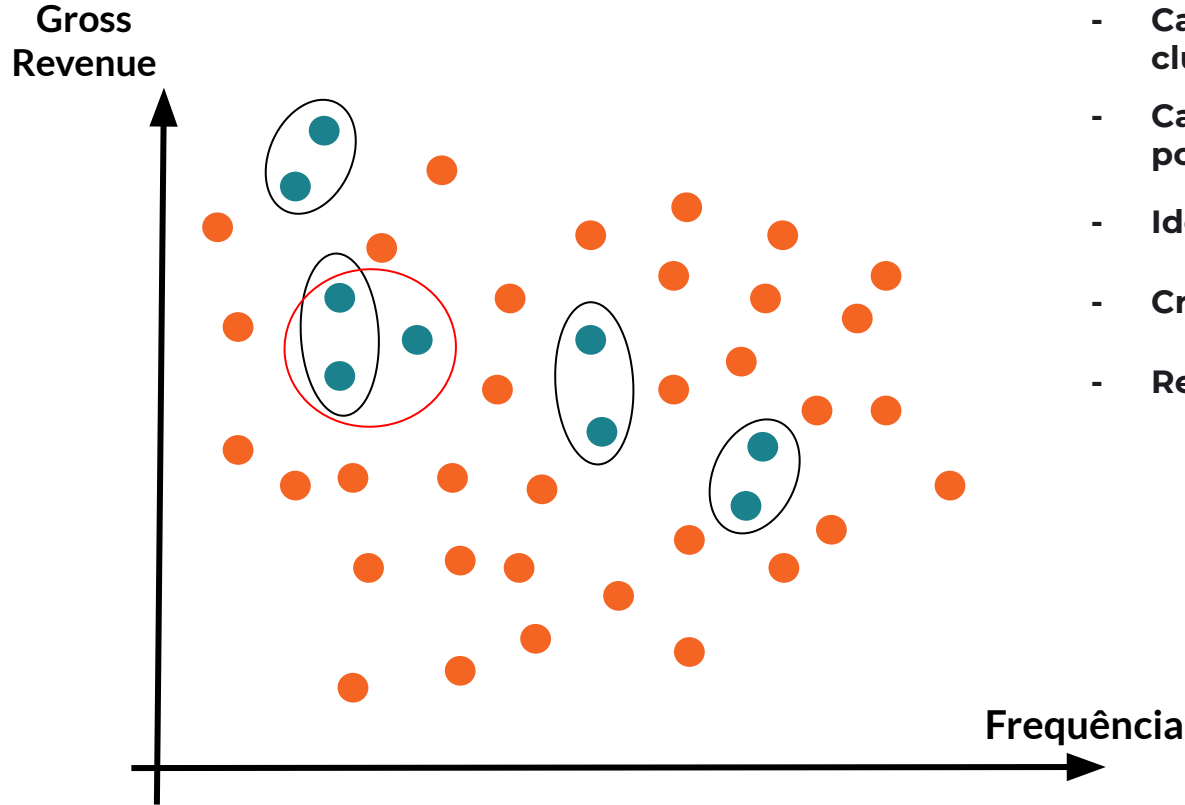
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



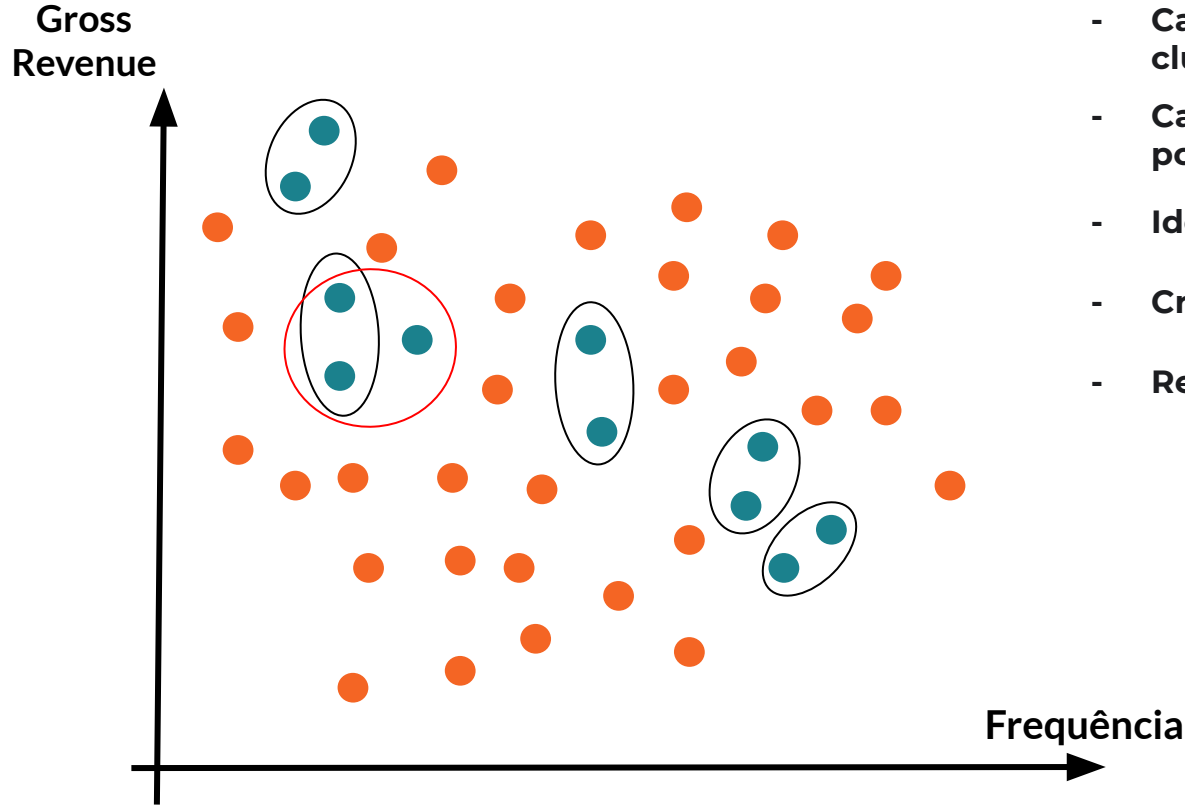
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



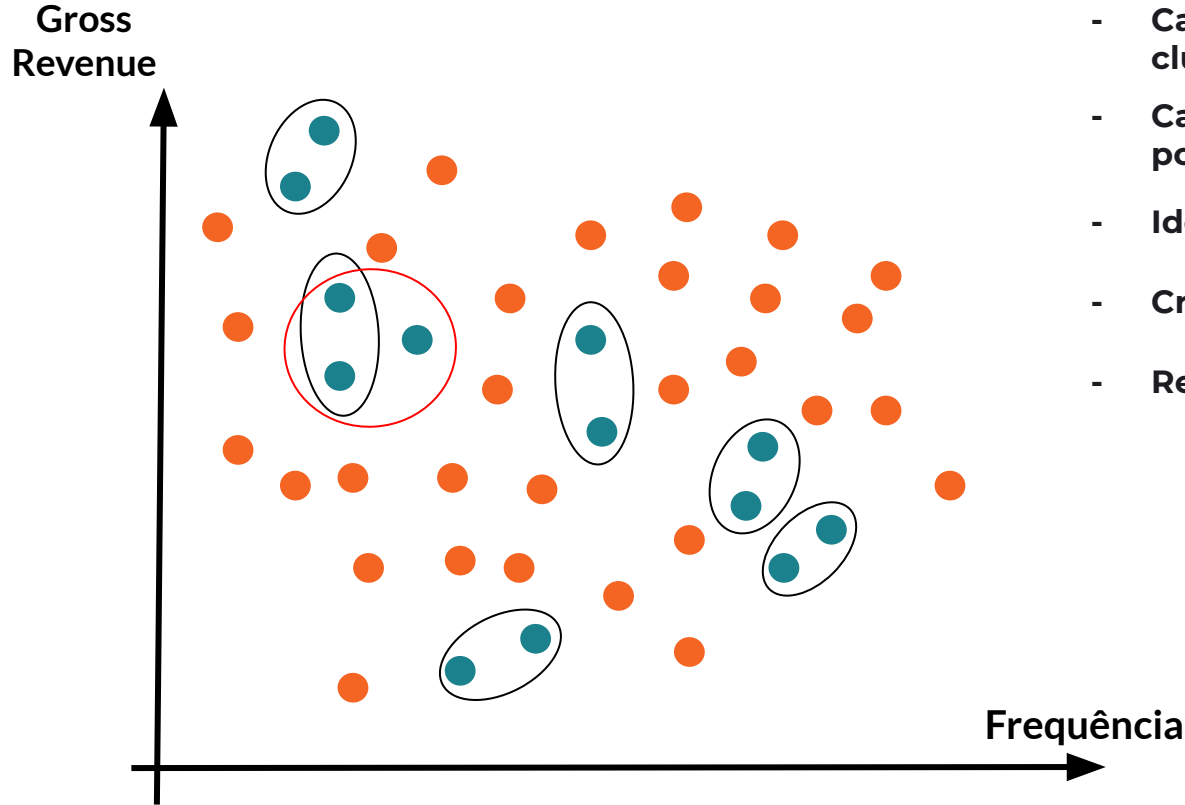
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



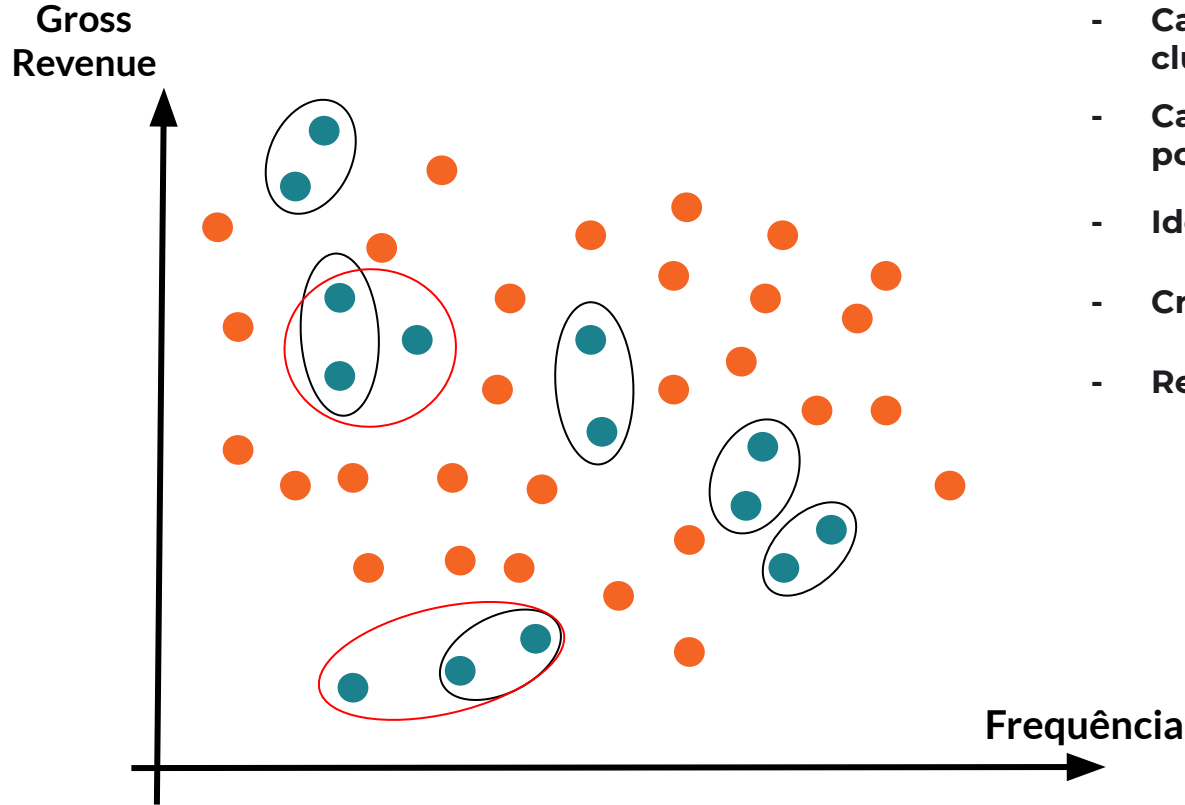
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



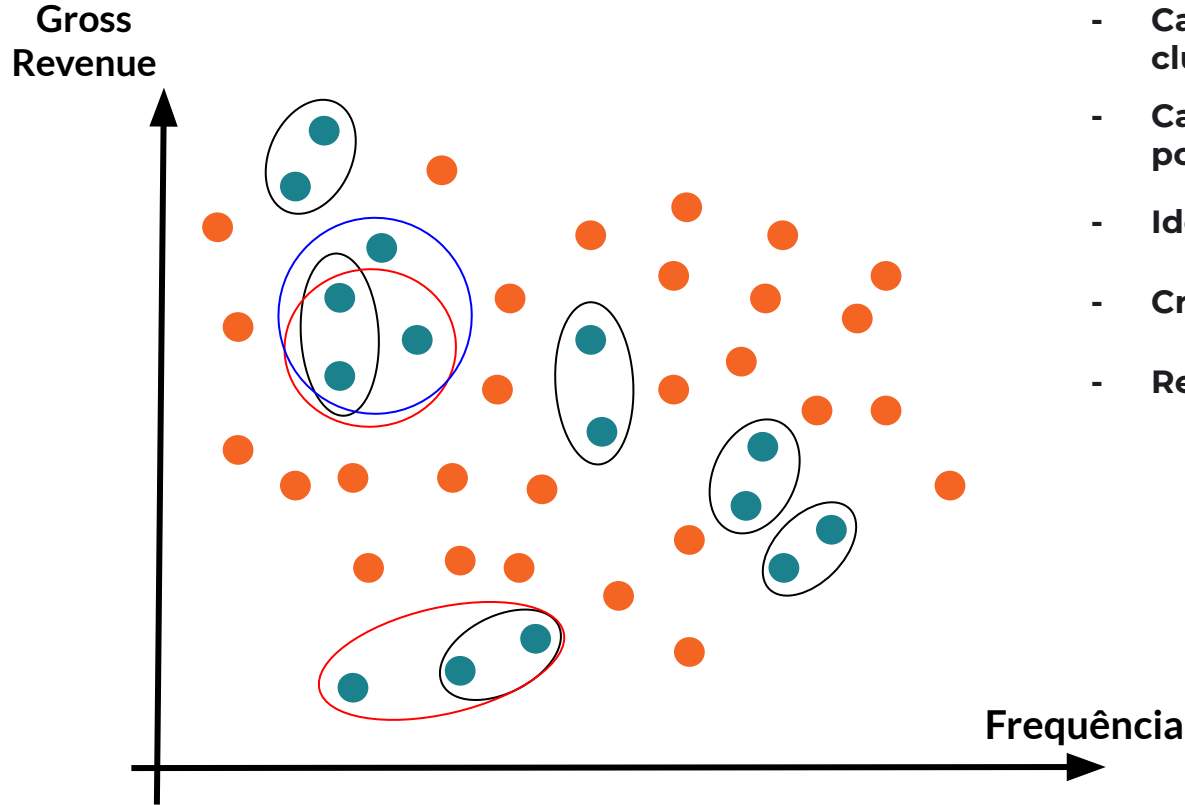
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



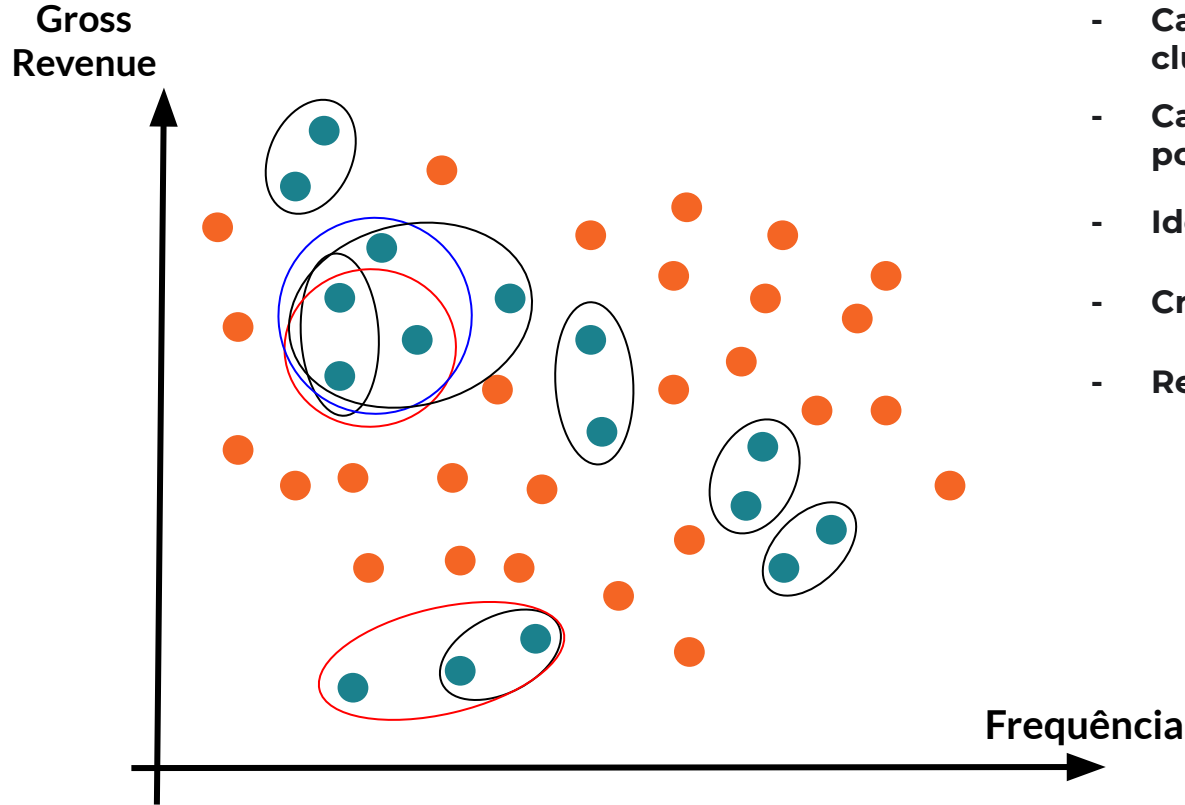
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



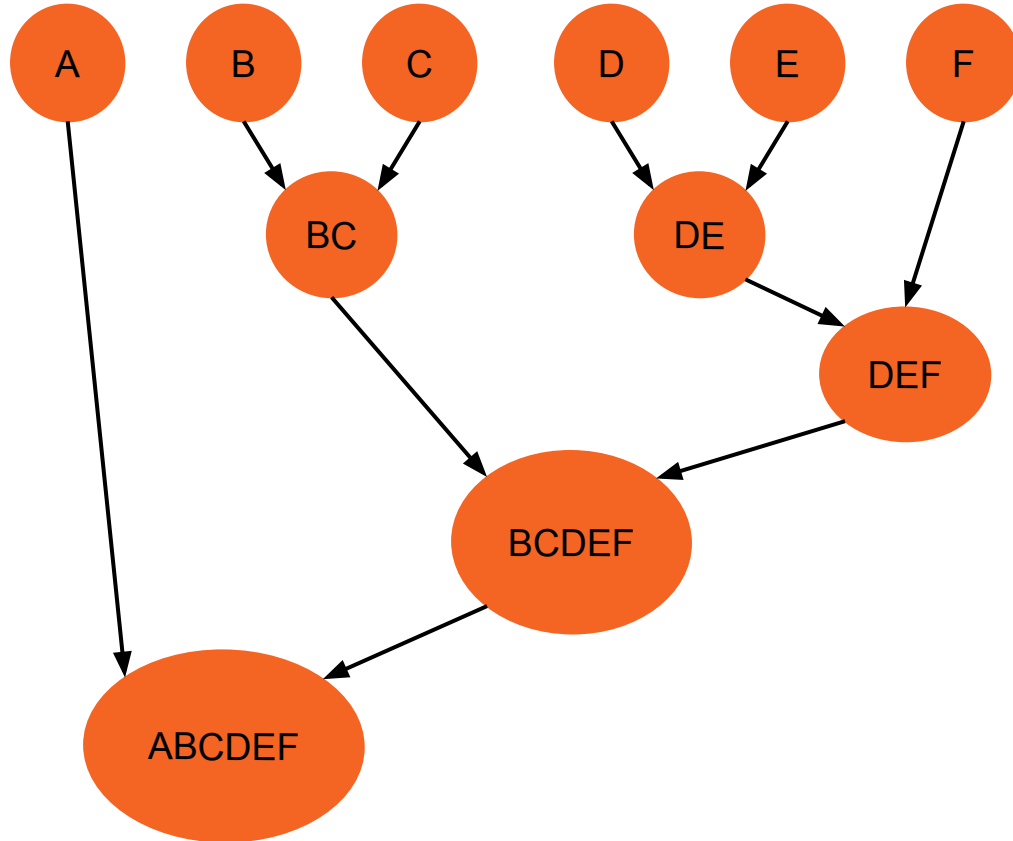
- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering



- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

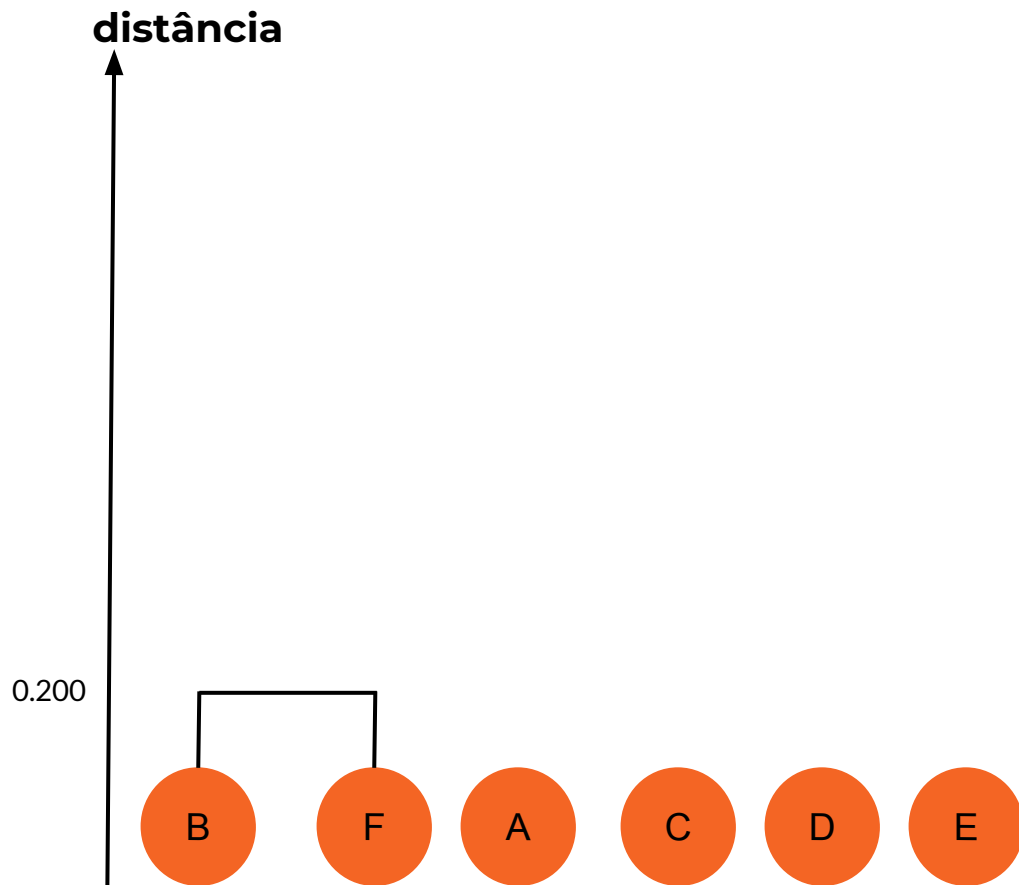
O funcionamento do H-Clustering



- Cada ponto é considerado um cluster.
- Calcula a distância entre todos os pontos do espaço.
- Identifica o vizinho mais próximo.
- Crie um novo cluster.
- Repete o processo.

O funcionamento do H-Clustering

	A	B	C	D	E	F
A	0.000	0.500	0.428	1.000	0.250	0.625
B	0.500	0.000	0.714	0.833	0.667	0.200
C	0.428	0.714	0.000	1.000	0.428	0.667
D	1.000	0.833	1.000	0.000	0.800	0.800
E	0.250	0.667	0.428	0.800	0.000	0.777
F	0.625	0.200	0.667	0.857	0.750	0.000



O funcionamento do H-Clustering

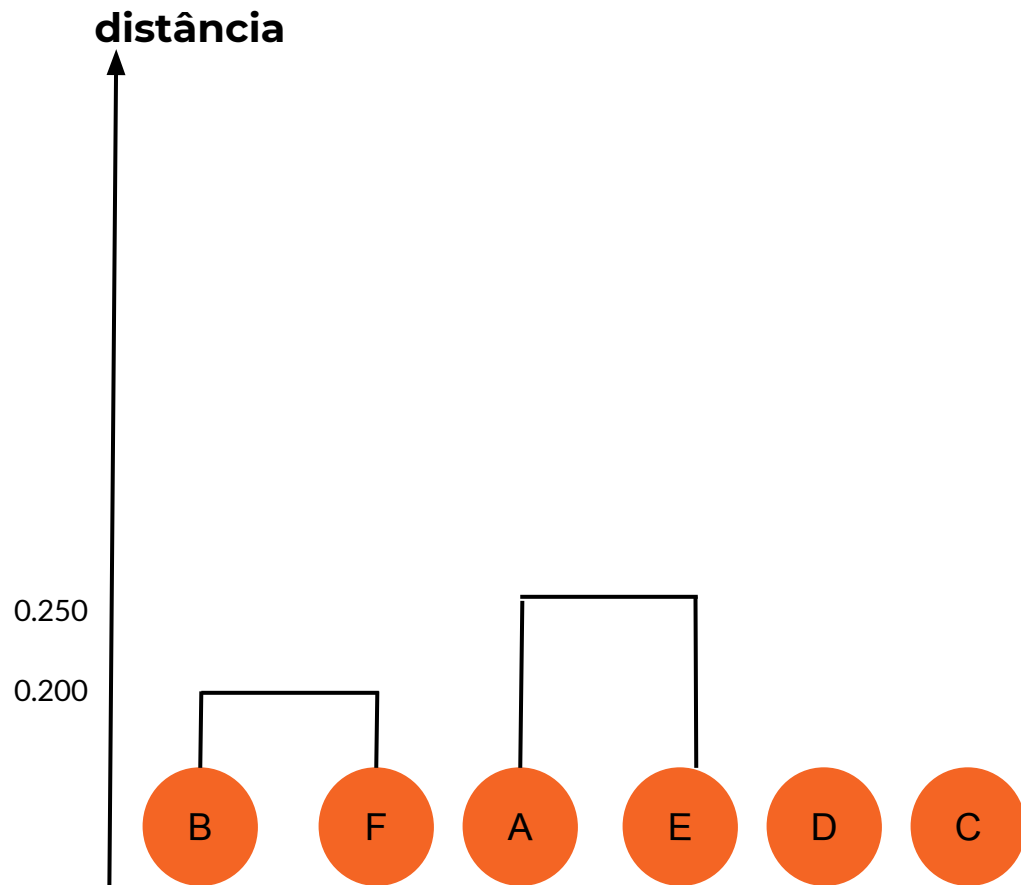
	A	B	C	D	E	F
A	0.000	0.500	0.428	1.000	0.250	0.625
B	0.500	0.000	0.714	0.833	0.667	0.200
C	0.428	0.714	0.000	1.000	0.428	0.667
D	1.000	0.833	1.000	0.000	0.800	0.800
E	0.250	0.667	0.428	0.800	0.000	0.777
F	0.625	0.200	0.667	0.857	0.777	0.000

	A	B,F	C	D	E
A	0.000	0.625	0.428	1.000	0.250
B,F	0.625	0.000	0.714	0.833	0.777
C	0.428	0.714	0.000	1.000	0.428
D	1.000	0.833	1.000	0.000	0.800
E	0.250	0.777	0.428	0.800	0.000

Complete-linkage

O funcionamento do H-Clustering

	A	B,F	C	D	E
A	0.000	0.625	0.428	1.000	0.250
B,F	0.625	0.000	0.714	0.833	0.777
C	0.428	0.714	0.000	1.000	0.428
D	1.000	0.833	1.000	0.000	0.800
E	0.250	0.777	0.428	0.800	0.000



O funcionamento do H-Clustering

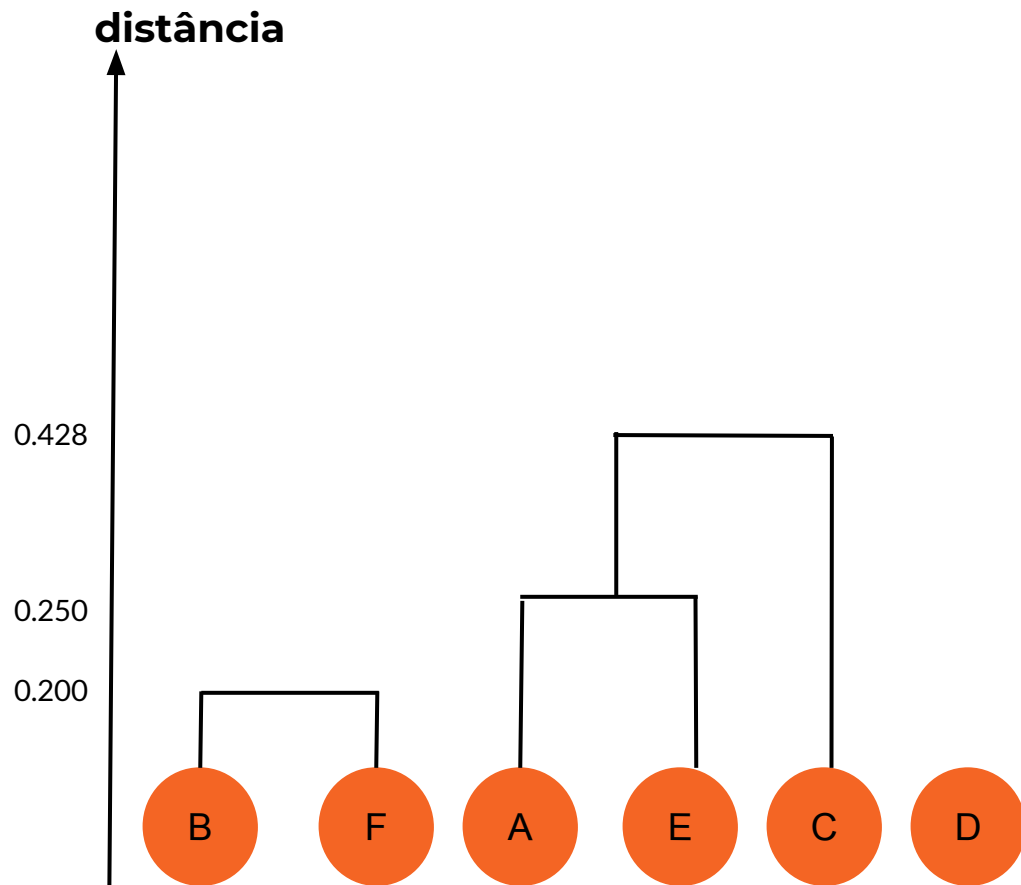
	A	B,F	C	D	E
A	0.000	0.625	0.428	1.000	0.250
B,F	0.625	0.000	0.714	0.833	0.777
C	0.428	0.714	0.000	1.000	0.428
D	1.000	0.833	1.000	0.000	0.800
E	0.250	0.777	0.428	0.800	0.000

	A,E	B,F	C	D
A,E	0.000	0.777	0.428	1.000
B,F	0.777	0.000	0.714	0.833
C	0.428	0.714	0.000	1.000
D	1.000	0.833	1.000	0.000

Complete-linkage

O funcionamento do H-Clustering

	A,E	B,F	C	D
A,E	0.000	0.777	0.428	1.000
B,F	0.777	0.000	0.714	0.833
C	0.428	0.714	0.000	1.000
D	1.000	0.833	1.000	0.000



O funcionamento do H-Clustering

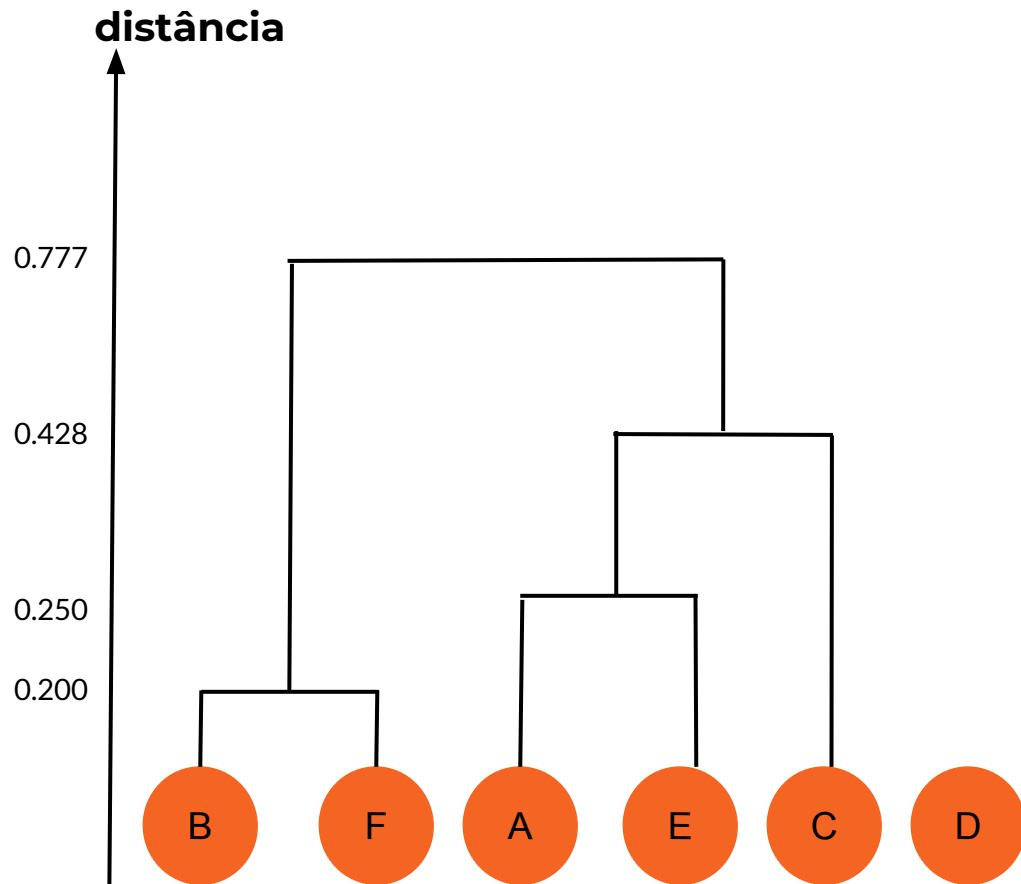
	A,E	B,F	C	D
A,E	0.000	0.777	0.428	1.000
B,F	0.777	0.000	0.714	0.833
C	0.428	0.714	0.000	1.000
D	1.000	0.833	1.000	0.000

	A,E,C	B,F	D
A,E,C	0.000	0.777	1.000
B,F	0.777	0.000	0.833
D	1.000	0.833	0.000

Complete-linkage

O funcionamento do H-Clustering

	A,E,C	B,F	D
A,E,C	0.000	0.777	1.000
B,F	0.777	0.000	0.833
D	1.000	0.833	0.000



O funcionamento do H-Clustering

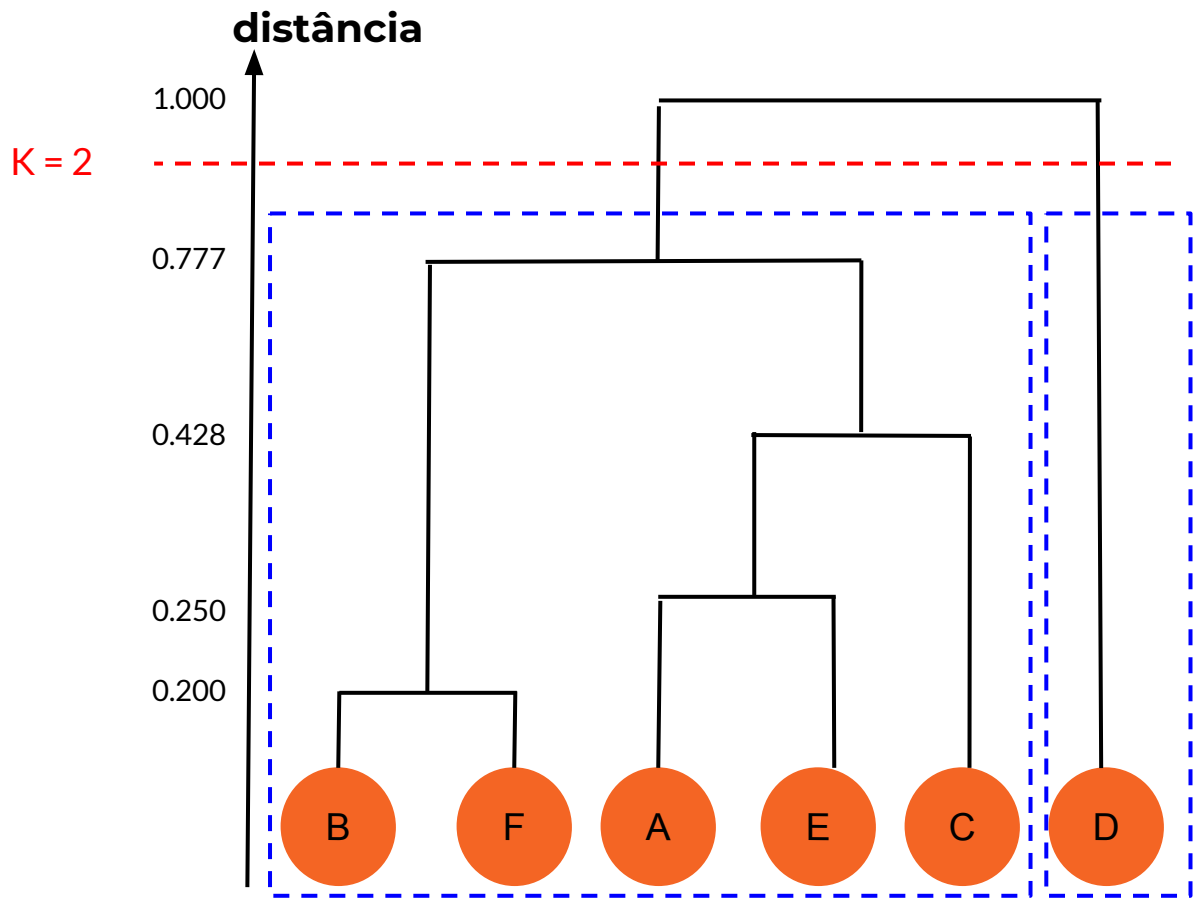
	A,E,C	B,F	D
A,E,C	0.000	0.777	1.000
B,F	0.777	0.000	0.833
D	1.000	0.833	0.000

	A,E,C,B, F	D
A,E,C,B,F	0.000	1.000
D	1.000	0.000

Complete-linkage

O funcionamento do H-Clustering

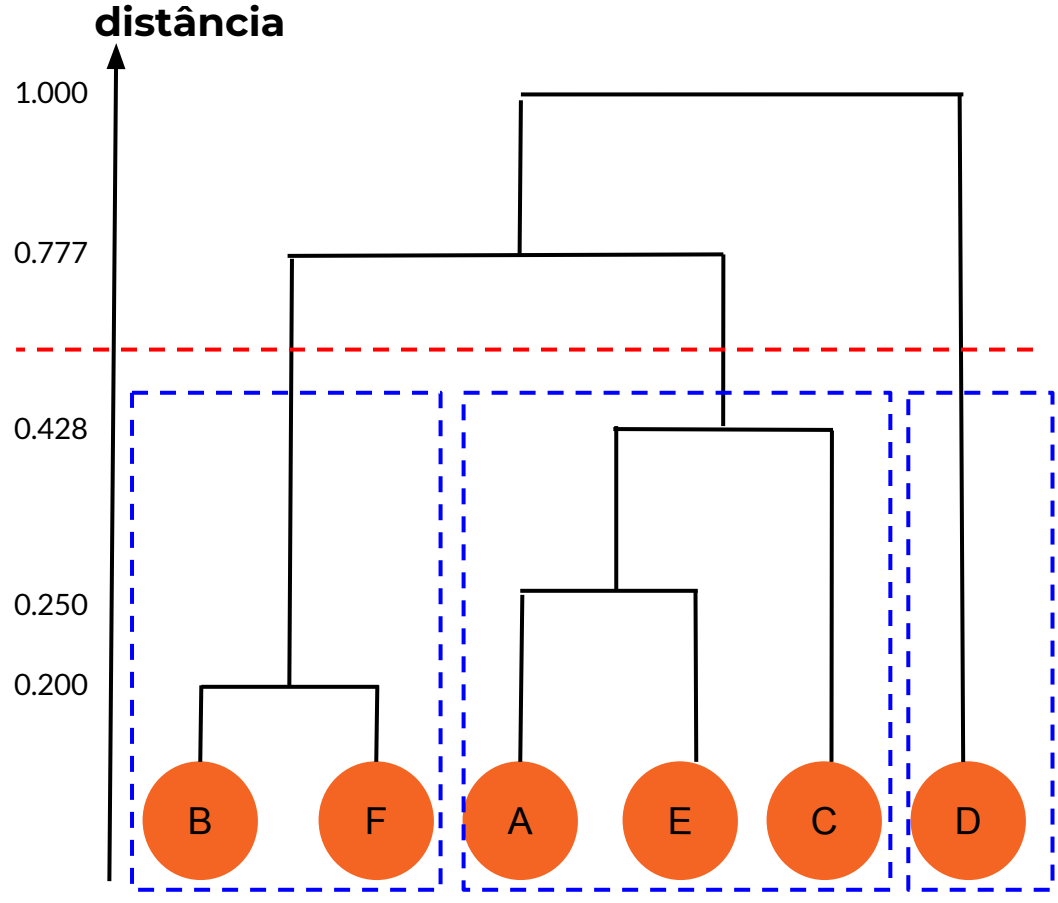
	A,E,C,B, F	D
A,E,C,B,F	0.000	1.000
D	1.000	0.000



O funcionamento do H-Clustering

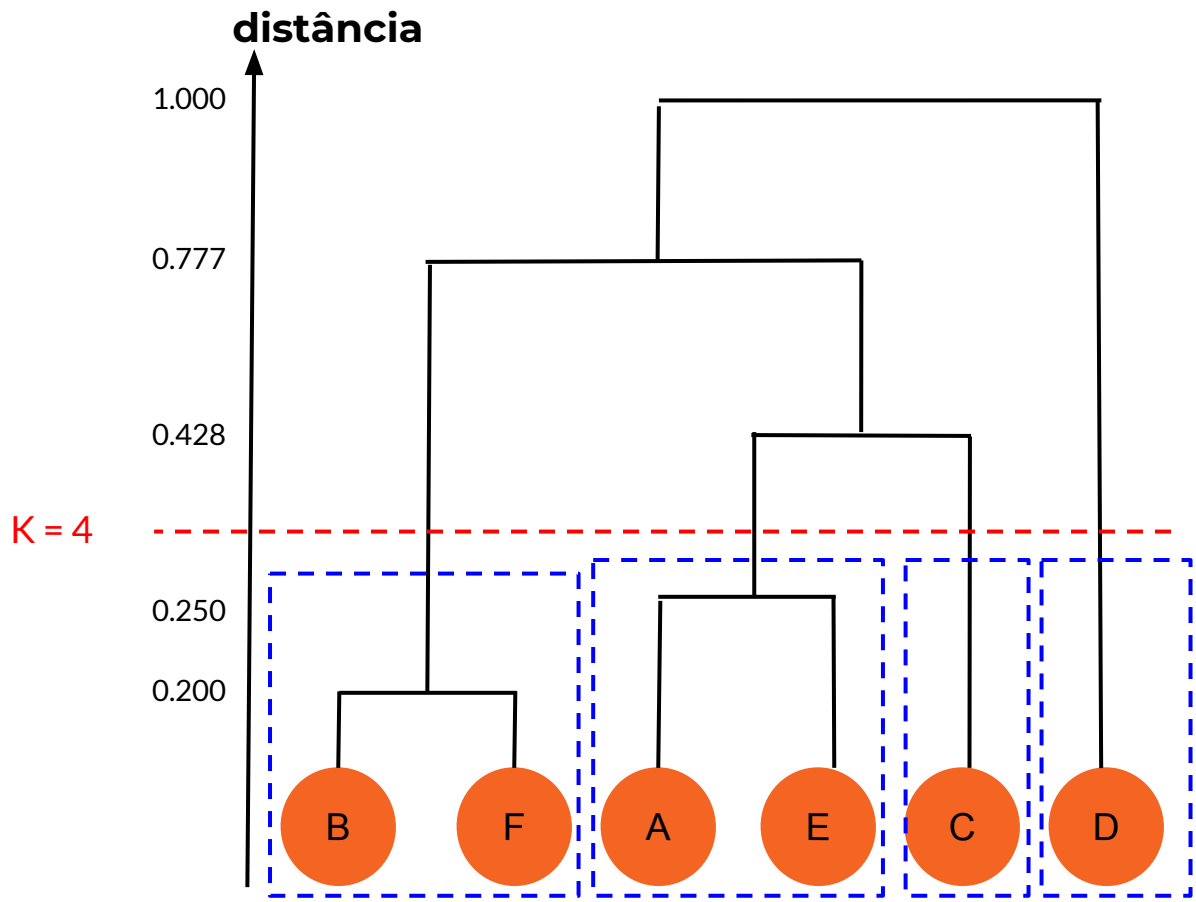
	A,E,C,B, F	D
A,E,C,B,F	0.000	1.000
D	1.000	0.000

K = 3



O funcionamento do H-Clustering

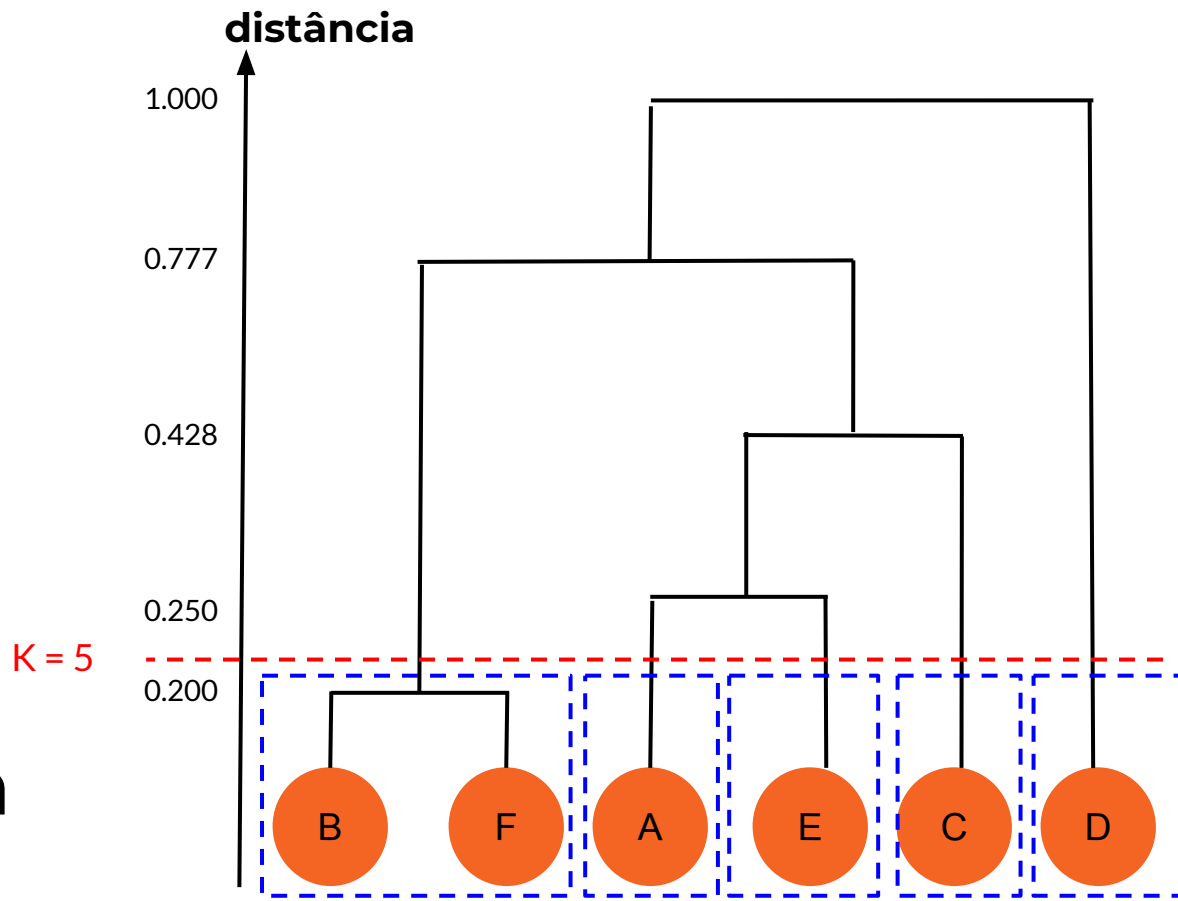
	A,E,C,B, F	D
A,E,C,B,F	0.000	1.000
D	1.000	0.000



O funcionamento do H-Clustering

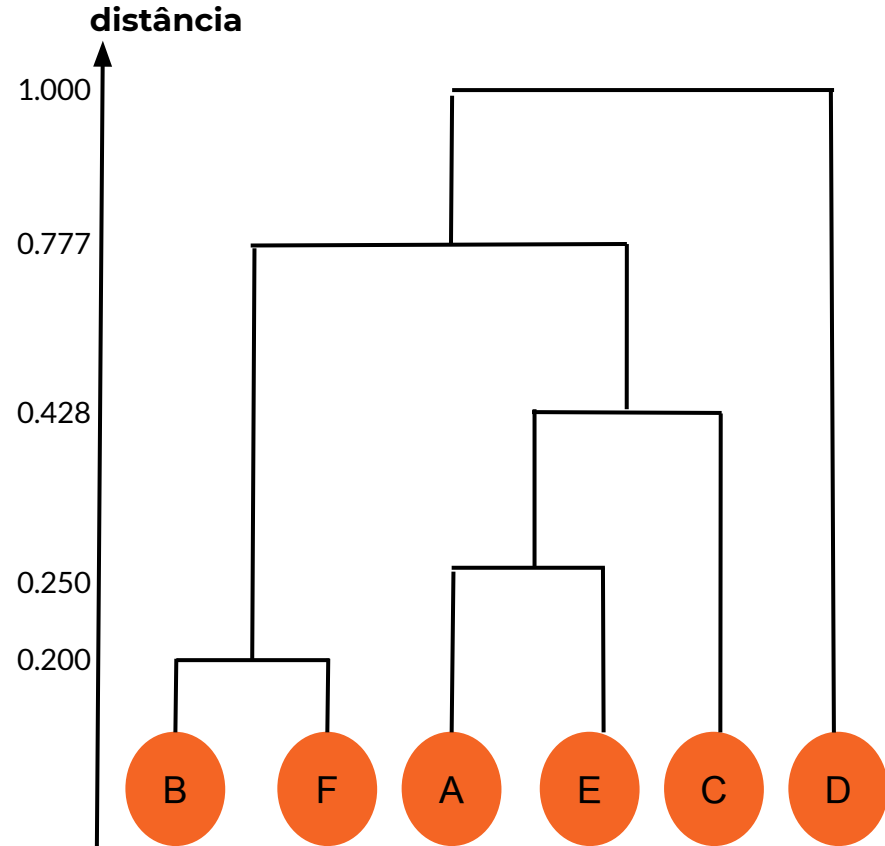
	A,E,C,B,F	D
A,E,C,B,F	0.000	1.000
D	1.000	0.000

Dendograma



3. Características do Hierarchical Clustering

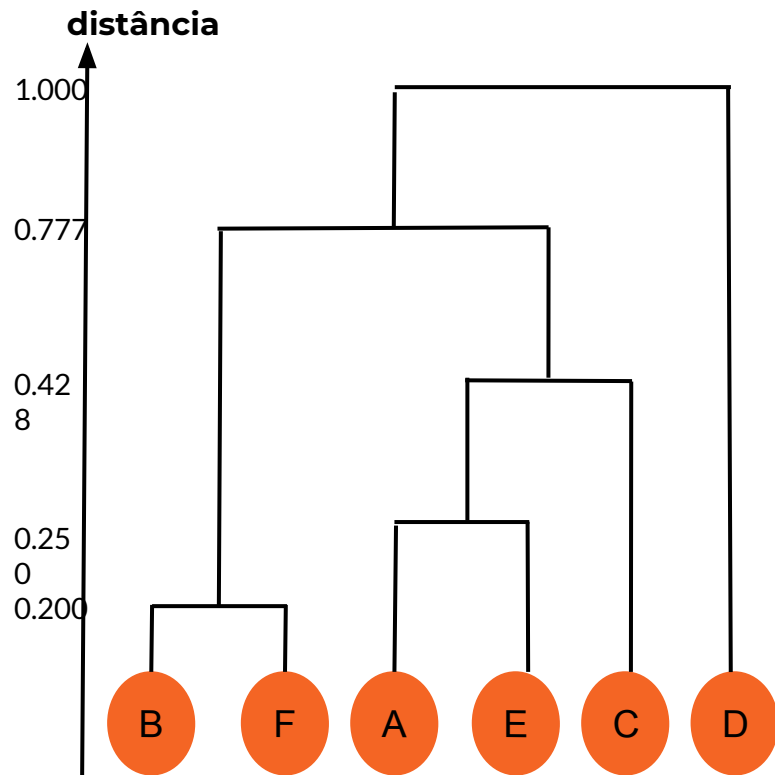
Características do H-Clustering



- Número de cluster é um hiperparâmetro.
- Métricas de distância.
- Definição do tipo de ligação.
- Definição do K, através do dendograma.

2. Vantagens & Desvantagens do H-Clustering

Características do H-Clustering



Vantagens

- Simples de implementar.
- Fácil de definir o número k de clusters.

Desvantagens

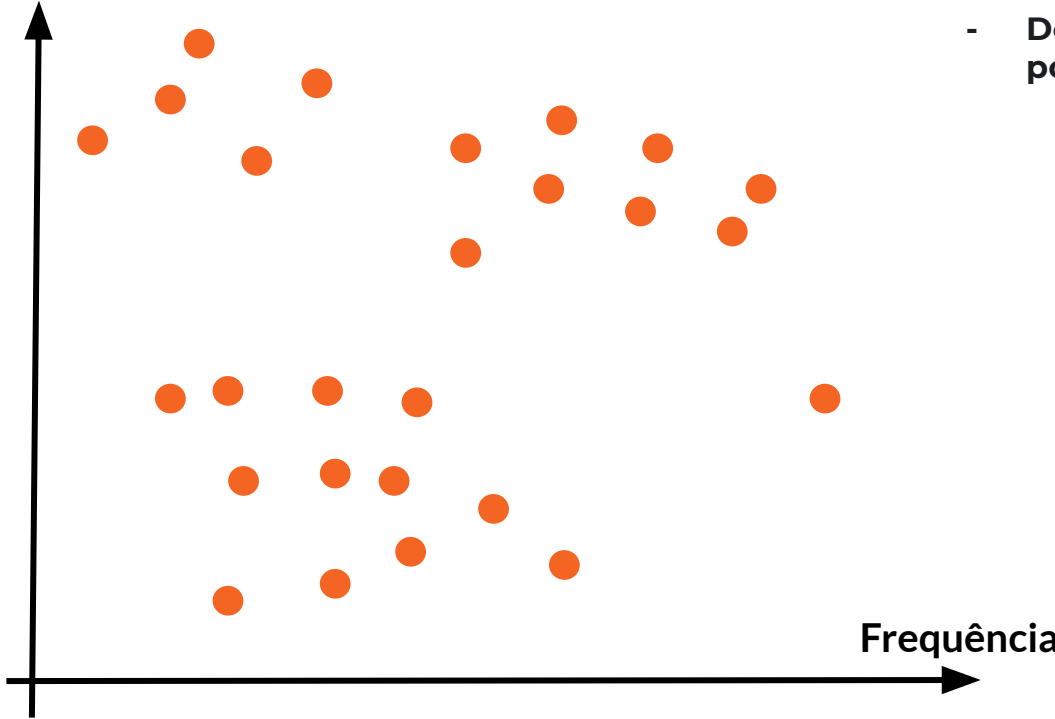
- Uma vez o ponto atribuído a um cluster, não há modificações.
- Alta complexidade. Lento para conjunto de dados muito grandes.
- Sensível na presença de outliers.
- Escolha do critério de ligação.

4. O funcionamento do DBSCAN

(Density-Based Spatial Clustering Application with Noise)

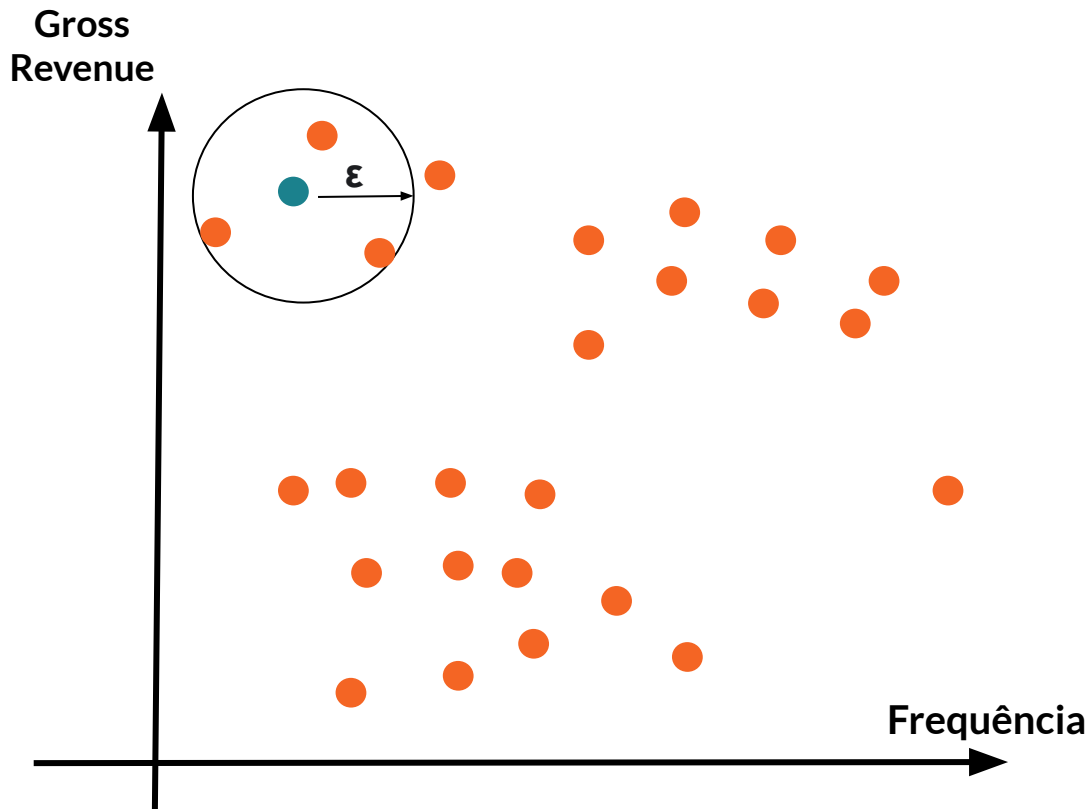
O funcionamento do DBSCAN

Gross
Revenue



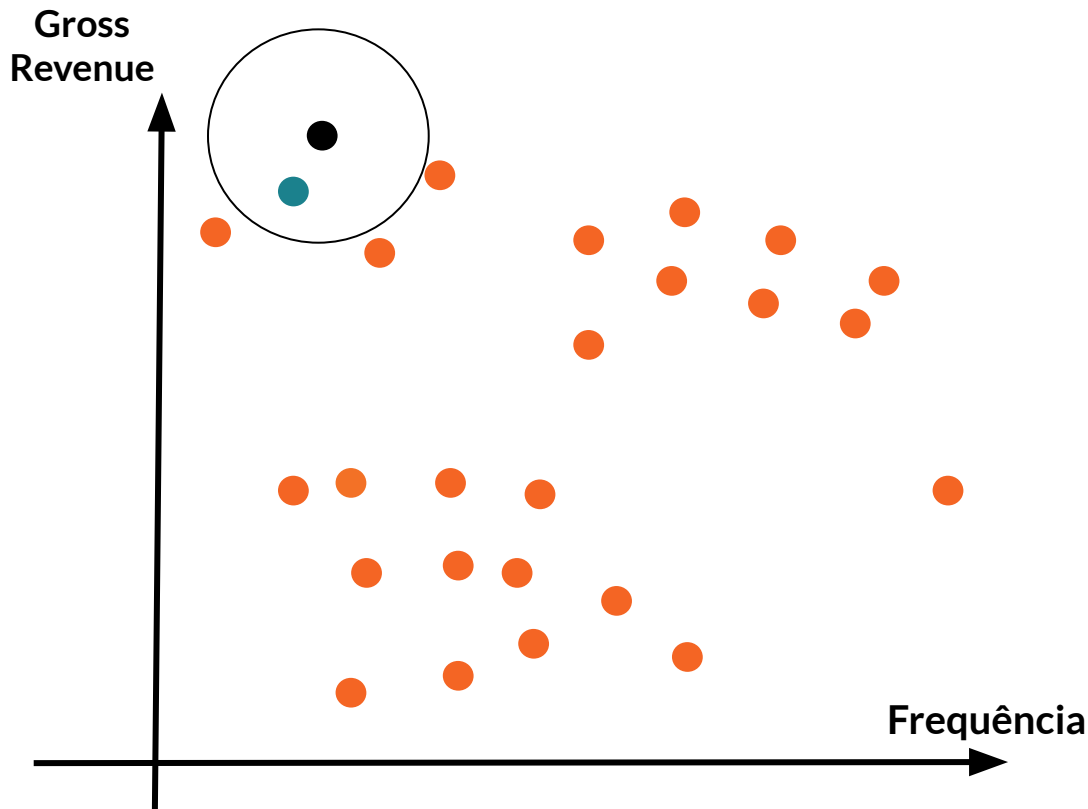
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N

O funcionamento do DBSCAN



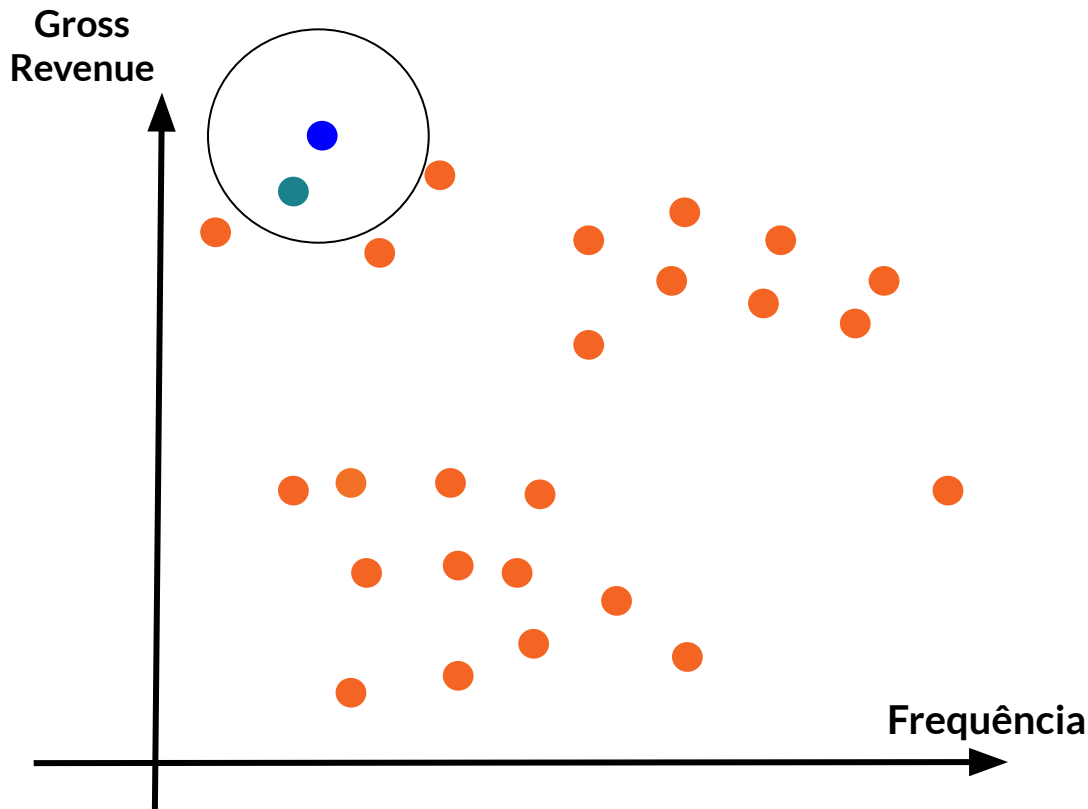
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “**core point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “**border point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “**noise point**”.

O funcionamento do DBSCAN



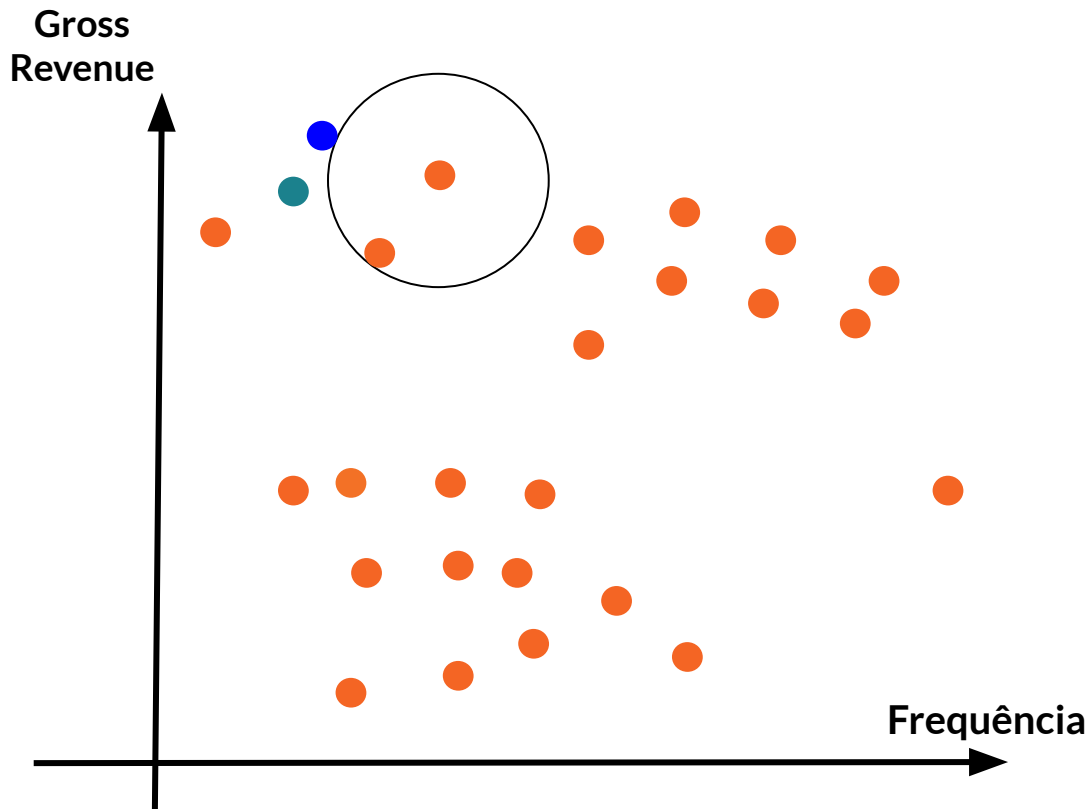
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



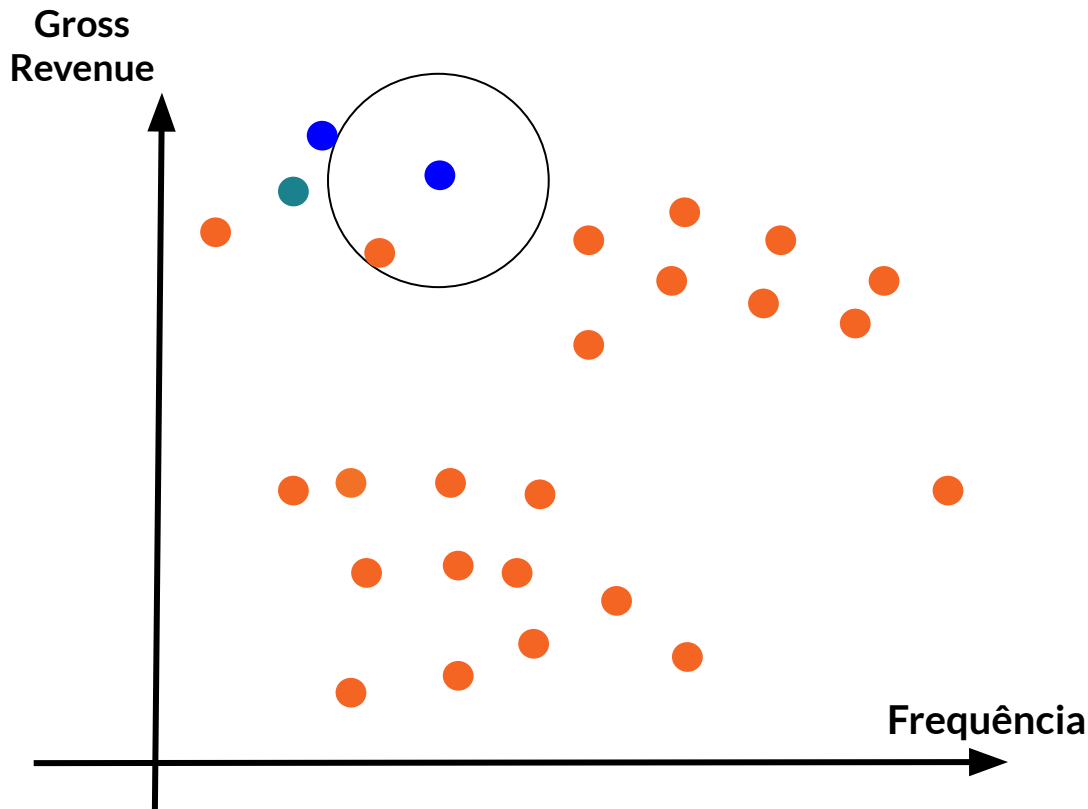
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



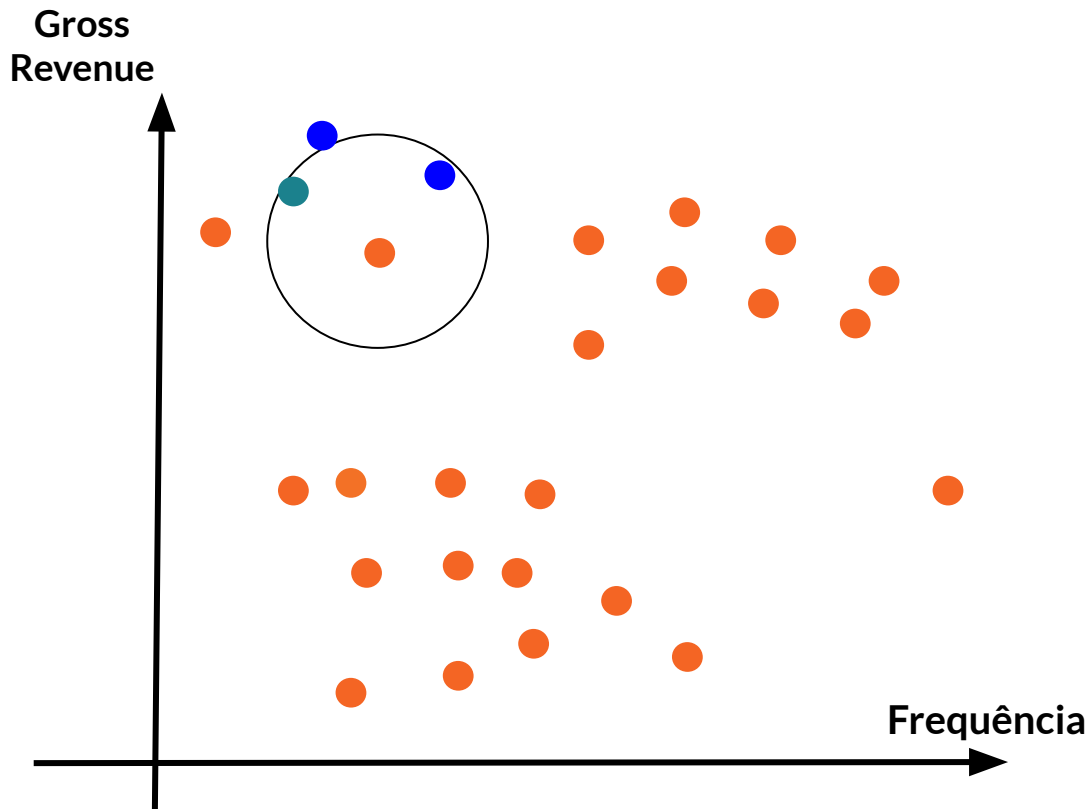
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



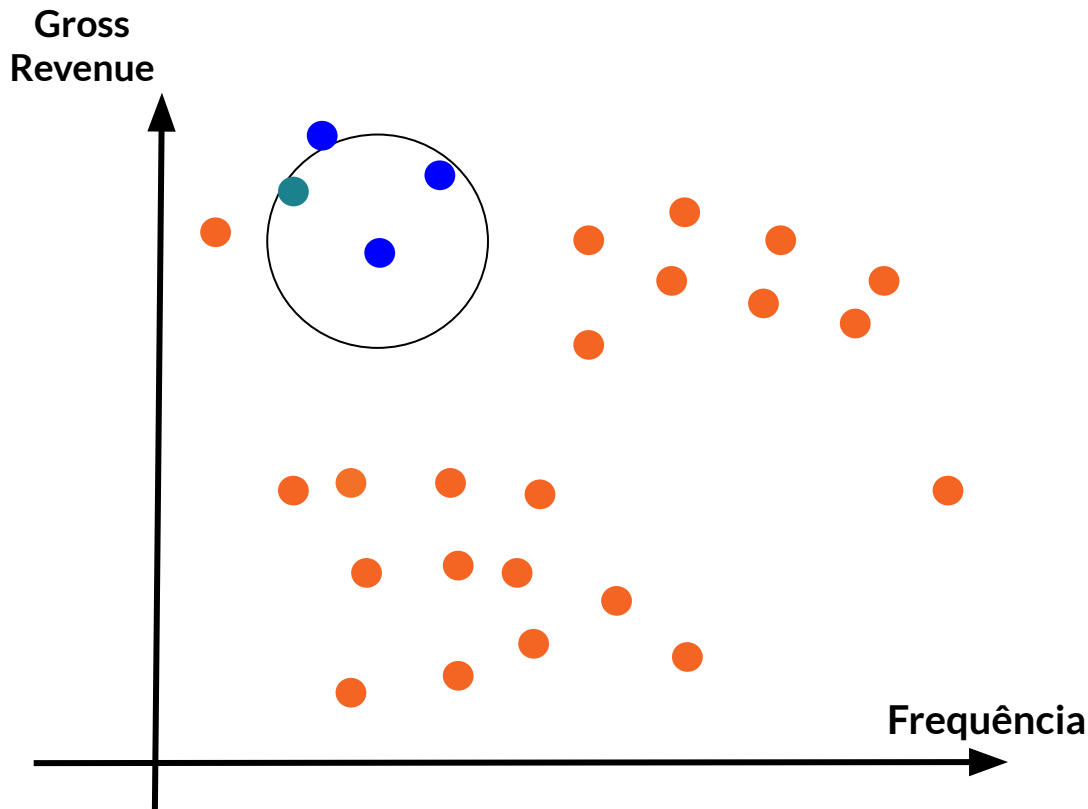
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “**core point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “**border point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “**noise point**”.

O funcionamento do DBSCAN



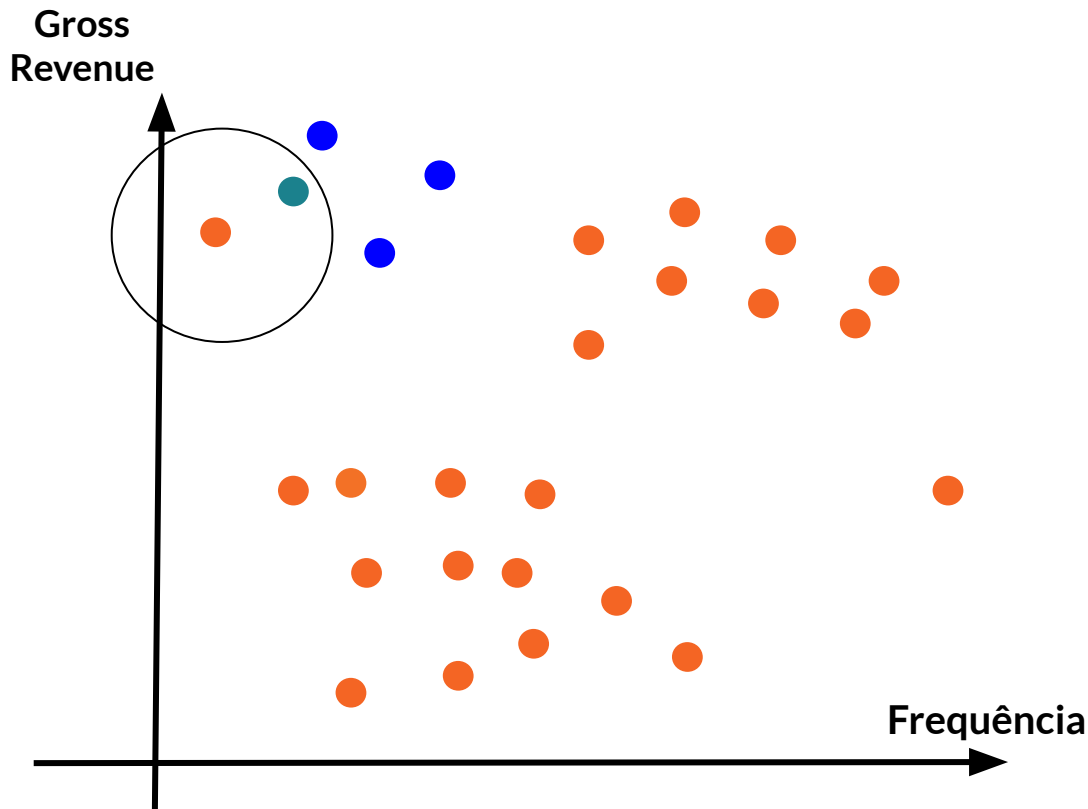
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “**core point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “**border point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “**noise point**”.

O funcionamento do DBSCAN



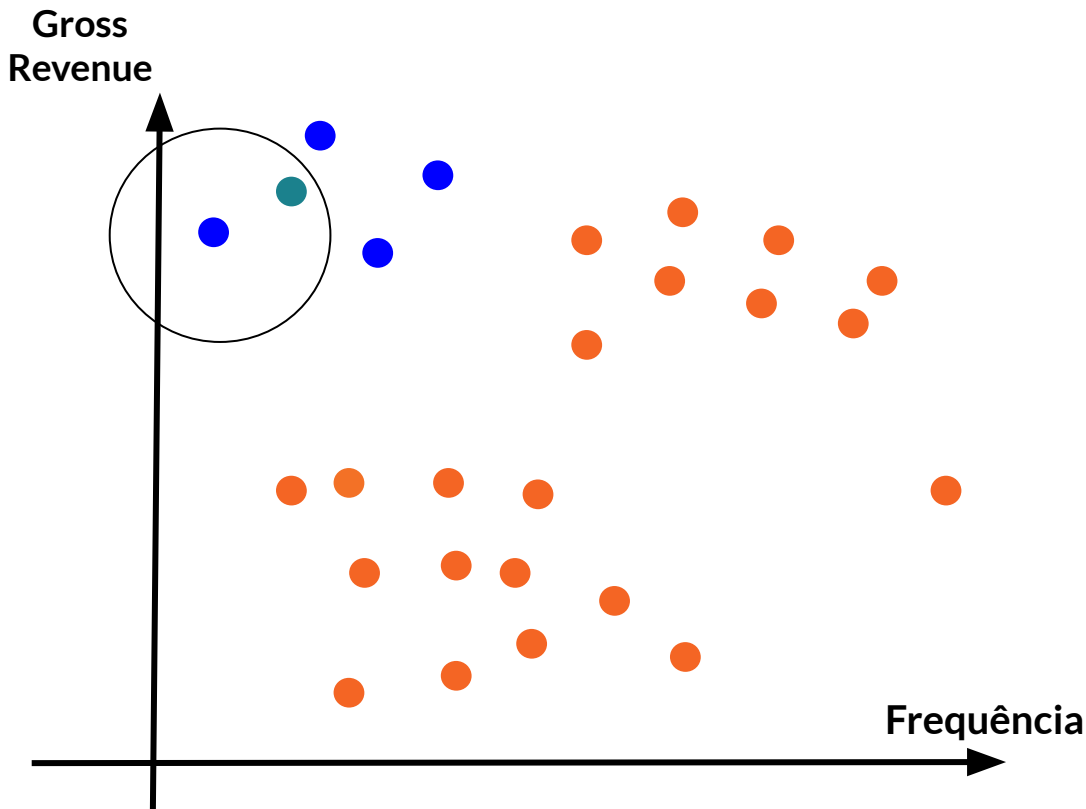
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “**core point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “**border point**”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “**noise point**”.

O funcionamento do DBSCAN



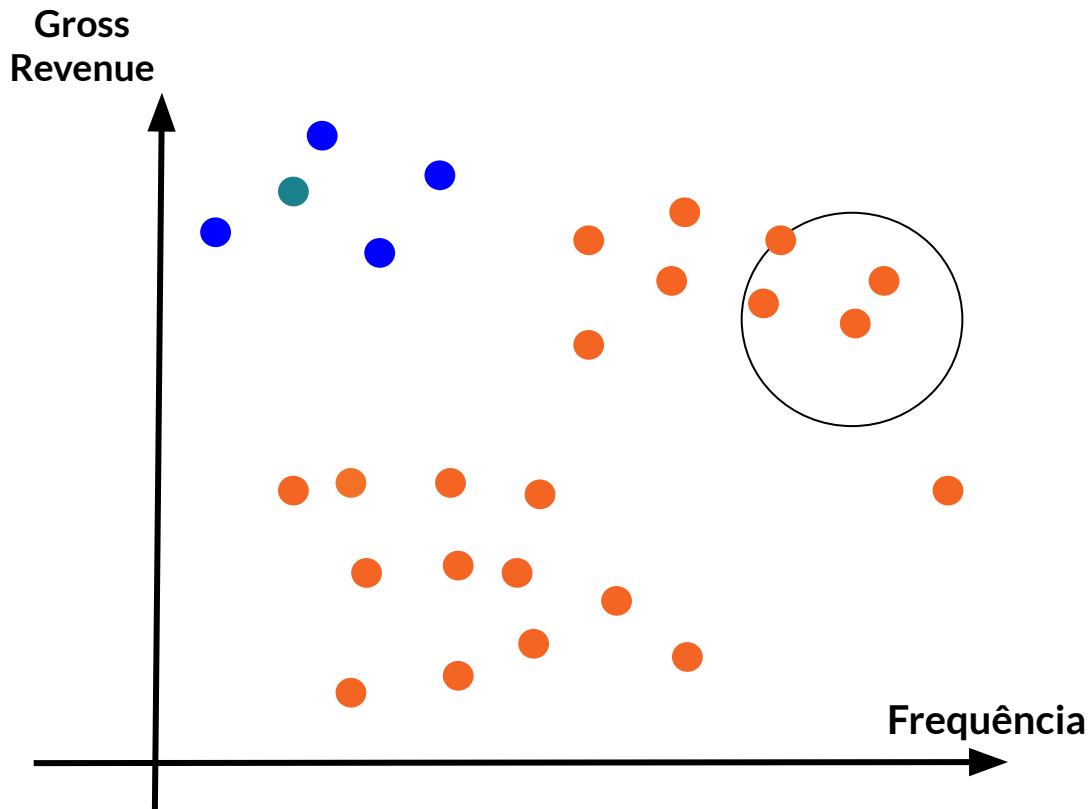
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



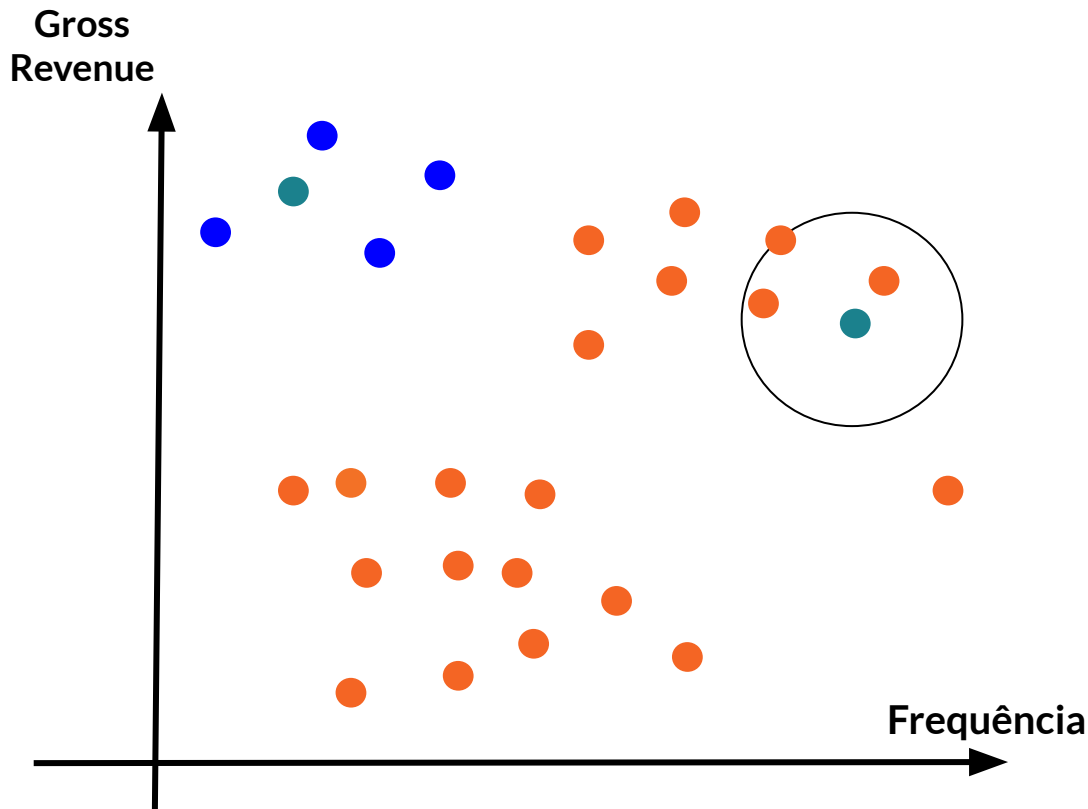
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



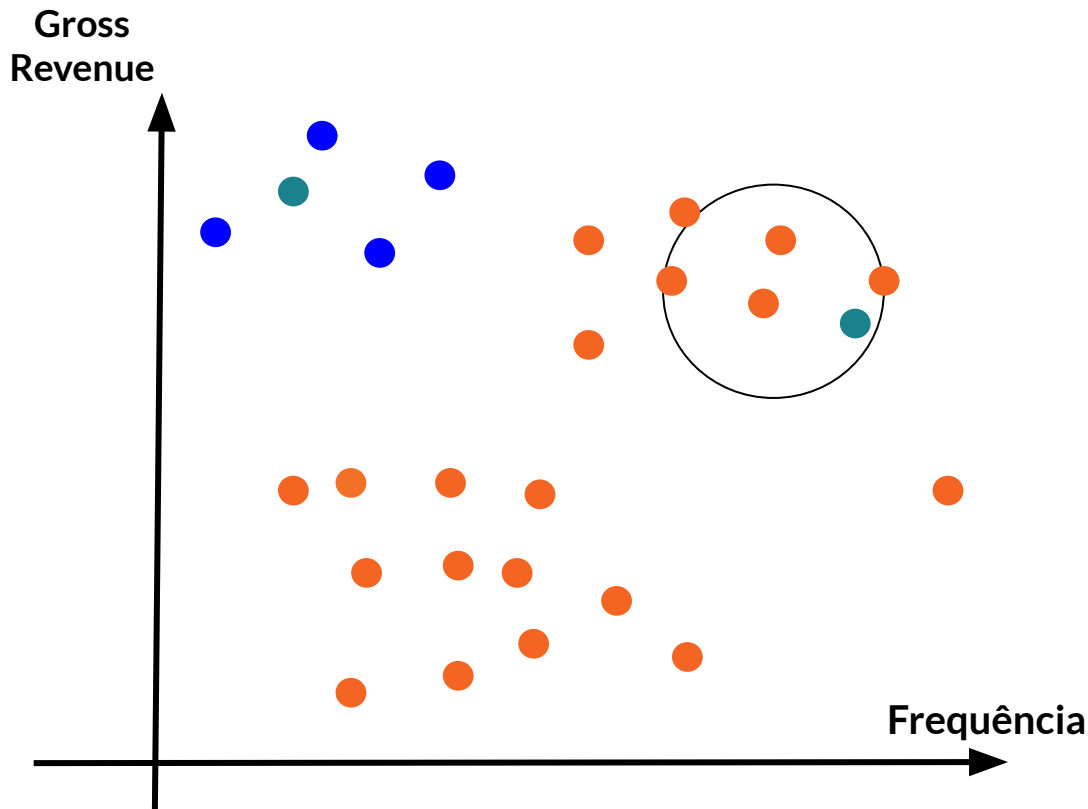
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



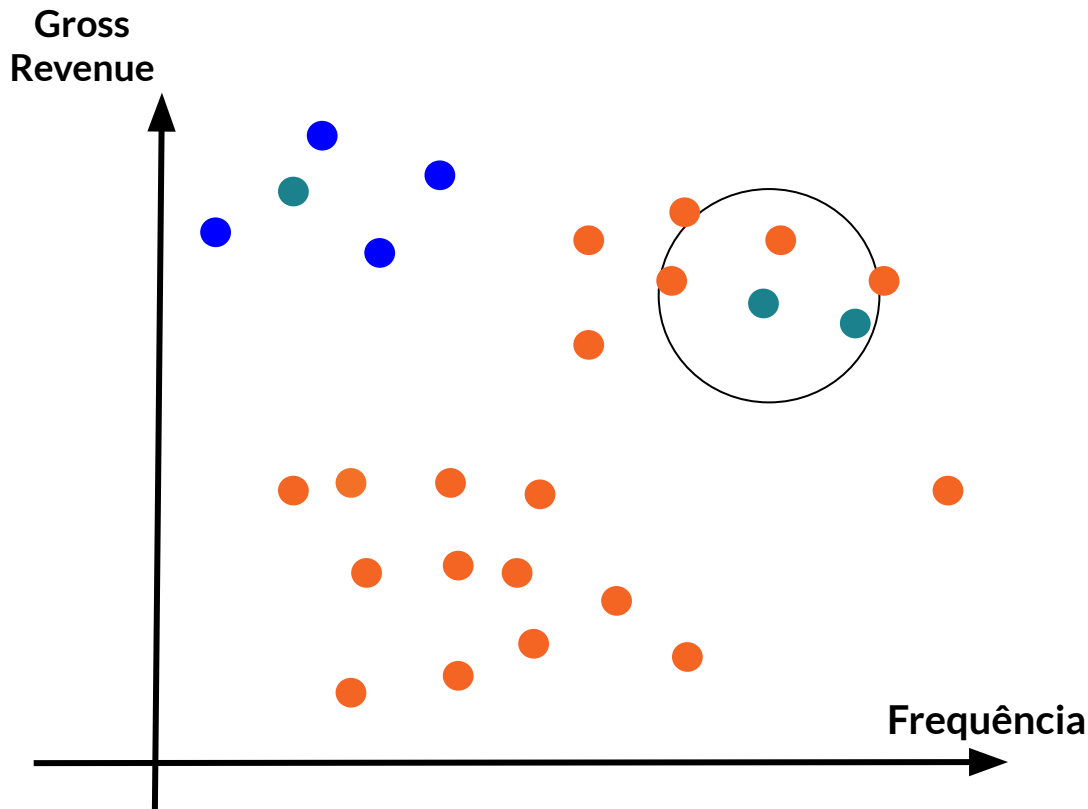
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



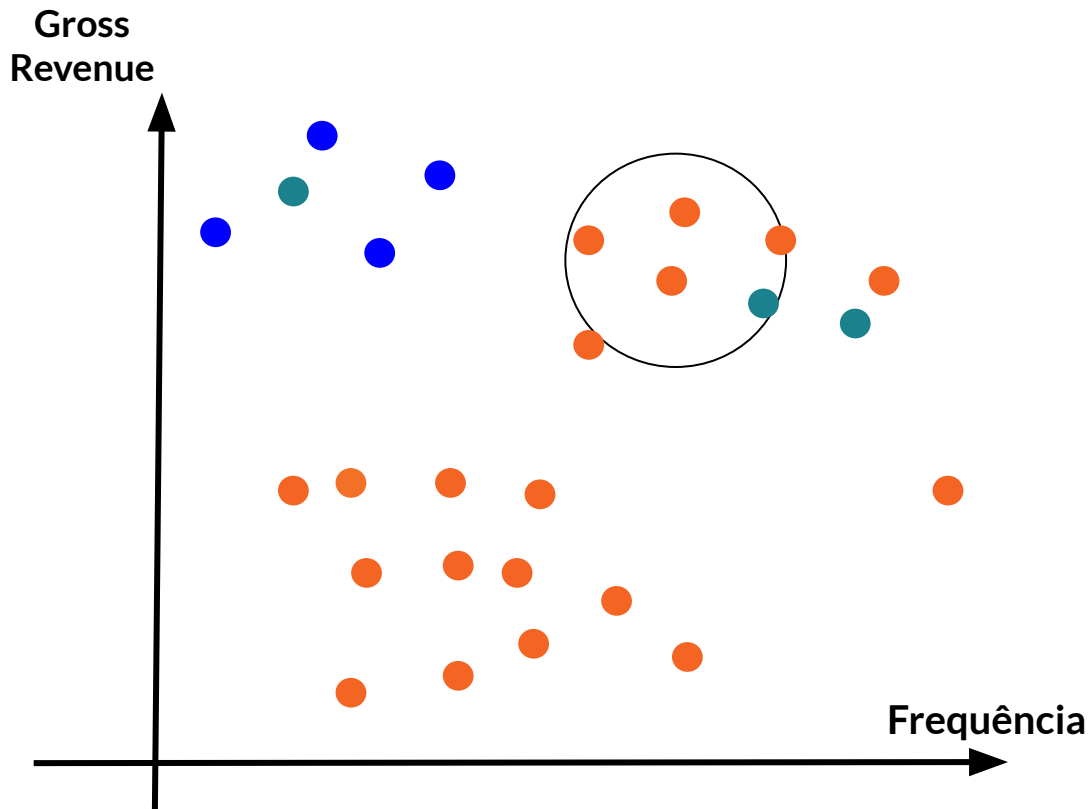
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



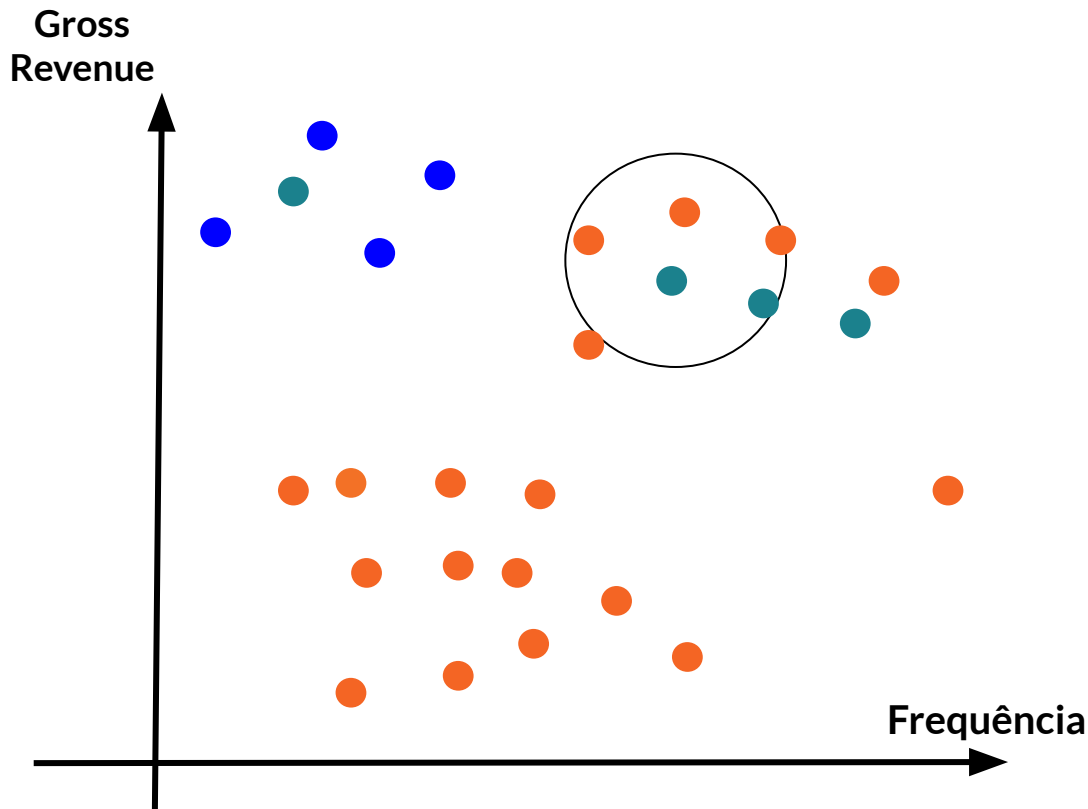
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



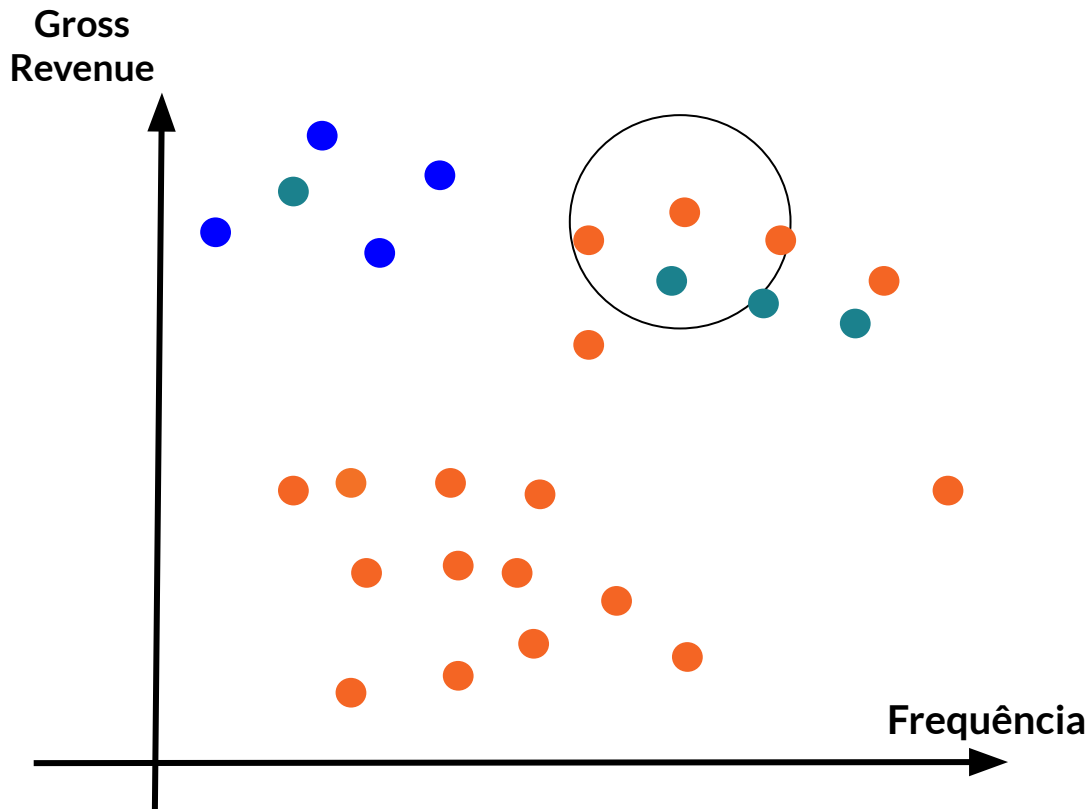
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



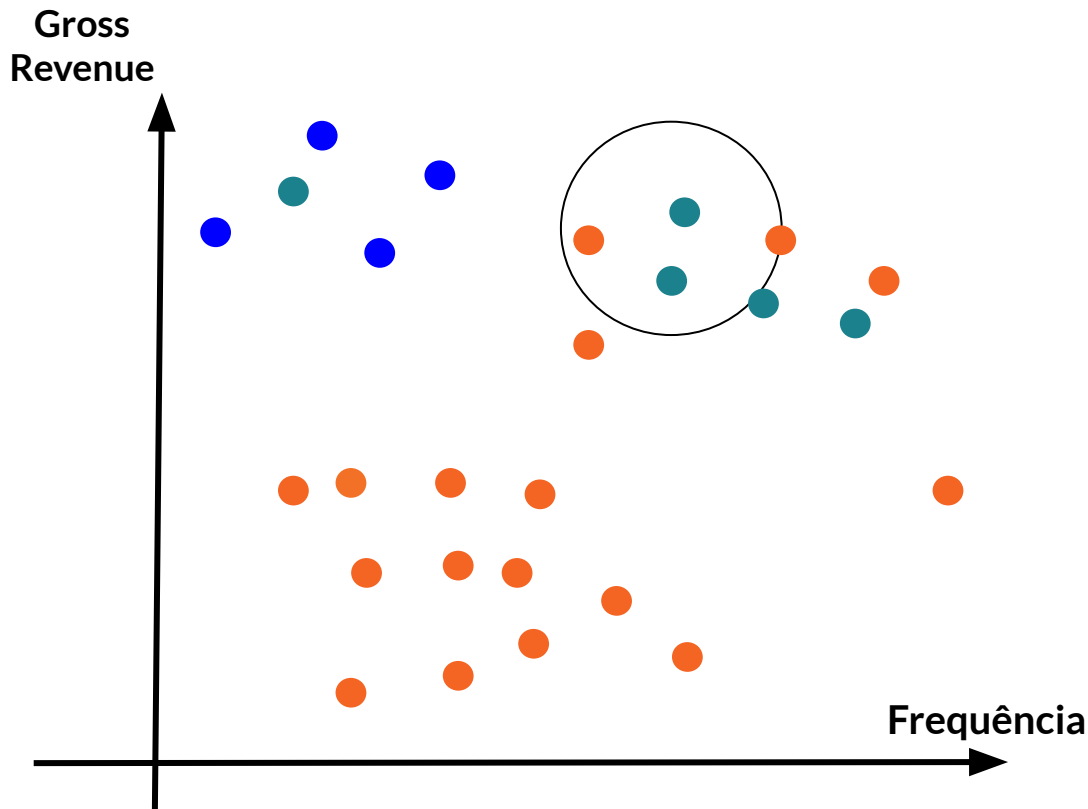
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



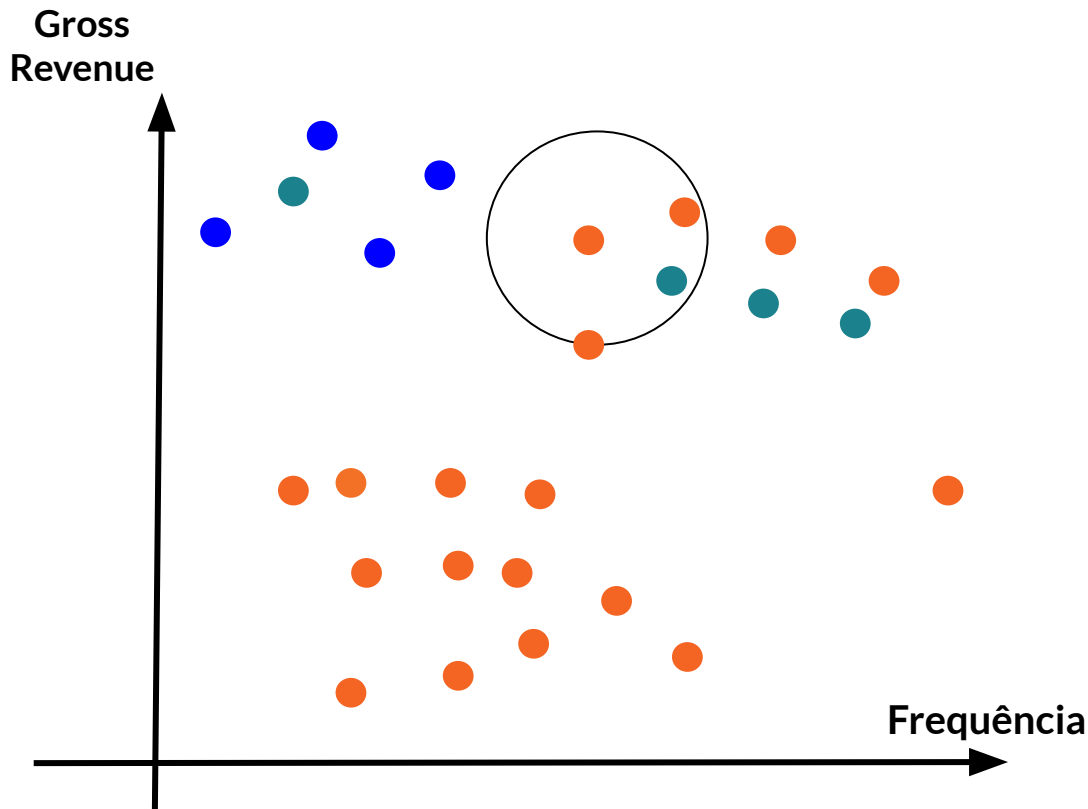
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



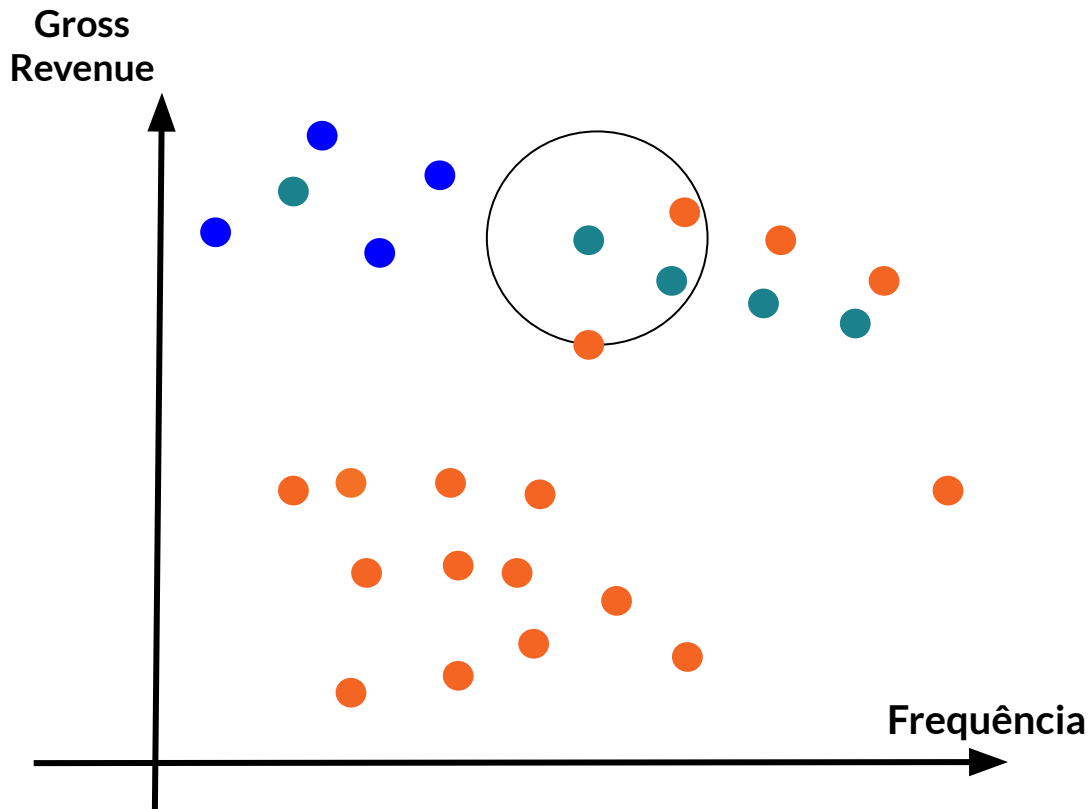
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



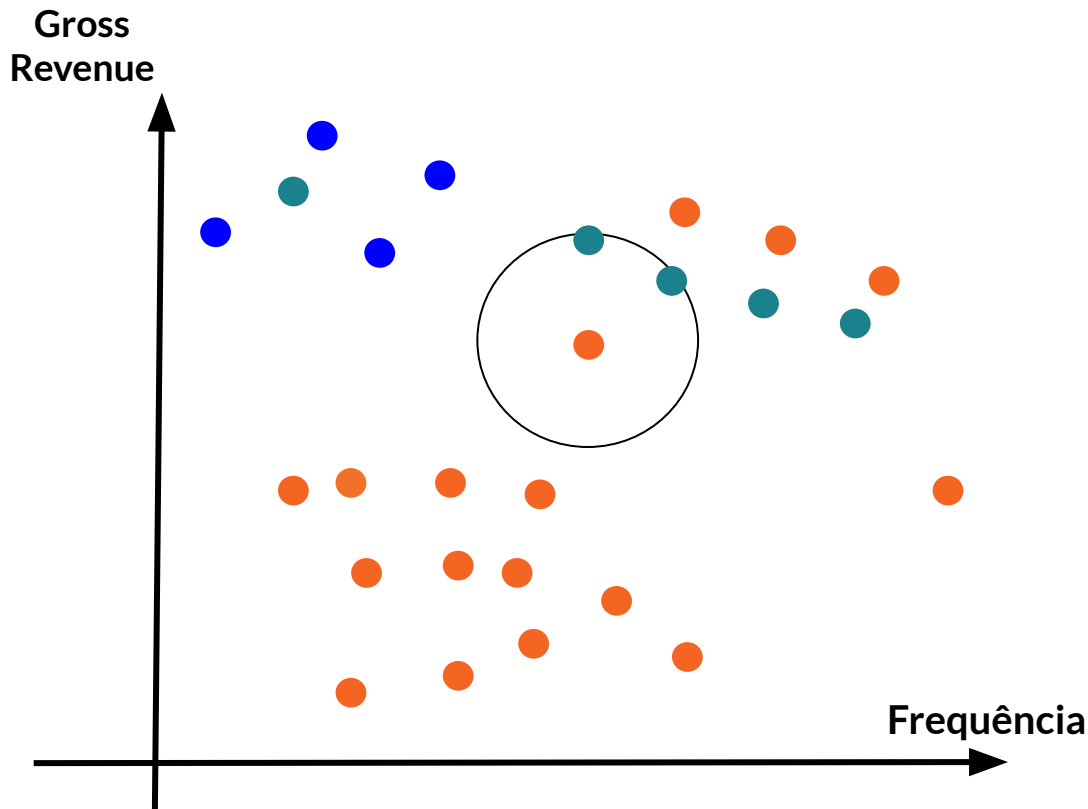
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



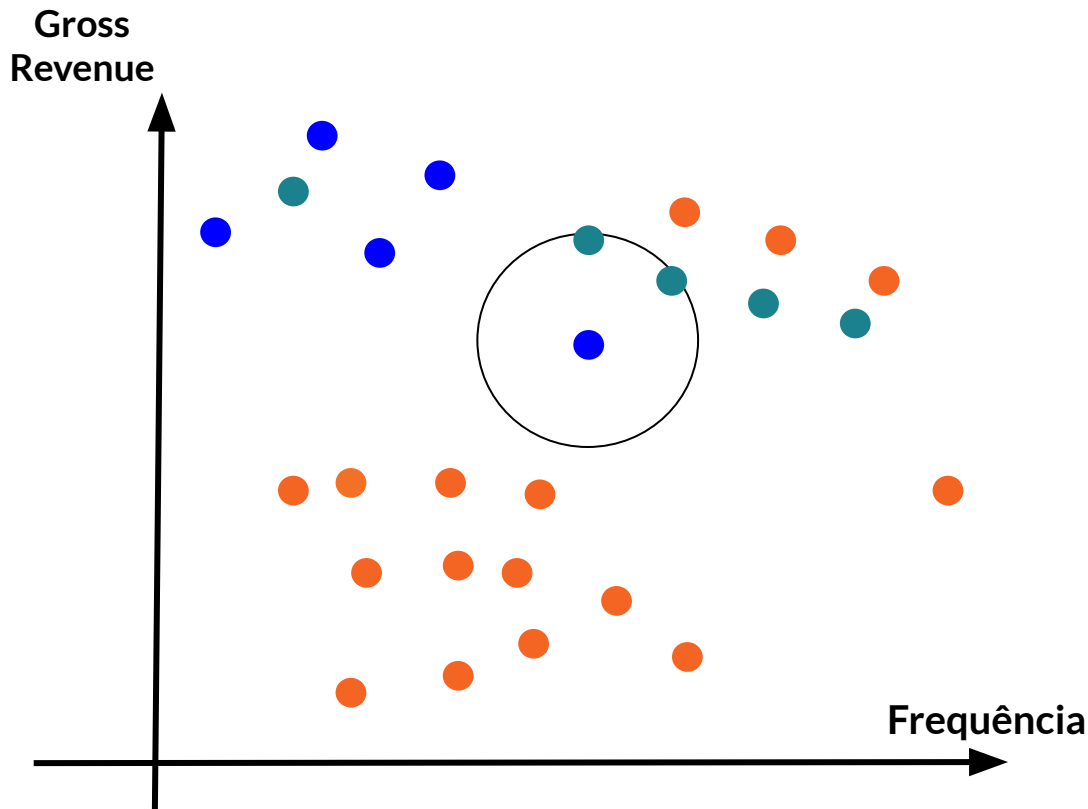
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



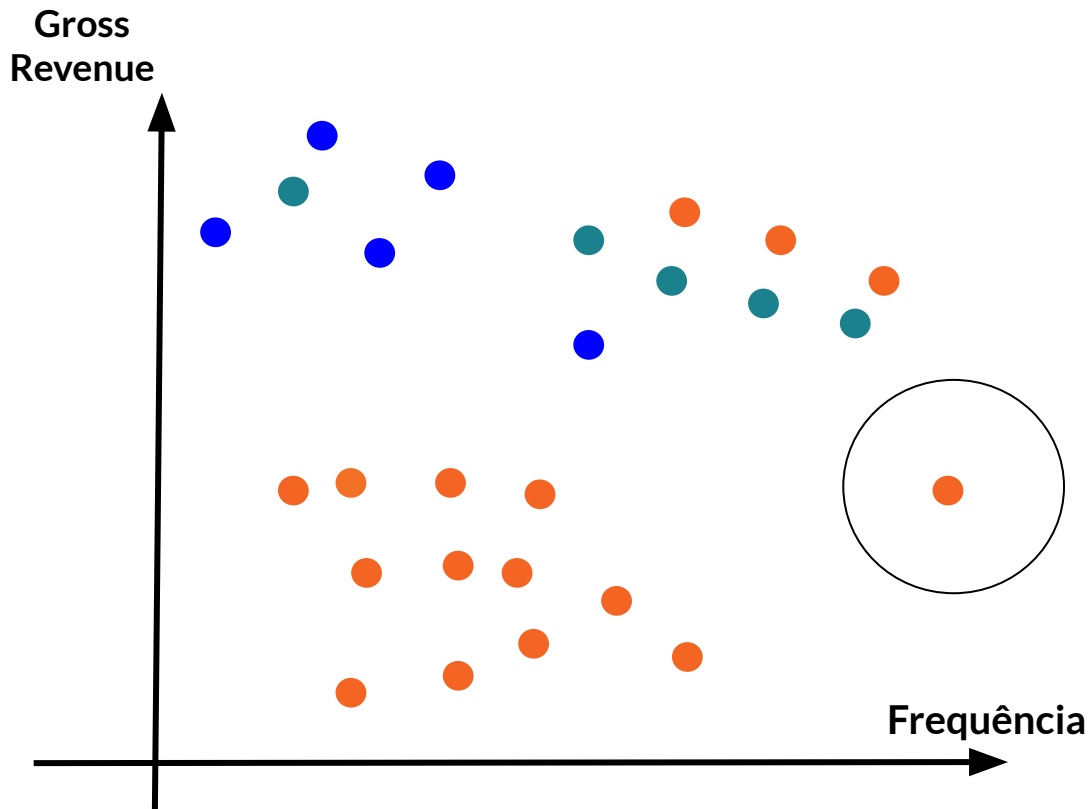
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



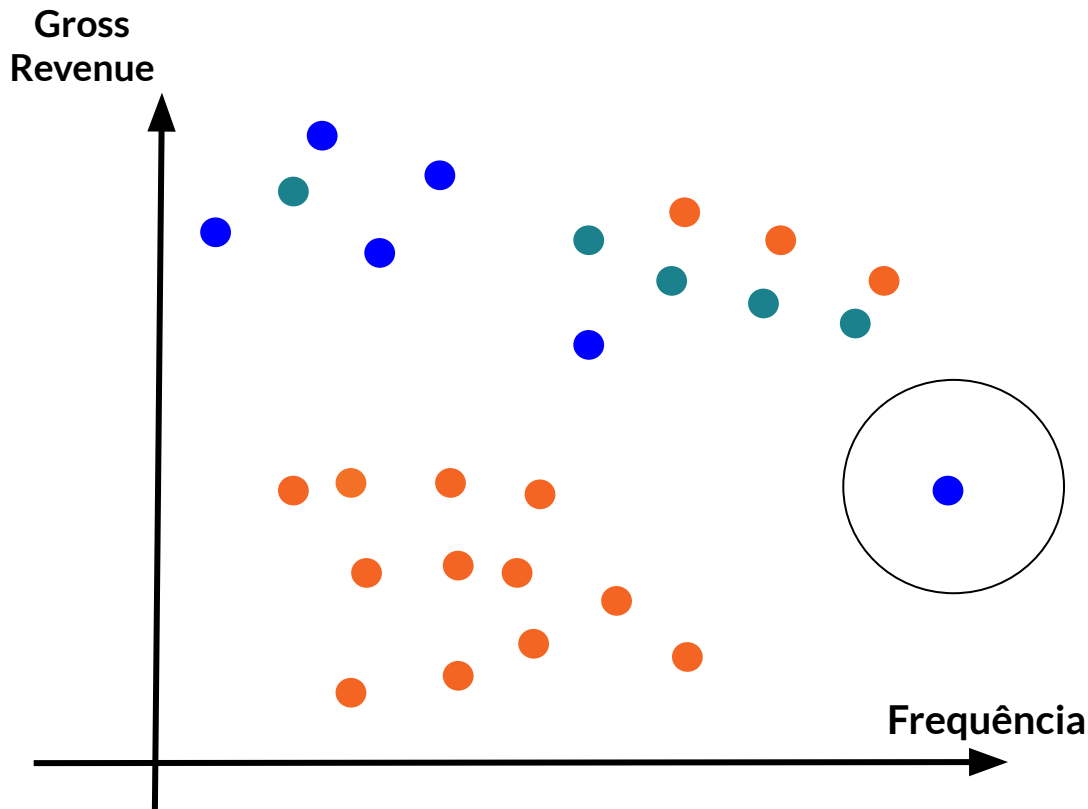
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

O funcionamento do DBSCAN



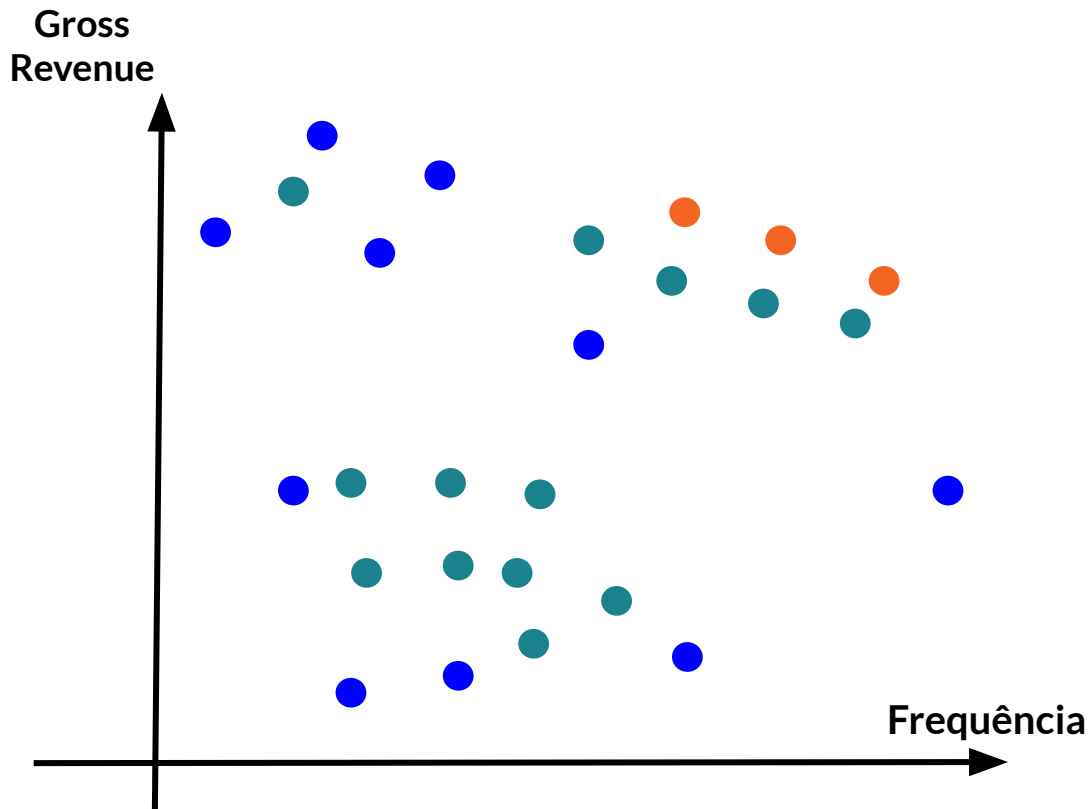
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



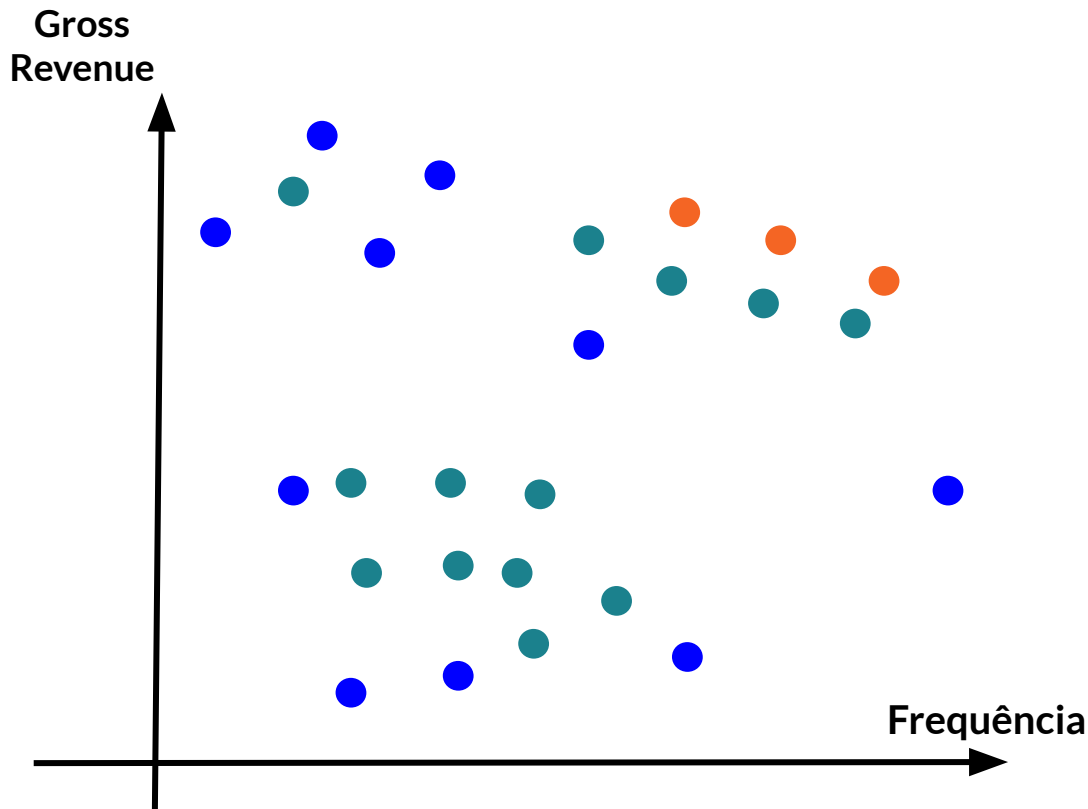
- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.

O funcionamento do DBSCAN



- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

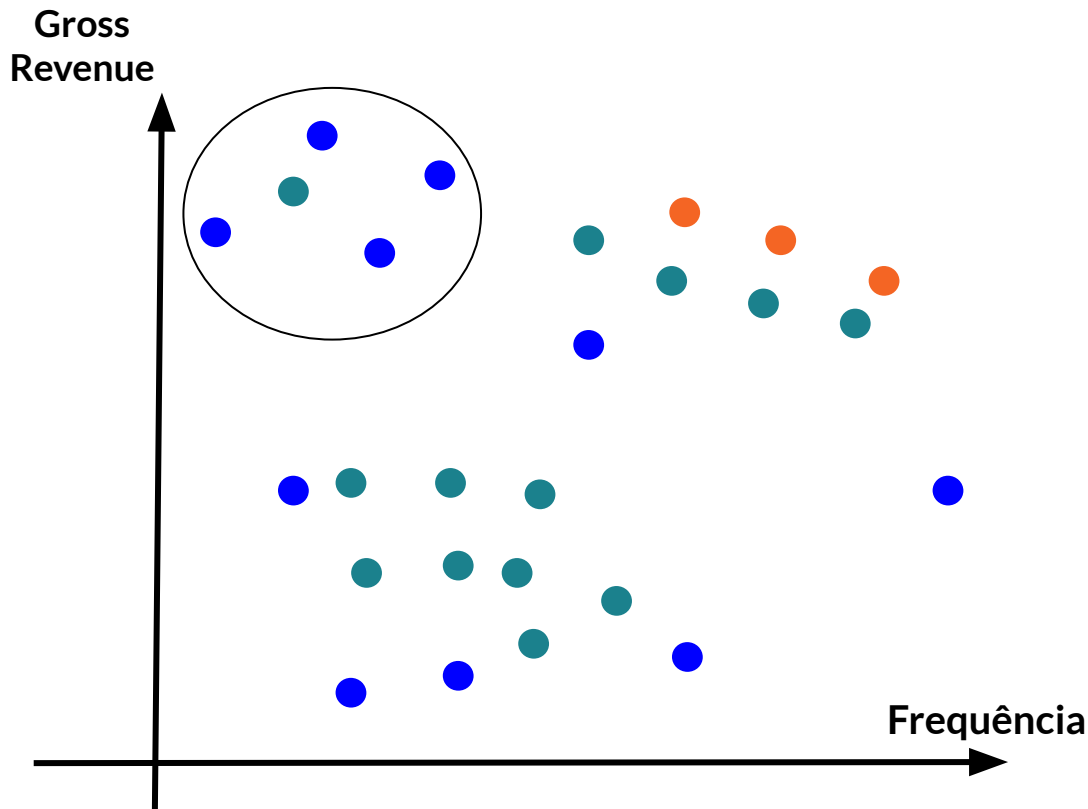
O funcionamento do DBSCAN



- Definir um raio ϵ
- Definir o número mínimo de pontos da vizinhança N
- Se o ponto X , dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, “core point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, “border point”.
- Se o ponto X , dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, “noise point”.

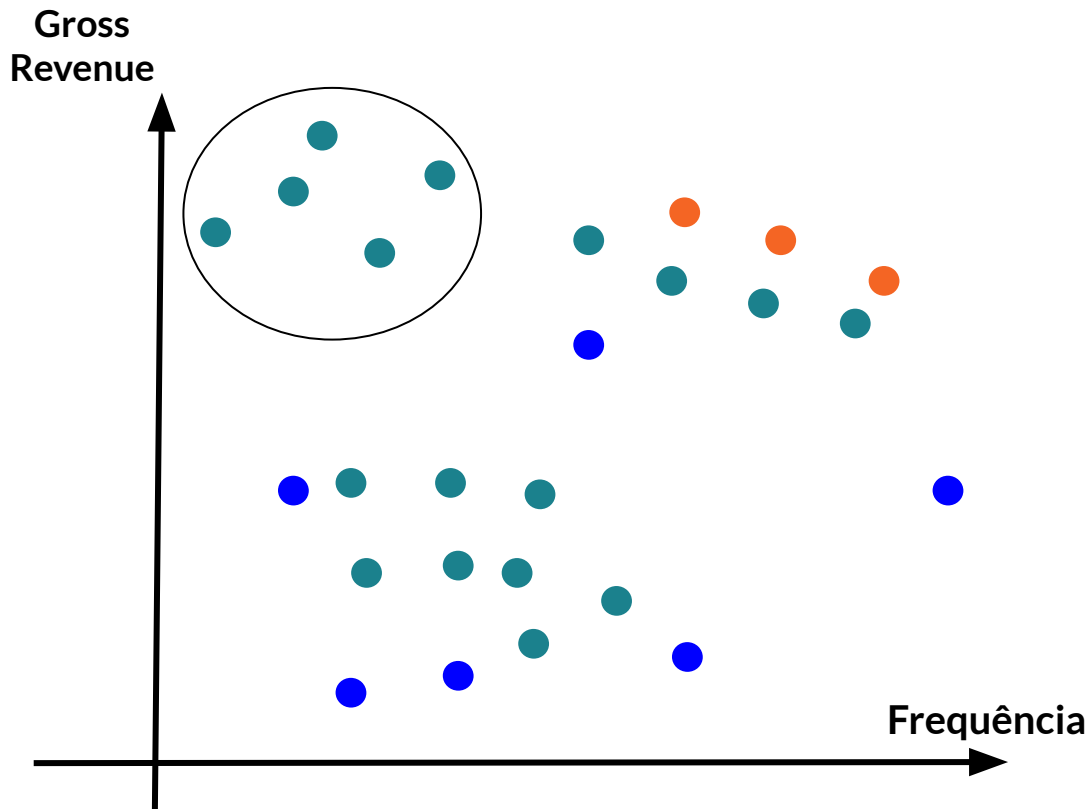
Avalia os pontos marcados

O funcionamento do DBSCAN



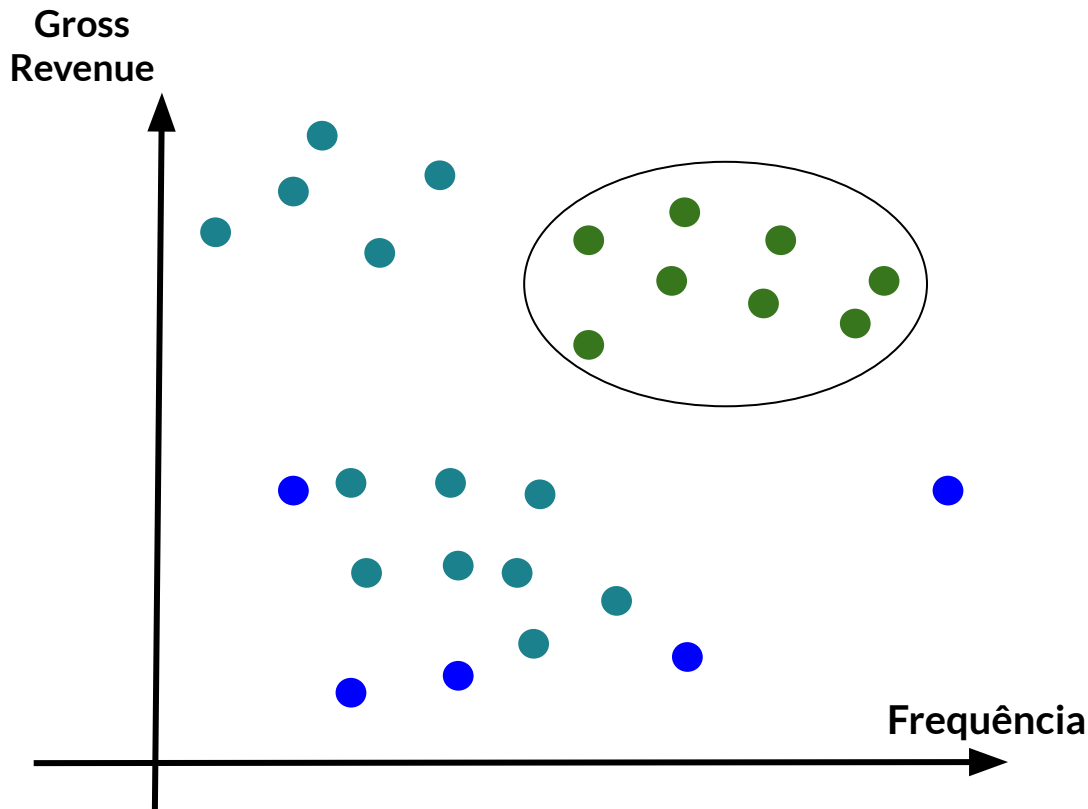
- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

O funcionamento do DBSCAN



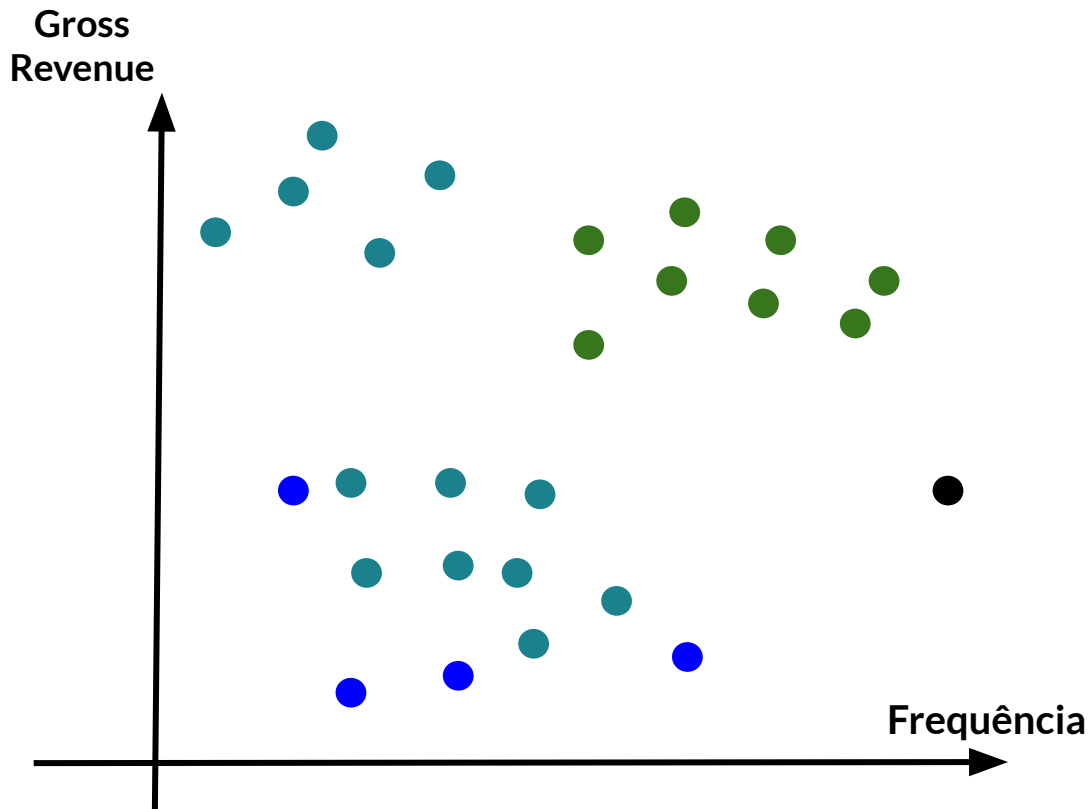
- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

O funcionamento do DBSCAN



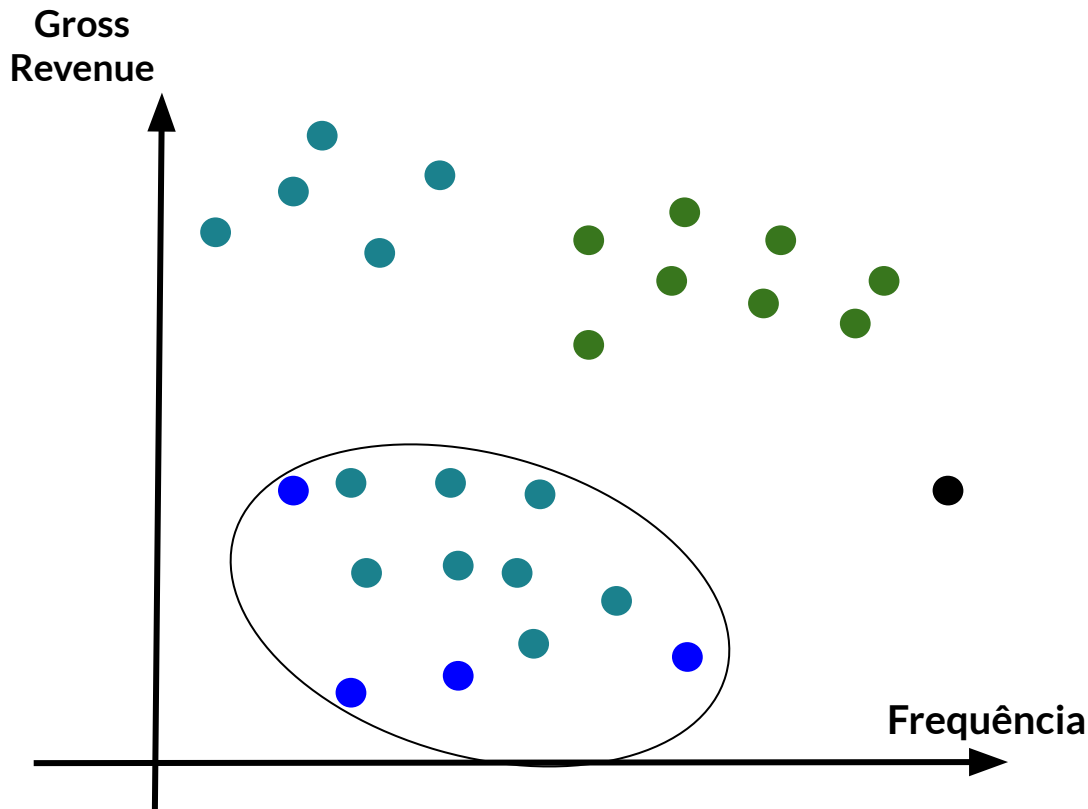
- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

O funcionamento do DBSCAN



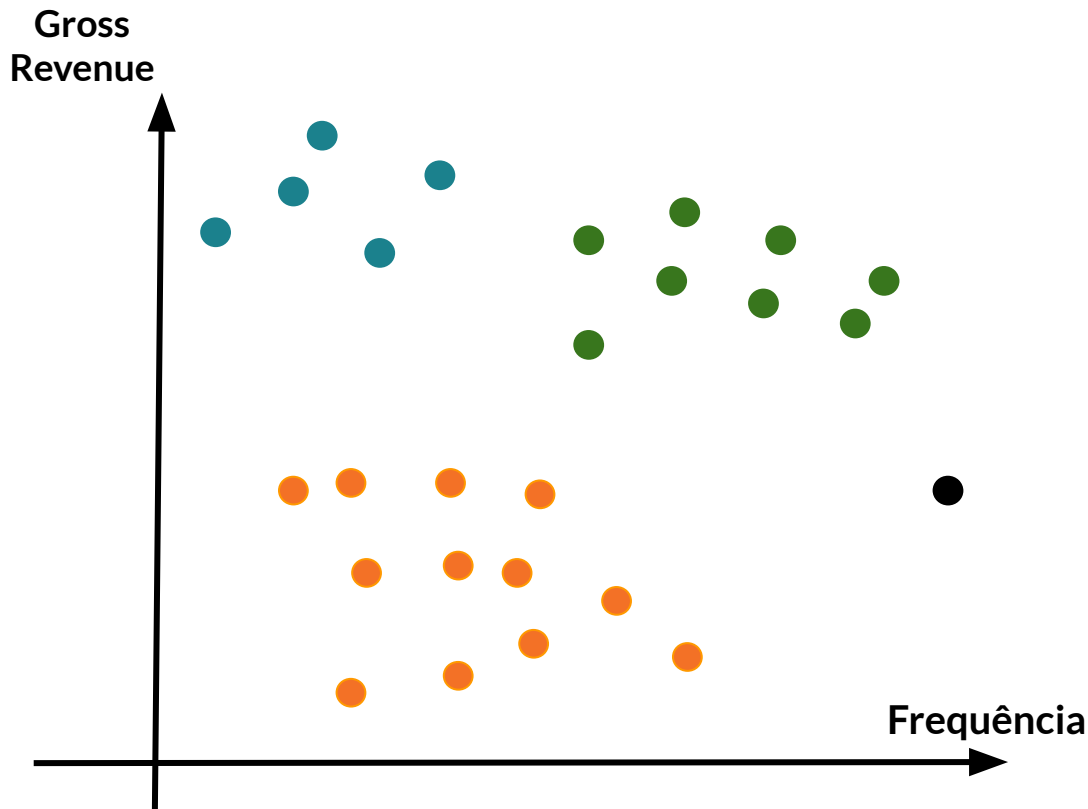
- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

O funcionamento do DBSCAN



- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

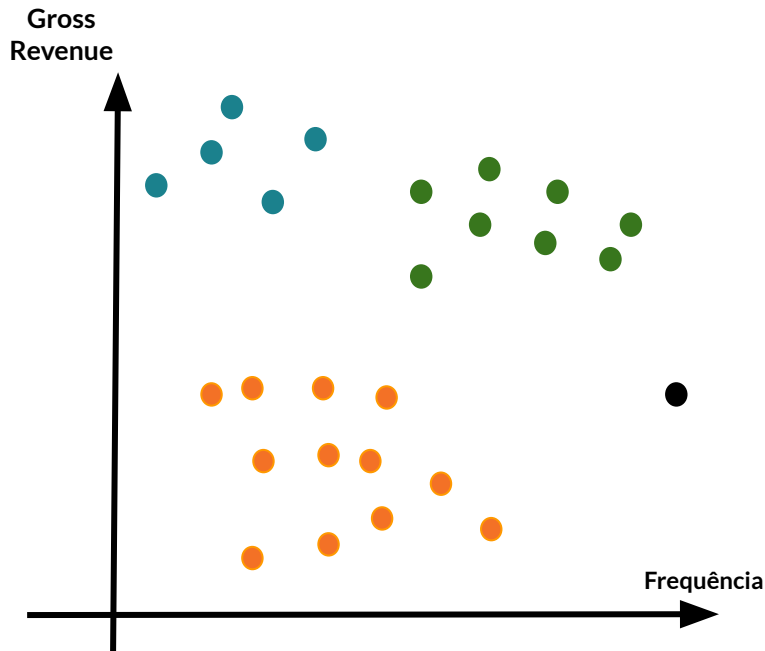
O funcionamento do DBSCAN



- Se o ponto X, dentro de uma vizinhança ϵ tem mais pontos que o mínimo. Logo, **“core point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele é vizinho de um “core point”. Logo, **“border point”**.
- Se o ponto X, dentro de uma vizinhança ϵ tem menor pontos que o mínimo e ele não é vizinho de um “core point”. Logo, **“noise point”**.
- Para cada “core point” avalia os pontos da vizinhança em “border point” ou “noise point”.

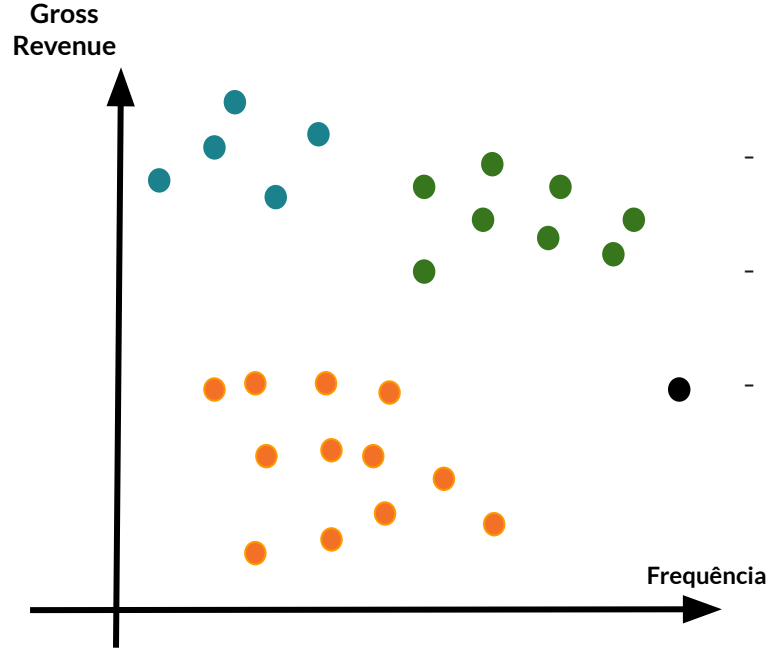
Características do DBSCAN

Características do DBSCAN



- Número de cluster não é um hiperparâmetro.
- Definição do raio da vizinhança.
- Definição do número mínimo de pontos da vizinhança.
- Definição da métricas de distância.

Características do DBSCAN



Vantagens

- Encontra cluster de qualquer formato.
- Encontra o número de clusters automaticamente.

Desvantagens

- Muito sensível aos parâmetros.
- Difícil o fine-tuning.

Perguntas?

Referência

Hierarchical Clustering

- Flexível em relação aos formatos dos clusters.
- Permite incluir incertezas na atribuição do ponto ao cluster.
- Robusto aos outliers.
- Não depende de métricas de distâncias.