

Aprendizado Não Supervisionado

Definição

- O objetivo principal é **encontrar estruturas ocultas ou padrões em um conjunto de dados**, mas sem a orientação explícita de rótulos ou saídas desejadas;
- Diferentemente do aprendizado supervisionado, onde o modelo é treinado com exemplos rotulados, **o aprendizado não supervisionado lida com dados não rotulados e visa descobrir informações úteis a partir deles.**

Agrupamento

**Redução de
Dimensionalidade**

**Detecção de
Anomalias**

**Regras de
Associação**

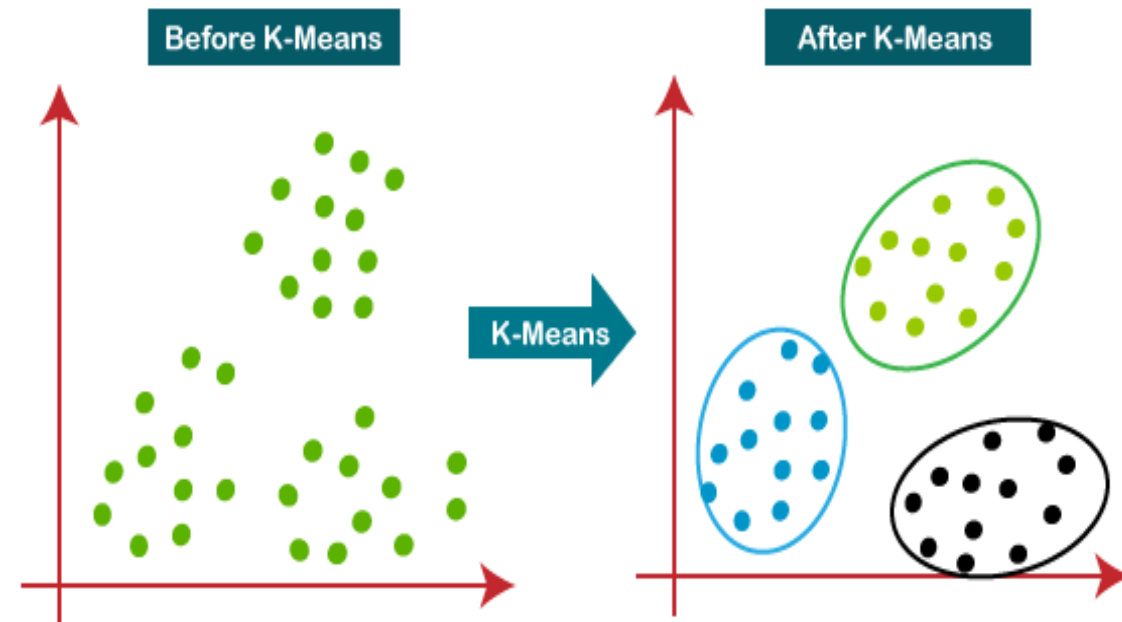
Agrupamento	Redução de Dimensionalidade	Detecção de Anomalias	Regras de Associação
Categorizar os dados em grupos ou <i>clusters</i> com base em similaridades entre os pontos de dados.	Reduzir o número de variáveis em um conjunto de dados, mantendo o máximo de informações possível.	Identificar pontos de dados que se desviam significativamente do comportamento normal em um conjunto de dados.	Descobrir padrões ou regras que indicam associações frequentes entre itens em conjuntos de dados.

Algoritmos (Agrupamento)

Algoritmo	K-Means	Agrupamento Hierárquico	DBSCAN
Funcionamento	Agrupa dados em clusters onde cada ponto de dados pertence ao cluster com a média mais próxima	Cria hierarquia de clusters em árvore	Identifica clusters com densidade similar
Vantagens	Simplicidade, escalável para grandes conjuntos	Visão hierárquica dos dados	Detecta formas arbitrárias
Desvantagens	Sensível à inicialização de centróides	Complexidade computacional mais alta	Sensível a hiperparâmetros

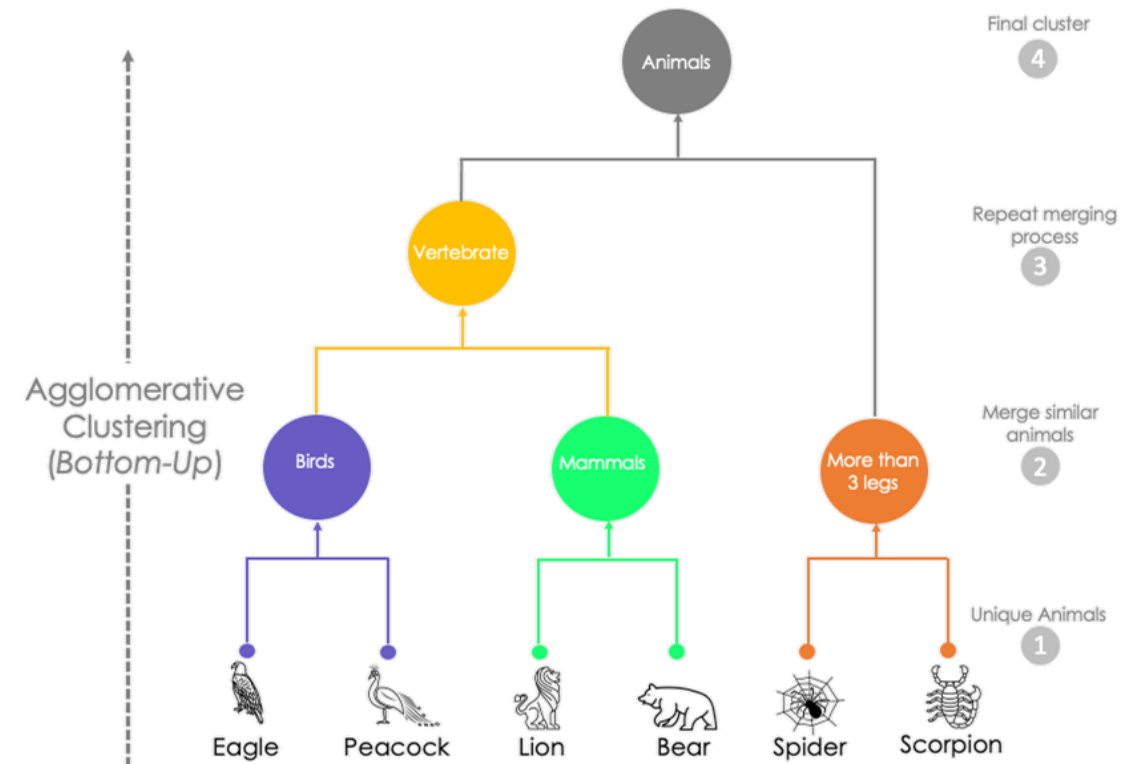
K-Means

- 1.Inicialização: Escolher K pontos iniciais (centróides) aleatoriamente ou com algum método específico;
- 2.Associação: Atribuir cada ponto de dados ao centróide mais próximo;
- 3.Atualização: Recalcular os centróides com base nos pontos atribuídos;
- 4.Repetir os passos 2 e 3 até que os centróides não mudem significativamente ou um número máximo de iterações seja atingido.



Agrupamento Hierárquico

1. Início: Cada ponto de dados é inicialmente considerado um cluster individual;
2. Mesclagem Recursiva: Os clusters mais próximos são mesclados em um único cluster em cada passo, formando uma árvore hierárquica;
3. Terminação: O processo continua até que todos os pontos de dados estejam em um único cluster (agrupamento completo) ou até que um critério específico seja atingido.



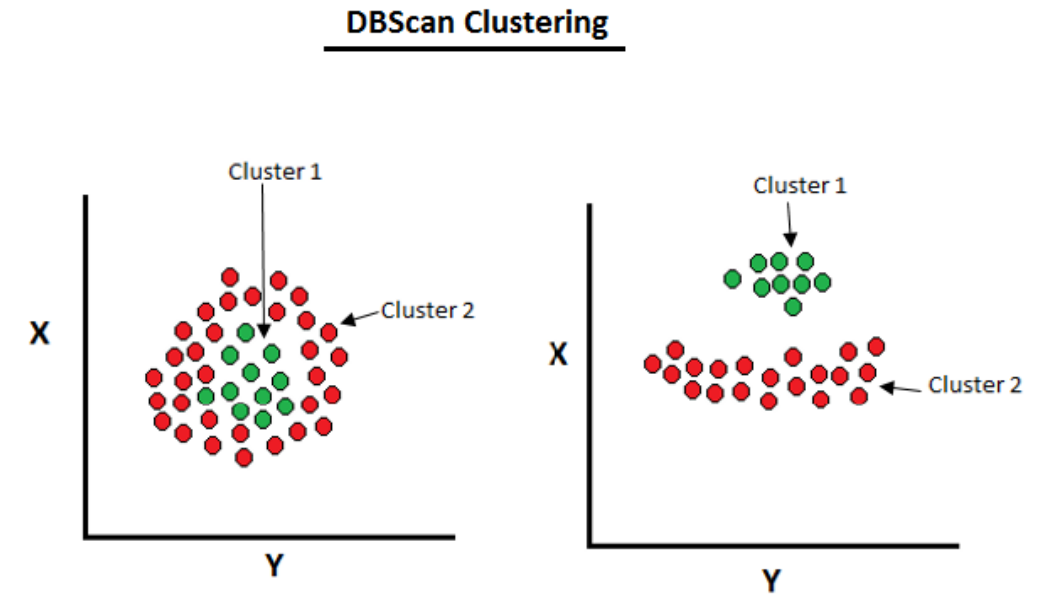
DBSCAN

1. Ponto Central: Começar com um ponto de dados aleatório não visitado ou rótulo de ruído;

2. Vizinhança: Encontrar todos os pontos de dados em uma vizinhança específica (definida por uma distância epsilon) do ponto central;

3. Verificação de Densidade: Se o número de pontos na vizinhança exceder um limite mínimo (definido por minPts), um novo cluster é formado;

4. Expansão: O algoritmo se expande iterativamente adicionando pontos vizinhos aos clusters existentes até que não haja mais pontos a serem adicionados.



Algoritmos (Redução de Dimensionalidade)

Algoritmo	PCA (<i>Principal Component Analysis</i>)	t-SNE (<i>t-Distributed Stochastic Neighbor Embedding</i>)
Funcionamento	Encontra novas variáveis que são combinações lineares das variáveis originais	Visualiza dados de alta dimensão em um espaço de baixa dimensão
Vantagens	Reduz dimensionalidade, preserva informações importantes	Mantém relações locais entre dados
Desvantagens	Componentes podem ser difíceis de interpretar	Não é adequado para redução de dimensionalidade extrema

PCA

1. Centralização dos Dados: Os dados originais são centrados subtraindo-se a média de cada recurso;
2. Cálculo da Matriz de Covariância: A matriz de covariância entre os recursos é calculada. Isso ajuda a entender as relações entre eles;
3. Cálculo dos Componentes Principais: Os componentes principais são calculados a partir da matriz de covariância. O primeiro componente principal captura a maior variação nos dados, o segundo o segundo maior e assim por diante;
4. Redução de Dimensionalidade: Os componentes principais são classificados em ordem decrescente de importância. Você pode escolher um número menor de componentes principais para representar os dados, reduzindo assim a dimensionalidade.

t-SNE

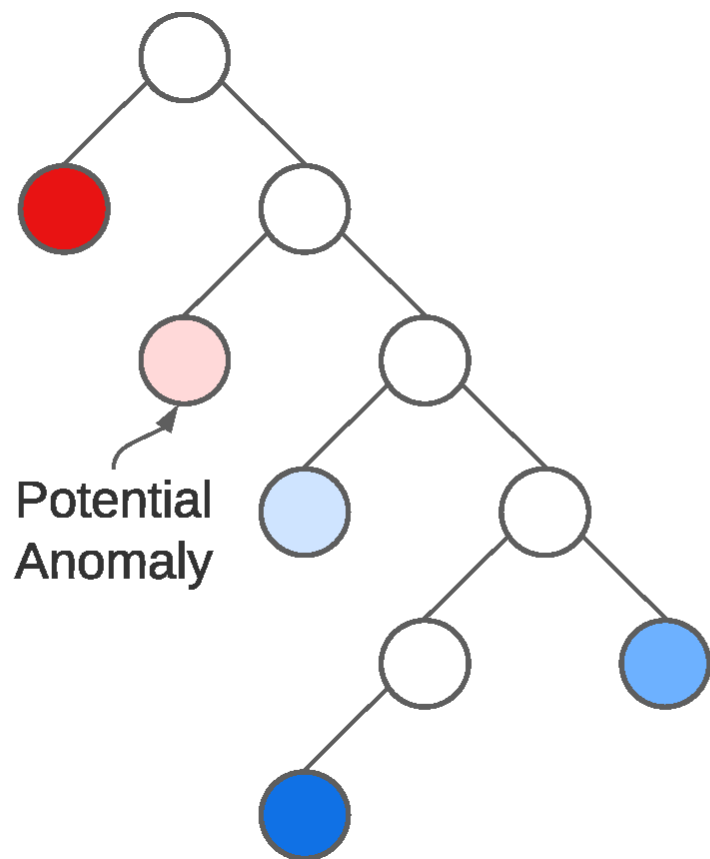
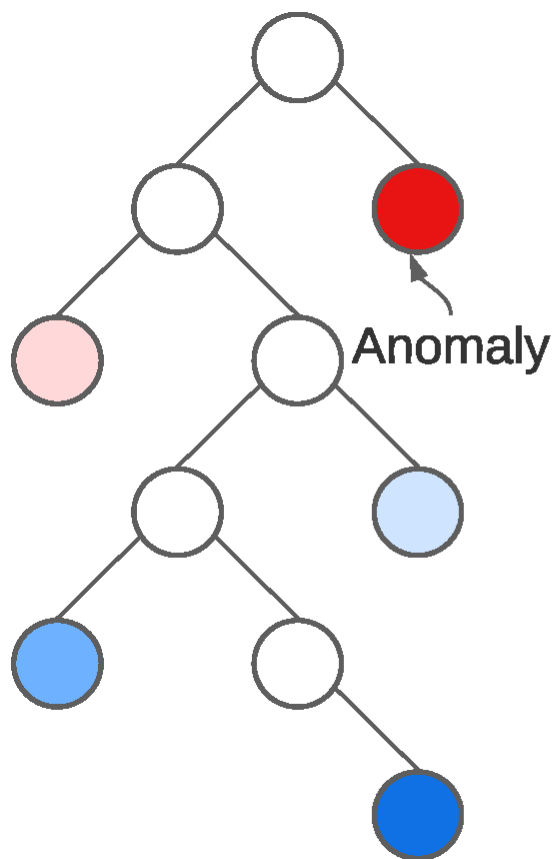
1. Medição de Similaridade: O t-SNE mede a similaridade entre pares de pontos de dados de alta dimensão. Ele usa uma distribuição de probabilidade para representar quão semelhantes são os pontos uns aos outros;
2. Projeção para o Espaço de Baixa Dimensão: Em seguida, o algoritmo mapeia esses pontos de alta dimensão para um espaço de baixa dimensão, como 2D ou 3D. Ele tenta preservar as similaridades entre pontos, de modo que pontos semelhantes ainda estejam próximos no espaço de baixa dimensão;
3. Otimização: O t-SNE otimiza sua projeção de forma iterativa, ajustando as posições dos pontos no espaço de baixa dimensão para minimizar a diferença entre as distribuições de probabilidade de similaridade na alta e baixa dimensão.

Algoritmos (Detecção de Anomalias)

Algoritmo	<i>Isolation Forest</i>	One-Class SVM
Funcionamento	Isola anomalias usando árvores de decisão	Cria um limite que contém a maioria dos dados
Vantagens	Eficiente para dados com muitas dimensões	Pode identificar anomalias locais
Desvantagens	Limitado a identificar anomalias globais	Sensível à escolha de hiperparâmetros

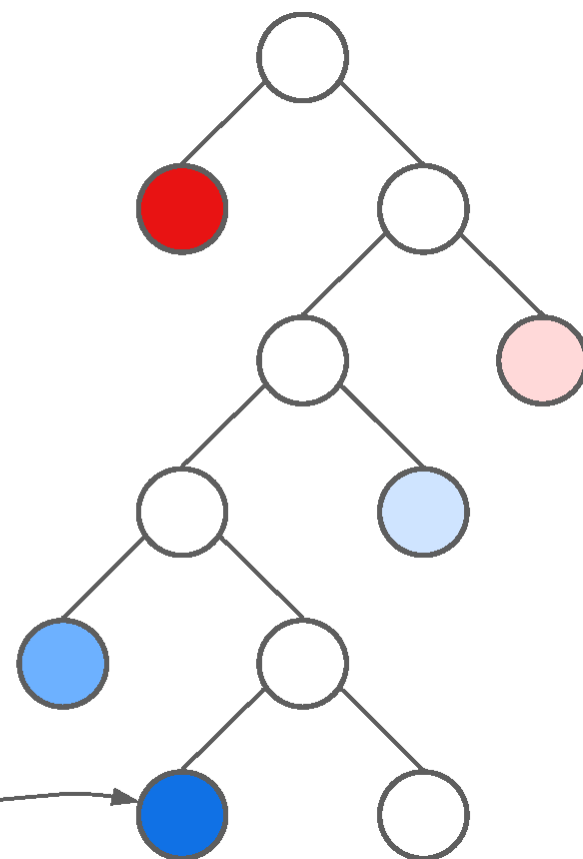
Isolation Forest

1. Construção de Árvores Aleatórias: O algoritmo cria várias árvores de decisão binárias de forma aleatória. Cada árvore é construída selecionando aleatoriamente um subconjunto de características e, em seguida, dividindo os dados em dois ramos com base em uma característica escolhida aleatoriamente;
2. Isolamento de Anomalias: Durante a construção de cada árvore, o algoritmo tenta isolar as amostras de anomalias em ramos mais curtos, porque as anomalias devem ser alcançadas mais rapidamente do que as amostras normais em árvores de decisão;
3. Pontuação de Anomalia: Para detectar anomalias em um novo conjunto de dados, o algoritmo calcula uma pontuação de anomalia para cada amostra. Essa pontuação é baseada em quantas vezes uma amostra foi isolada em ramos curtos em todas as árvores. Quanto mais isolada uma amostra, maior será a sua pontuação de anomalia.



.....

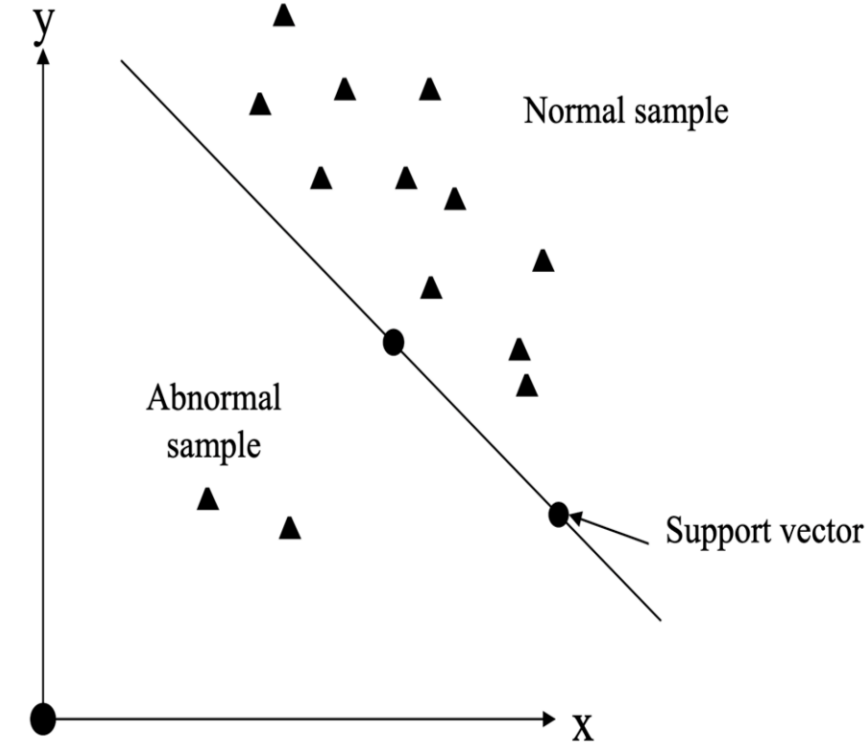
Normal Instance



One-Class SVM

1. Aprendizado com Dados de Uma Classe: O One-Class SVM é treinado apenas com exemplos da classe majoritária (classe normal). O objetivo é encontrar um limite (hiperplano) que encapsule a maioria dos exemplos dessa classe;

2. Detecção de Anomalias: Quando o modelo é alimentado com novos dados, ele avalia a distância de cada ponto de dados ao hiperplano aprendido durante o treinamento. Se a distância de um ponto ao hiperplano for maior que um certo limite (chamado de margem), ele é considerado uma anomalia.



Algoritmos (Regras de Associação)

Algoritmo	Apriori
Funcionamento	Extrai regras de associação a partir de dados de transações
Vantagens	Pode encontrar associações frequentes em grandes conjuntos de dados
Desvantagens	Pode gerar muitas regras algumas não úteis

1. Suporte: O Apriori usa uma métrica chamada "suporte" para identificar os itens frequentes. O suporte de um item é a porcentagem de transações que contém esse item. A ideia é que itens com suporte acima de um limite mínimo sejam considerados frequentes;
2. Geração de Conjuntos Candidatos: O algoritmo começa com itens individuais e gradualmente gera conjuntos de itens maiores (itens frequentes) por meio da criação de conjuntos candidatos. Isso é feito aplicando-se regras que exploram a propriedade do "apriori", que afirma que se um conjunto de itens não for frequente, seus subconjuntos também não serão;
3. Filtragem por Suporte: Depois de gerar todos os conjuntos candidatos, o algoritmo calcula o suporte de cada conjunto e retira aqueles que não atendem ao limite mínimo de suporte;
4. Regra de Associação: O próximo passo é gerar regras de associação entre os itens frequentes. Uma regra de associação é uma expressão condicional do tipo "Se {A} então {B}", indicando que a presença do conjunto de itens A frequentemente leva à presença do conjunto de itens B.

Métrica	Fórmula	Descrição
Suporte (Support)	$\text{Suporte}(A \rightarrow B) = (\text{Número de transações contendo A e B}) / (\text{Número total de transações})$	Mede a frequência com que a regra é verdadeira no conjunto de dados.
Confiança (Confidence)	$\text{Confiança}(A \rightarrow B) = (\text{Número de transações contendo A e B}) / (\text{Número de transações contendo A})$	Mede a probabilidade condicional de que o item B seja comprado dado que o item A foi comprado.
Lift	$\text{Lift}(A \rightarrow B) = (\text{Suporte}(A \rightarrow B)) / (\text{Suporte}(A) * \text{Suporte}(B))$	Indica o quão mais frequentemente os itens A e B são comprados juntos em comparação com o que seria esperado se eles fossem comprados independentemente.
Leverage	$\text{Leverage}(A \rightarrow B) = (\text{Suporte}(A \rightarrow B)) - (\text{Suporte}(A) * \text{Suporte}(B))$	Mede a diferença entre a frequência observada de coocorrência de A e B e a frequência esperada se eles fossem independentes.
Convicção (Conviction)	$\text{Conviction}(A \rightarrow B) = (1 - \text{Suporte}(B)) / (1 - \text{Confiança}(A \rightarrow B))$	Mede o grau em que a ocorrência de B é independente de A, considerando o suporte.