Algoritmos Não Supervisionados – Tópicos Complementares

ROCKETSEAT

Plataforma completa de aprendizado contínuo em programação.

#BoostingPeople

rocketseat.com.br

Algoritmos Não Supervisionados

Tópicos Complementares

O objetivo deste módulo é apresentar algumas técnicas complementares que podem ser empregadas em algoritmos não supervisionados, tais como clusterização usando GMM e detecção de anomalias.

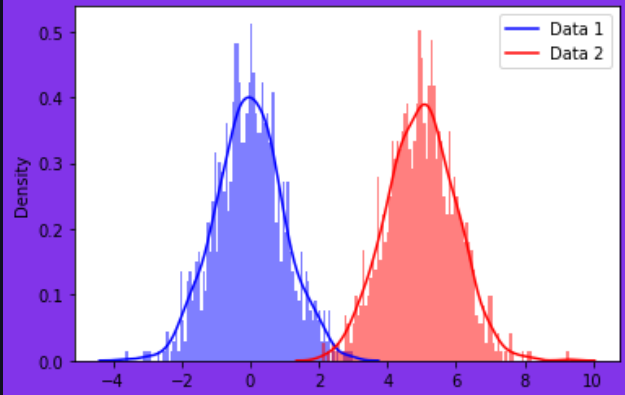
Agenda

- GMM (Gaussian Mixture Models)
- Detecção de Anomalias

GMM (Gaussian Mixture Models)

Gaussian Mixture Models (GMMs) são modelos probabilísticos utilizados em problemas de agrupamento (clusterização) de dados, que assumem que os dados são gerados a partir de uma mistura de várias distribuições gaussianas.

Cada componente da mistura (cluster) é uma distribuição normal (gaussiana), e a distribuição total é uma combinação ponderada dessas distribuições.

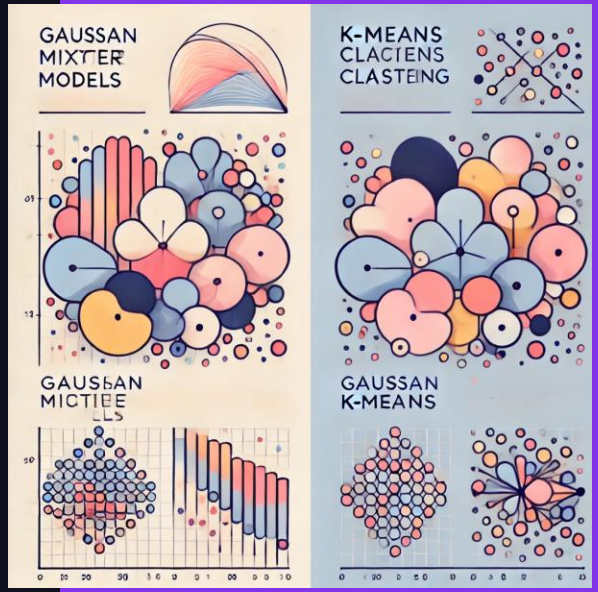


GMM (Gaussian Mixture Models)

GMM possuem certa similaridade com o algoritmo K-Means, também usado para clusterização, porém podemos dizer que GMM são uma versão probabilística do K-Means.

Porém, comparado ao GMM, o K-Means possui algumas limitações como:

- Assume que os clusters são esféricos e de tamanho similar, o que não é sempre verdade em casos reais
- É um método de hard cluster, o que significa que cada ponto de dados é designado para apenas um cluster

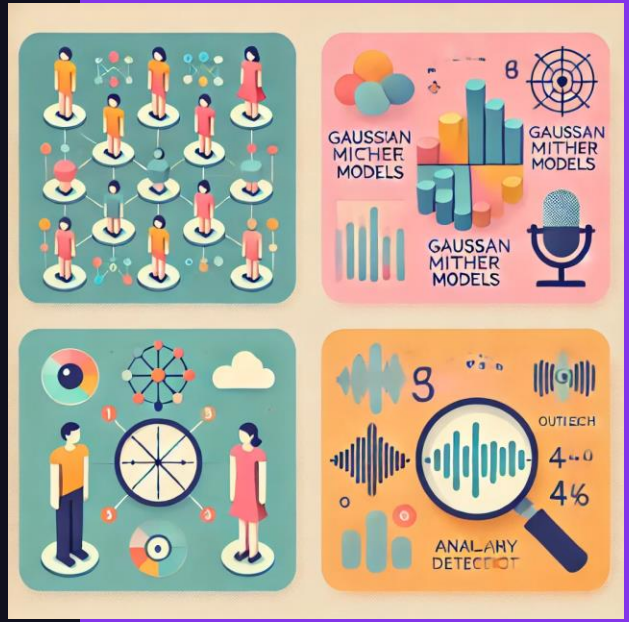


GMM (Gaussian Mixture Models)

Casos de Uso

Agrupamento de Dados (Clustering): GMMs são usados para agrupar dados em subconjuntos homogêneos, onde cada grupo é modelado por uma distribuição gaussiana diferente.

Modelagem de Distribuições Complexas: São utilizados para modelar distribuições de dados que são multi-modais ou têm formas complexas que não podem ser bem representadas por uma única distribuição gaussiana.



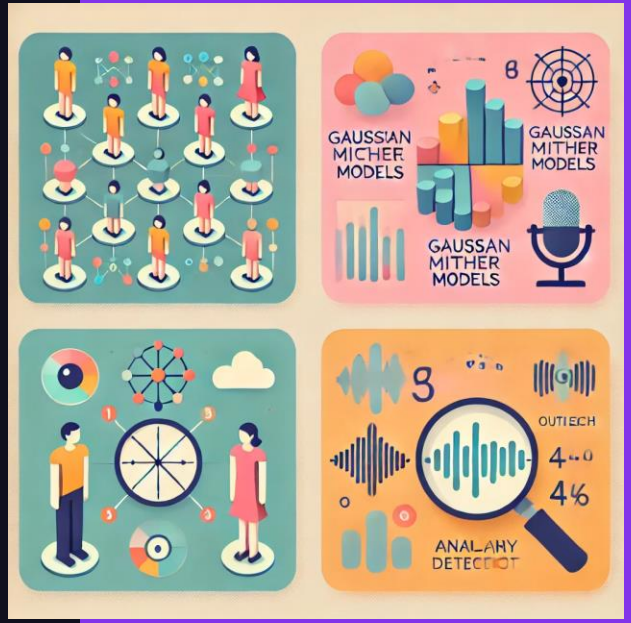
GMM (Gaussian Mixture Models)

Casos de Uso

Segmentação de Imagens: Na análise de imagens, GMMs são aplicados para segmentar uma imagem em diferentes regiões com características semelhantes.

Reconhecimento de Padrões e Fala: GMMs são usados em reconhecimento de fala para modelar as características acústicas de diferentes fonemas.

Detecção de Anomalias: Utilizados para identificar comportamentos anômalos ou raros em conjuntos de dados, comparando as observações com a distribuição modelada.



GMM (Gaussian Mixture Models)

Vantagens e Desvantagens

Vantagens:

- Alta flexibilidade para modelar dados complexos.
- Capacidade de lidar com dados de diferentes formas e orientações.
- Aplicável em diversas áreas como visão computacional, processamento de fala e detecção de anomalias.

Desvantagens:

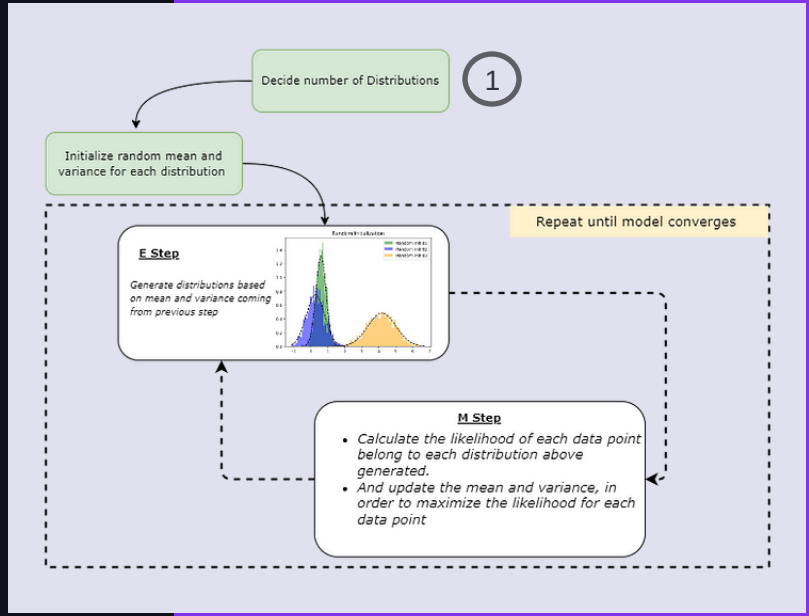
- Pode convergir para um ótimo local dependendo da inicialização dos parâmetros.
- Sensível à escolha do número de componentes (K).
- Computacionalmente intensivo para grandes conjuntos de dados.

GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 1: Definir a quantidade de clusters

Para definir a quantidade de clusters, temos um desafio similar ao que ocorre no K-Means. Desta forma, podemos definir este parâmetro k com base no conhecimento do domínio do problema a ser modelado ou podemos usar uma métrica para validar o resultado do modelo. No caso dos GMM, a métrica mais utilizada é chamada de BIC (Bayesian Information Criteria)



GMM (Gaussian Mixture Models)

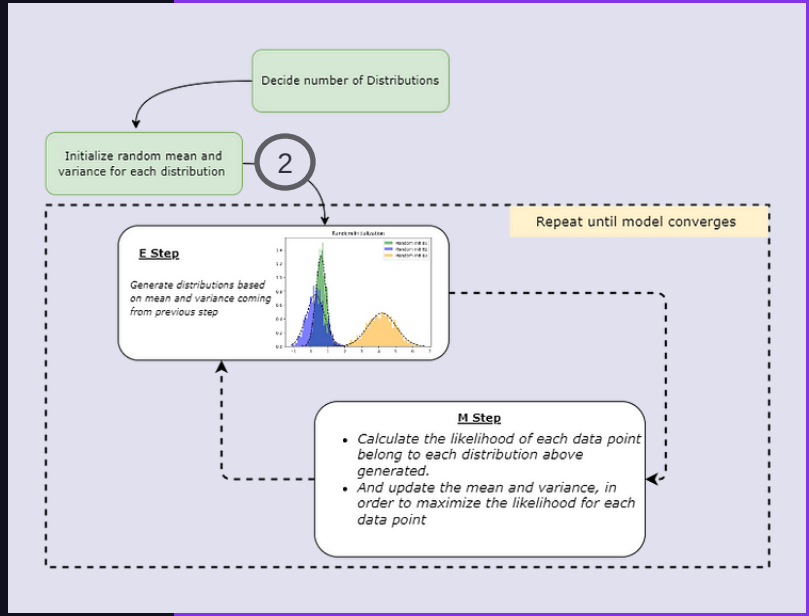
Como funciona o algoritmo GMM?

Passo 2: Inicializa Média, Covariância e Pesos

Média: Centro de cada Gaussiana. Inicializa de forma aleatória.

Covariância: Forma e orientação de cada Gaussiana. Inicializa de forma aleatória.

Peso: É a proporção que cada componente gaussiano contribui para a mistura total. Inicializa com base no parâmetro k. Ex: Se k = 3, cada componente é inicializado com 1/3.



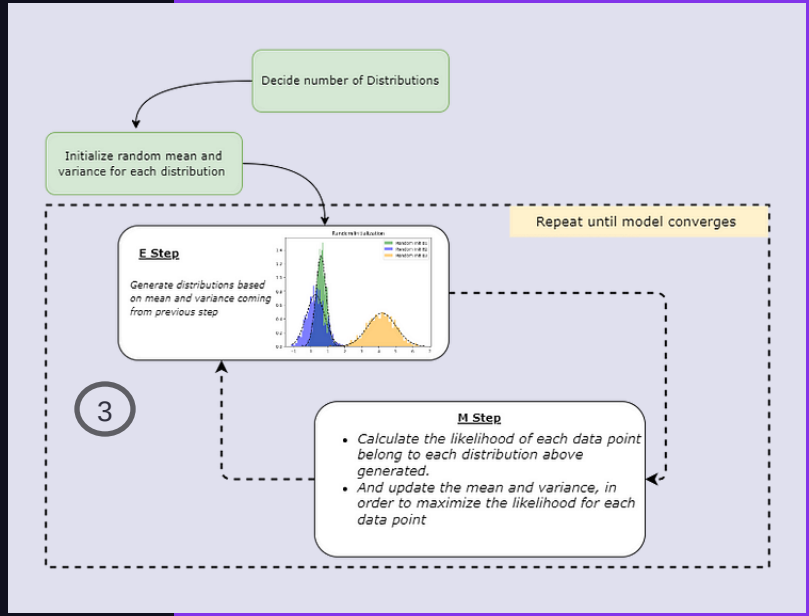
GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 3: Usar o algoritmo EM (Expectation-Maximization)

O algoritmo EM (Expectation-Maximization) é uma técnica usada para encontrar parâmetros em modelos estatísticos, como Gaussian Mixture Models (GMMs), onde é usado para ajustar os parâmetros das distribuições gaussianas (médias, covariâncias e pesos) para melhor modelar os dados observados.

A cada iteração, o modelo se ajusta para melhorar a representação dos dados, resultando em uma melhor compreensão da estrutura subjacente dos dados.



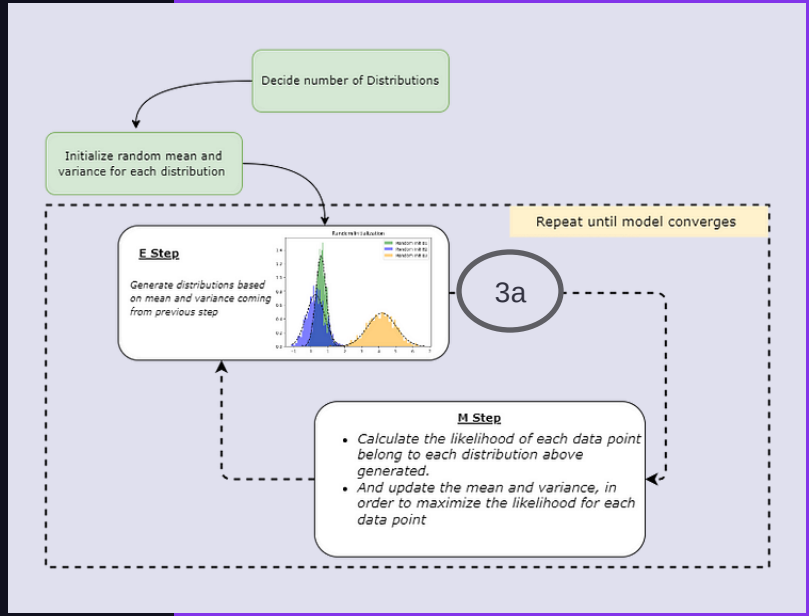
GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 3a: Etapa Expectation (E)

O passo E do algoritmo EM calcula a responsabilidade de cada componente da mistura por cada ponto de dado, usando as estimativas atuais dos parâmetros (média, covariância e peso).

Essas responsabilidades indicam a probabilidade de que cada ponto de dado pertença a cada componente, e são usadas na próxima etapa do algoritmo (passo M) para atualizar as estimativas dos parâmetros do modelo.



GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 3a: Etapa Expectation (E)

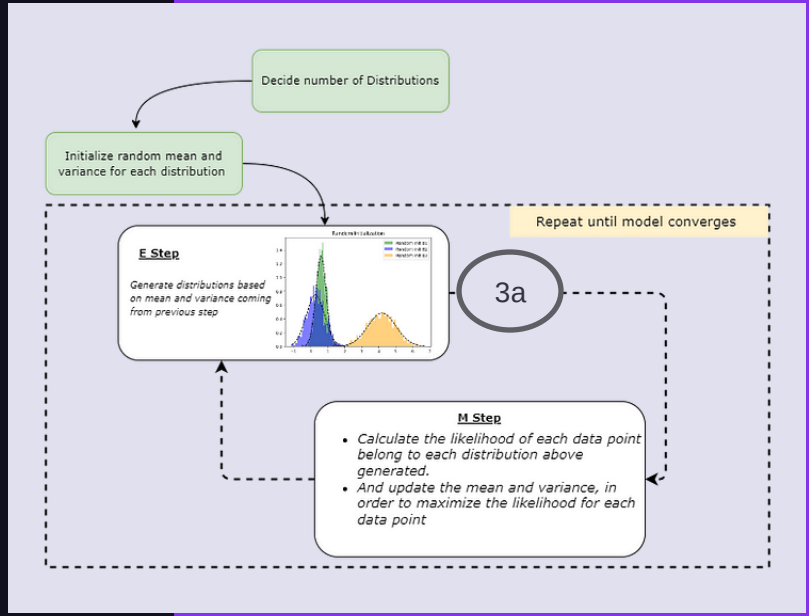
A fórmula abaixo é usada para a etapa E (Expectation)

$$r_{ic} = \frac{\pi_c N(x_i|\mu_c, \sum_c)}{\sum_{k=1}^{k} \pi_k \, N(x_i|\mu_k, \sum_k)}$$

π_c = Peso da distribuição no passo anterior

$N(x_i|\mu_c, \sum_c)$ = Função de Densidade de Probabilidade (PDF)

$\sum_{k=1}^{k} \pi_k \, N(x_i|\mu_k, \sum_k)$ = Soma ponderada das densidades para todos os componentes

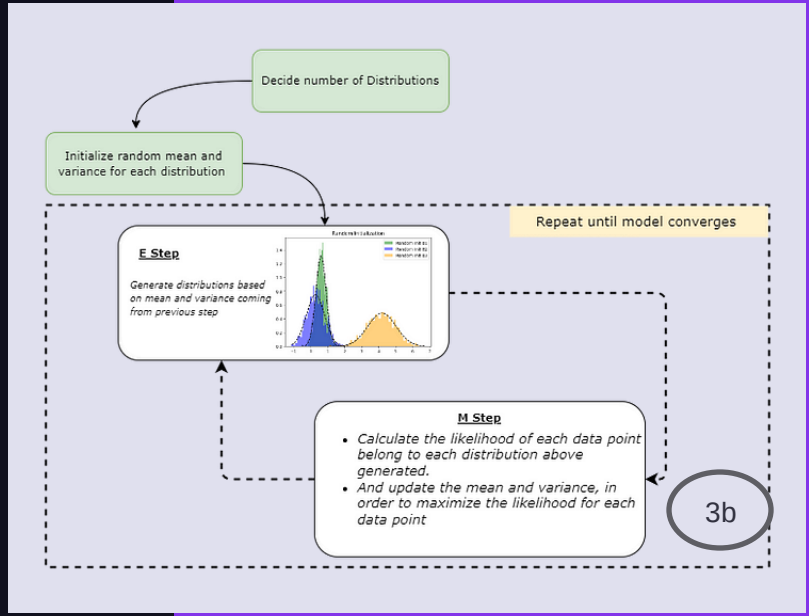


GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 3b: Etapa Maximization (M)

O objetivo do passo M do algoritmo EM é atualizar os parâmetros do modelo (médias, covariâncias e pesos das distribuições) de forma a maximizar a probabilidade dos dados observados, dadas as responsabilidades calculadas na Etapa de Expectativa.



GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 3b: Etapa Maximization (M)

Média

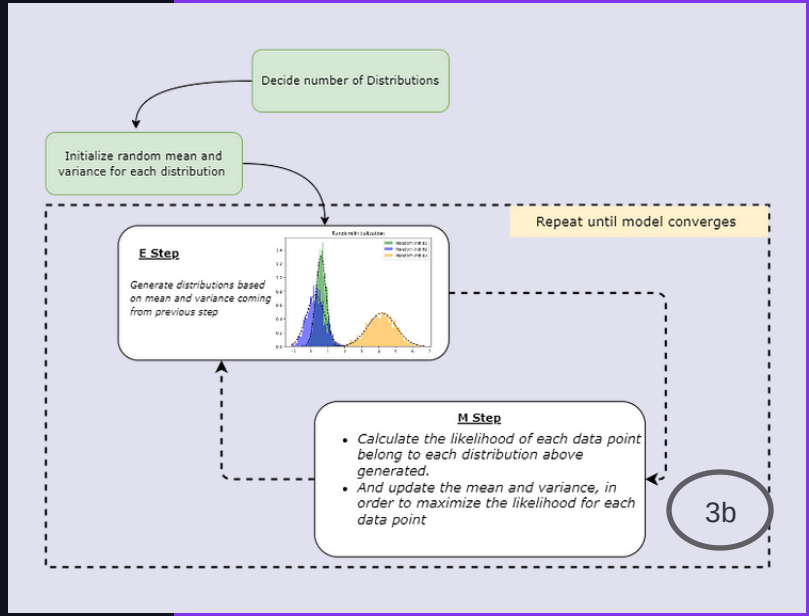
$$\mu_c = \frac{\sum_{i=1}^{m} r_{ic} x_i}{\sum_{i=1}^{m} r_{ic}}$$

Covariância

$$\sum_c = \frac{\sum_{i=1}^{m} r_{ic} (x_i - \mu_c)^2}{\sum_{i=1}^{m} r_{ic}}$$

Peso

$$\pi_c = \frac{\sum_{i=1}^{m} r_{ic}}{m}$$

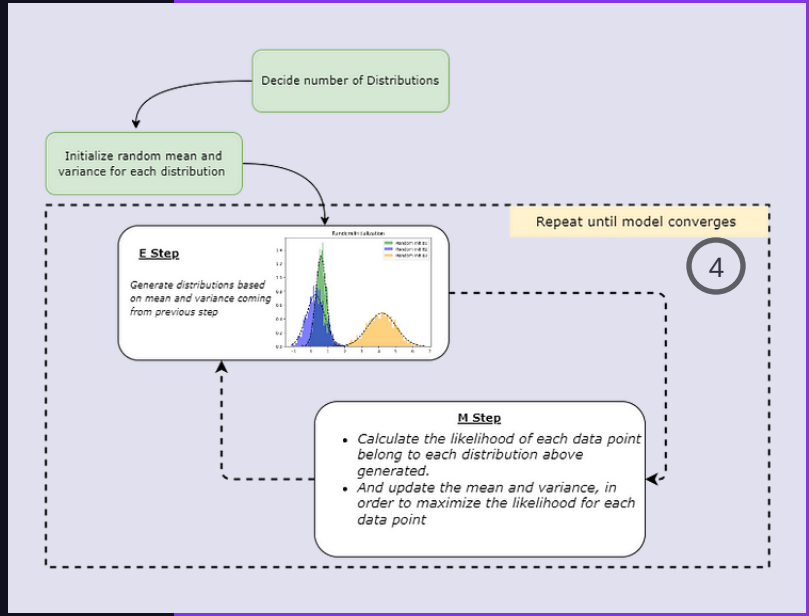


GMM (Gaussian Mixture Models)

Como funciona o algoritmo GMM?

Passo 4: Itere no algoritmo EM até convergir

Execute os passos E e M até que ocorre a convergência do algoritmo, ou seja, que os modelos dos parâmetros não se alterem significativamente de uma iteração a outra.



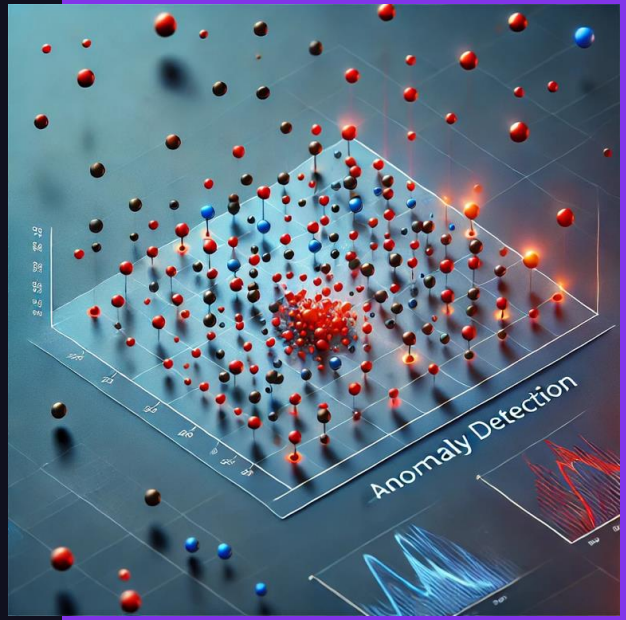
Code Time . . .

rocketseat.com.br

Detecção de Anomalias

A detecção de anomalias é o processo de identificar padrões ou valores que se desviam significativamente do comportamento normal ou esperado em um conjunto de dados. Isso pode incluir valores extremos, outliers, erros de coleta de dados ou ataques mal-intencionados.

A detecção de anomalias é essencial para garantir a qualidade e a confiabilidade dos dados, bem como para detectar e prevenir ameaças cibernéticas.



Detecção de Anomalias

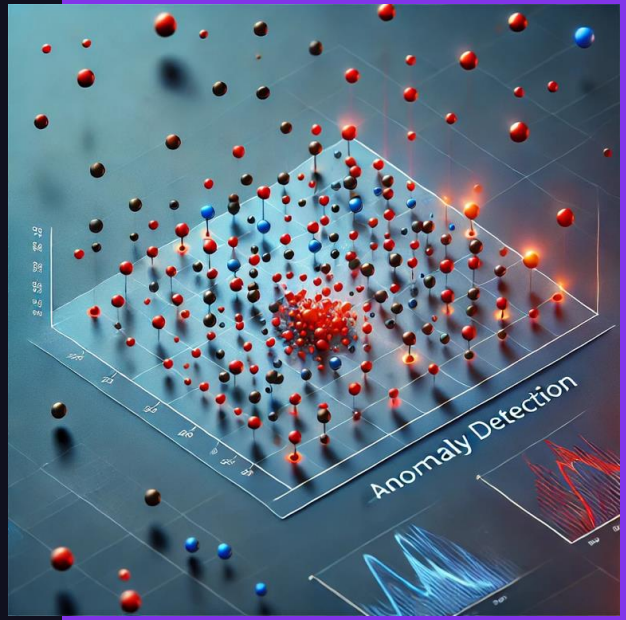
Alguns Casos de Uso

Detecção de Fraude: Identificar transações suspeitas em transações financeiras, comércio eletrônico e outras indústrias.

Monitoramento e análise de tráfego de rede: Detectar ataques cibernéticos como invasões, malware e ransomware.

Controle de qualidade: Detectar defeitos em produtos ou processos industriais.

Segurança: Identificar comportamentos suspeitos em sistemas de segurança, como acesso não autorizado.



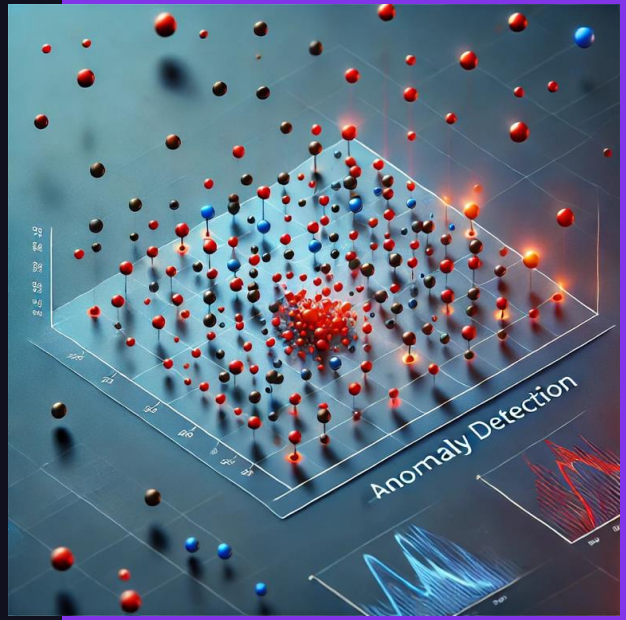
Detecção de Anomalias

Alguns Casos de Uso

Doenças e Anomalias Médicas: Detectar padrões anômalos em exames de sangue ou registros médicos que possam indicar uma condição médica rara ou uma doença.

Anomalias em Séries Temporais: Identificar pontos de dados anômalos em séries temporais, como dados de sensores ou métricas de desempenho.

Análise de Desempenho de Aplicações Web: Identificar picos de tempo de resposta ou falhas no servidor que não correspondem ao comportamento normal da aplicação.



Um passeio pelos algoritmos de detecção de anomalias

DBSCAN

Agrupa pontos de dados em clusters de alta densidade, tratando pontos que não pertencem a nenhum cluster como ruído (anomalias).

Local Outlier Factor (LOF)

Mede a densidade local de um ponto de dados em relação à densidade local de seus vizinhos. Pontos com densidade significativamente menor são considerados anômalos.

Isolation Forest

Baseado em árvores de decisão que isolam pontos de dados particionando repetidamente o espaço de dados aleatoriamente. Pontos que são isolados rapidamente (com menos partições) são considerados anômalos.

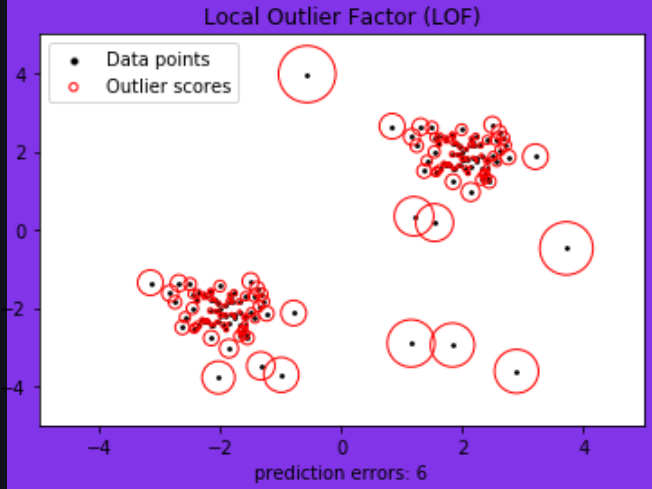
Autoencoders

Redes neurais treinadas para reconstruir dados de entrada. Pontos de dados que não podem ser reconstruídos com precisão (grande erro de reconstrução) são considerados anômalos.

LOF (Local Outlier Factor)

LOF (Local Outlier Factor) é uma técnica de aprendizado de máquina não supervisionado usada para detectar anomalias em conjuntos de dados. Ele compara a densidade de cada ponto de dado com a densidade dos seus vizinhos mais próximos. Se um ponto de dado está em uma região de densidade significativamente menor do que a de seus vizinhos, é considerado uma anomalia.

O LOF é útil em situações onde os dados têm densidades variáveis, permitindo identificar pontos que estão isolados localmente, mesmo em presença de clusters com diferentes densidades.



LOF (Local Outlier Factor)

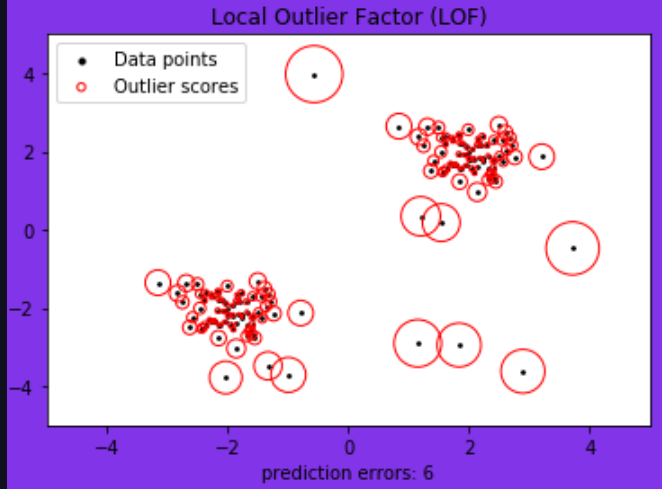
Como funciona o algoritmo LOF?

Passo 1: Definir a quantidade de vizinhos (k)

Define o número de vizinhos (k) a serem considerados para cada ponto de dados.

Determinar este número é crucial e pra isso podemos realizar testes empíricos com diversos valores, análise de estabilidade e a própria visualização num gráfico de dispersão.

Além disso, podemos realizar uma comparação da densidade entre pontos anômalos e normais, para avaliar as diferenças entre os mesmos.



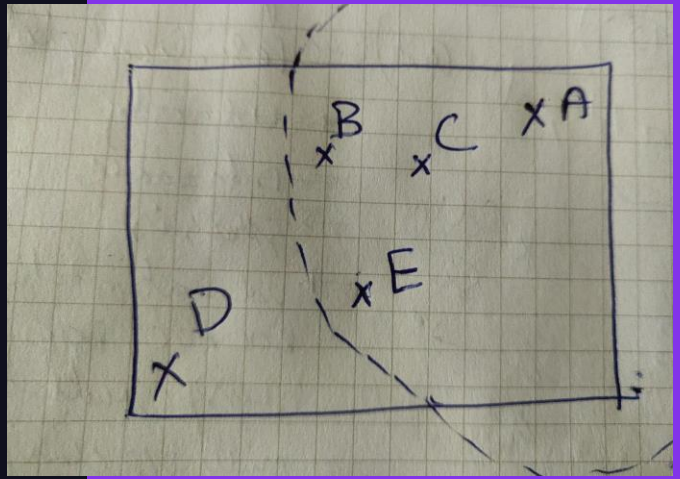
LOF (Local Outlier Factor)

Como funciona o algoritmo LOF?

Passo 2: Cálculo da Distância k-Distância

K-distância é a distância do ponto (A) ao seu (k)-ésimo vizinho mais próximo.

Como exemplo e seguindo a figura ao lado, considere que escolhemos k = 3 e que precisamos calcular a k-distância do ponto A. Na figura, podemos dizer a que a k-distância de A é a distância entre o ponto A e o terceiro ponto (vizinho) mais próximo, que seria o ponto E.



LOF (Local Outlier Factor)

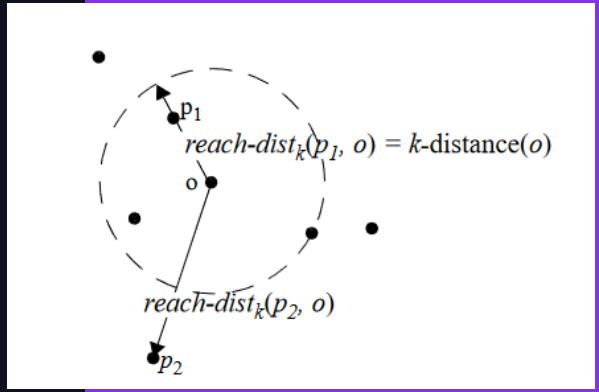
Como funciona o algoritmo LOF?

Passo 3: Cálculo do Alcance k-Distância (Reachability Distance ou RD)

Para dois pontos (A) e (B), o alcance k-distância é definido como:

$$RD_k(A, B) = \max(kdistancia(B), distancia(A, B))$$

A distância de alcance é o maior valor entre a k-distância de (B) e a distância direta entre (A) e (B). Isso garante que a distância de alcance não seja menor do que a k-distância de (B), evitando valores muito pequenos que poderiam distorcer a análise de densidade local. O uso da distância de alcance também ajuda a normalizar as distâncias e a evitar que outliers influenciem indevidamente os cálculos de densidade local.



LOF (Local Outlier Factor)

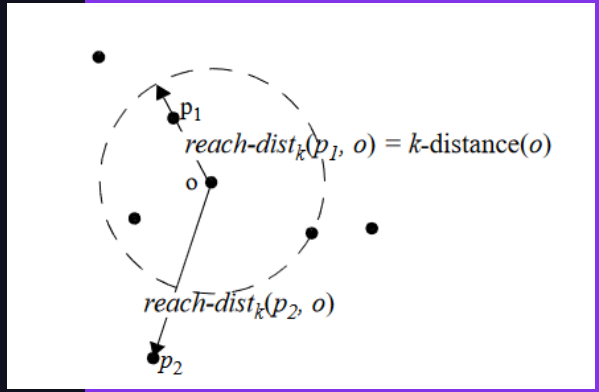
Como funciona o algoritmo LOF?

Passo 4: Cálculo da Densidade de Alcance Local (Local Reachability Density ou LRD)

A densidade de alcance local é a inversa da média do alcance k-distância para todos os (k) vizinhos de (A) e representa quão "densamente" um ponto de dado está rodeado por seus vizinhos.

$$LRD_k(A) = \frac{1}{\frac{1}{k}\sum B \in kNN(A) RD_k(A,B)}$$

Onde $kNN(A)$ é o número de vizinhos mais próximos de (A).



LOF (Local Outlier Factor)

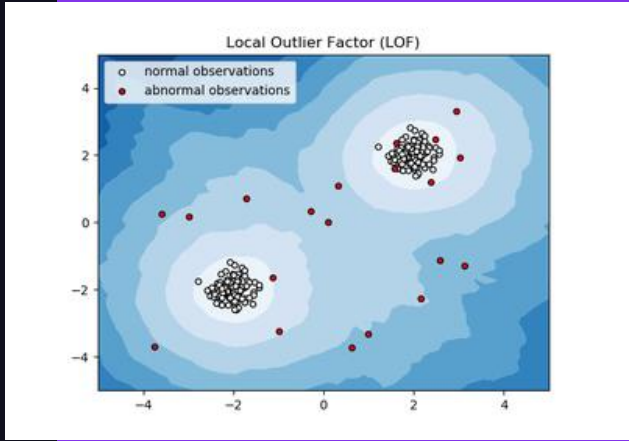
Como funciona o algoritmo LOF?

Passo 5: Cálculo do LOF

Compara a densidade local do ponto (A) com a densidade local de seus vizinhos. É calculado como a média das razões das densidades locais dos vizinhos de (A) pela densidade local de (A).

$$LOF_k(A) = \frac{1}{|kNN(A)|} \sum B \in kNN(A) \frac{LRD_k(B)}{LRD_{k(A)}}$$

Onde $kNN(A)$ é o número de vizinhos mais próximos de (A)



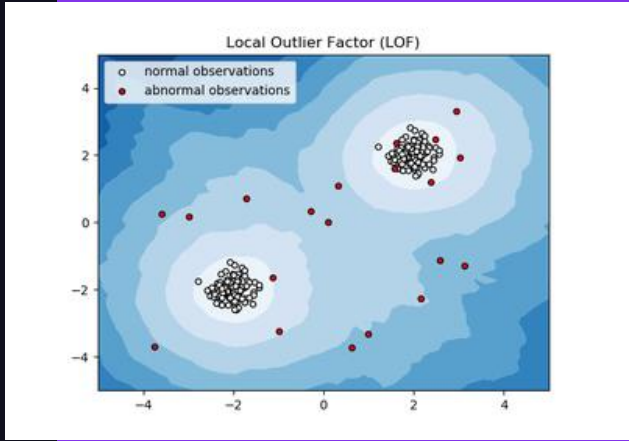
LOF (Local Outlier Factor)

Como funciona o algoritmo LOF?

Passo 5: Cálculo do LOF

Como interpretar o valor de LOF:

LOF ≈ 1: O ponto está em uma região de densidade semelhante à de seus vizinhos (não é uma anomalia).
LOF > 1: O ponto está em uma região de densidade menor do que a de seus vizinhos (é uma anomalia). Quanto maior o valor de LOF, mais provável é que o ponto seja uma anomalia.



Code Time . . .

rocketseat.com.br