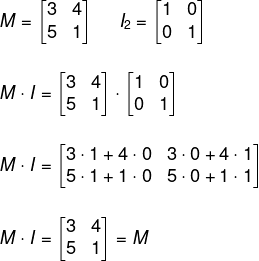
**Produto escalar de tuplas**

Texto, Aplicativo

Descrição gerada automaticamente

**Multiplicação de matrizes**



**Combinação linear**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Singular Value Decomposition**

A Singular Value Decomposition (SVD) é uma técnica de fatoração de matrizes amplamente utilizada em álgebra linear e aplicações de ciência de dados. Dada uma matriz AAA de dimensões m×n, a SVD a decompõe em três matrizes:

A=UΣV^T

* **U**: Uma matriz ortogonal m×m, cujas colunas são os vetores próprios da matriz AATAA^TAAT.
* **Σ**: Uma matriz diagonal mxn com valores singulares não-negativos ordenados em ordem decrescente.
* **V^T**: A transposta de uma matriz ortogonal n×n cujas colunas são os vetores próprios da matriz A^TA.

**Interpretação das Matrizes**

1. **Matriz U**:
   * As colunas de UUU são chamadas de "vetores singulares à esquerda" de A.
   * Representam as direções principais no espaço de entrada.
2. **Matriz Σ**:
   * Contém os "valores singulares" no seu eixo diagonal.
   * Os valores singulares representam a magnitude das direções principais.
3. **Matriz V^T**:
   * As colunas de V (ou as linhas de V^T) são chamadas de "vetores singulares à direita" de A.
   * Representam as direções principais no espaço de saída.

**Como SVD é Computado?**

Para calcular a SVD de uma matriz AAA:

**1. Calcula-se A^TA e AA^T**.

**2.Determina-se os autovalores e autovetores de AA^T**:

* + Os autovetores de **AA^T** formam as colunas da matriz U.

**3.Determina-se os autovalores e autovetores de A^TA**:

* + Os autovetores de **A^TA** formam as colunas da matriz V.

1. **Os autovalores de A^TA e AA^T** são os quadrados dos valores singulares.
2. **A matriz Σ** é formada pelos valores singulares ordenados na diagonal (restante dos elementos são zeros).

**Aplicações Práticas**

* **Redução de Dimensionalidade**: Usado em PCA (Principal Component Analysis).
* **Compressão de Imagens**: Usado para reduzir o tamanho das imagens mantendo a qualidade.
* **Recomendações**: Usado em sistemas de recomendação, como em filtragem colaborativa.
* **Análise de Dados**: Utilizado para encontrar padrões em conjuntos de dados complexos.

**Distância cosseno**

Texto

Descrição gerada automaticamente com confiança média

Para identificar outliers em dados transformados por SVD, você pode seguir estas etapas:

1. **Redução de Dimensionalidade:** Reduza a dimensionalidade dos dados usando apenas os principais componentes (valores singulares mais altos). Isso ajuda a capturar a maior parte da variabilidade nos dados enquanto elimina o ruído.
2. **Reconstrução dos Dados:** Reconstrua os dados a partir dos componentes principais. Compare os dados reconstruídos com os dados originais para calcular a "distância de reconstrução" (ou erro de reconstrução) para cada ponto de dados.
3. **Cálculo do Erro de Reconstrução:** O erro de reconstrução é dado pela diferença entre os dados originais e os dados reconstruídos. Pontos com um erro de reconstrução significativamente alto são candidatos a outliers.
4. **Definição de Limite para Outliers:** Use métodos estatísticos para definir um limite (threshold) para o erro de reconstrução. Pontos de dados que têm um erro de reconstrução acima desse limite podem ser considerados outliers.

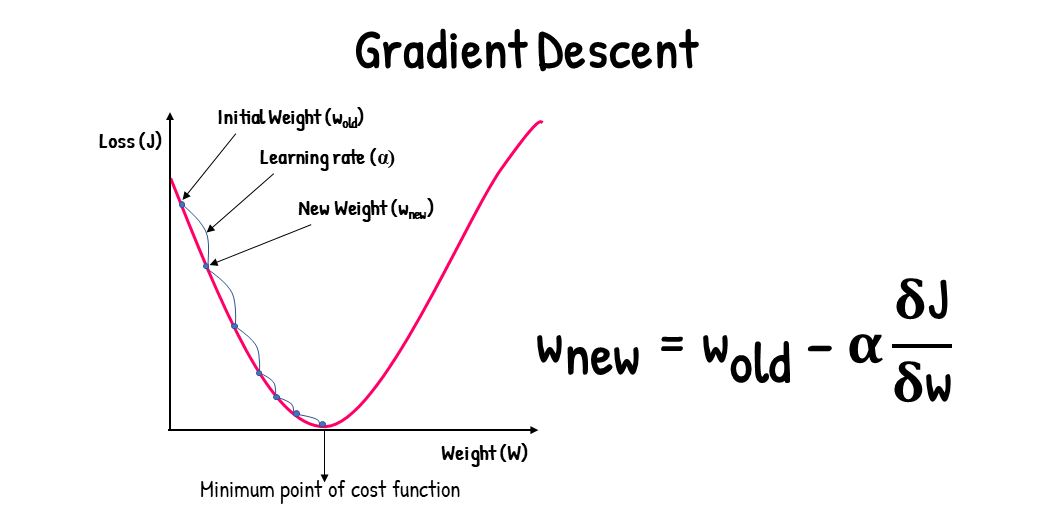
**Clustering**

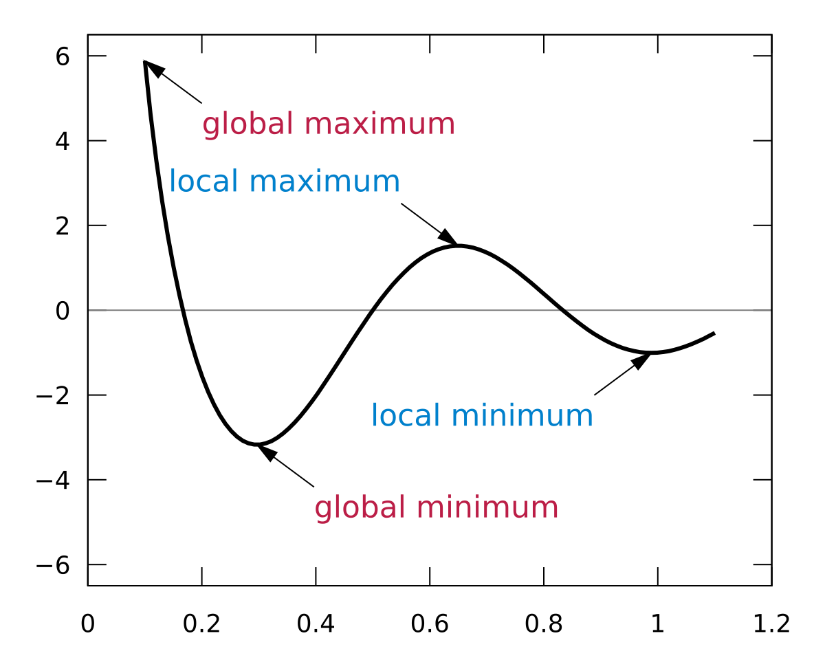
K-Means -> número de clusters criados

Silhouette -> método de calcular o erro dos clusters, quanto mais próximo de 1 melhor.

Possíveis erros: número de clusters errados e falta de normalização dos dados (um dado está em milhar e outro em milhão)

**Otimização**





**Equação normal**

Lousa branca com texto preto sobre fundo branco

Descrição gerada automaticamente

X incrementado.

**Feature Engineering**

Feature engineering é o processo de usar o conhecimento do domínio dos dados para criar variáveis (features) que ajudem a melhorar a performance de algoritmos de aprendizado de máquina. É uma etapa crucial no desenvolvimento de modelos preditivos, pois as features que você usa influenciam diretamente a capacidade do modelo de fazer previsões precisas.

Aqui estão algumas etapas e técnicas comuns de feature engineering:

**1. Seleção de Features**

* **Remoção de Redundâncias**: Eliminar features que são altamente correlacionadas entre si, para evitar multicolinearidade.
* **Relevância**: Escolher apenas as features que têm uma correlação significativa com o target.

**2. Criação de Novas Features**

* **Transformações Matemáticas**: Aplicar operações matemáticas simples, como somas, diferenças, multiplicações ou divisões.
* **Interações entre Features**: Criar novas features que representem interações entre features existentes.
* **Agregações**: Para dados temporais ou espaciais, criar features agregadas como médias, somas, máximos e mínimos.

**3. Transformações de Dados**

* **Normalização e Padronização**: Ajustar a escala das features para que tenham uma distribuição uniforme.
* **Transformações Logarítmicas ou Exponenciais**: Usar quando há uma grande diferença de escala ou distribuição log-normal.

**4. Codificação de Dados Categóricos**

* **One-Hot Encoding**: Transformar variáveis categóricas em uma série de variáveis binárias.
* **Label Encoding**: Converter categorias em números inteiros.
* **Target Encoding**: Substituir categorias pela média do target para aquela categoria.

**5. Tratamento de Valores Faltantes**

* **Imputação Simples**: Substituir valores faltantes por média, mediana ou moda.
* **Model-Based Imputation**: Usar algoritmos de aprendizado de máquina para prever os valores faltantes.

**6. Feature Extraction**

* **Análise de Componentes Principais (PCA)**: Reduzir a dimensionalidade dos dados, mantendo a maior parte da variância.
* **Feature Hashing**: Transformar grandes volumes de dados categóricos em um espaço de dimensões fixas.