**Caderno de Estudo – Ciência de Dados**

**Principais termos da Ciência de Dados**

**Média ()**

É uma medida estatística que representa o valor central de um conjunto de números. Ela é calculada somando todos os valores do conjunto e dividindo pelo número total de observações.

**Variância (/)**

É uma medida estatística que indica o quanto os valores de um conjunto de dados se desviam da média. Quanto maior a variância, mais dispersos são os dados; quanto menor, mais próximos eles estão da média. (Existe diferença na fórmula entre a variância amostral e a populacional)

**Desvio Padrão** **( /)**

O desvio padrão é a raiz quadrada da variância. Ele traz a unidade de medida de volta à mesma unidade dos dados originais, o que facilita a interpretação. O desvio padrão mostra o quanto, em média, os dados se afastam da média, e está em unidades dos dados, tornando-o mais intuitivo.

**Auto Covariância ( )**

A auto covariância mede a relação linear entre os valores de uma série temporal e seus próprios valores defasados (lags). Ou seja, ela indica como os valores passados influenciam os valores futuros da série.

**Séries Temporais**

**Definição**

Uma série temporal é uma sequência de observações registradas em momentos específicos no tempo.  
Exemplo: vendas diárias de um produto, temperaturas mensais ou valores de ações ao longo de semanas.

**Estrutura**

**Observações (yt​​)**: Valores medidos em momentos t.

**Tempo (t)**: Pode ser diário, semanal, mensal, anual ou em intervalos irregulares.

**Utilidade e Quando Usar Séries Temporais**

Séries temporais são úteis em cenários onde os dados variam ao longo do tempo e existe a necessidade de entender padrões, realizar análises ou prever eventos futuros. Elas são empregadas em diversas áreas devido à sua capacidade de capturar tendências, sazonalidades e dependências temporais.

A análise de séries temporais é essencial para previsão, identificação de padrões, monitoramento, tomada de decisão estratégica e otimização de processos.

A previsão permite estimar valores futuros com base em dados históricos, sendo útil para demandas de estoque, vendas, produção de energia e preços de mercado. Já a análise de padrões identifica tendências, sazonalidades e ciclos, auxiliando no entendimento de variações de vendas, padrões climáticos e comportamento do consumidor.

O monitoramento detecta anomalias em tempo real, como falhas em máquinas, fraudes financeiras e variações no tráfego de redes. Na tomada de decisão estratégica, essas informações ajudam no planejamento de campanhas, precificação dinâmica e infraestrutura urbana.

Por fim, a otimização permite ajustar processos automaticamente, como controlar a produção industrial, gerenciar o consumo de energia e otimizar transportes públicos.

**Componentes de Séries Temporais**

As séries temporais podem ser decompostas em quatro componentes principais:

**1. Tendência (Trend)**

Representa o movimento de longo prazo na série. Exemplo: Crescimento consistente no número de vendas ao longo dos anos.

**2. Sazonalidade (Seasonality)**

Padrões que se repetem em intervalos regulares de tempo. Exemplo: Aumento de vendas no Natal.

**3. Ciclo (Cycle)**

Flutuações que ocorrem devido a fatores econômicos ou outros ciclos (não tem a ver com o tempo). Exemplo: Alta e baixa de um mercado financeiro em períodos não fixos.

**4. Resíduo (Noise)**

Variação aleatória ou irrelevante na série. Exemplo: Pequenas flutuações inesperadas em vendas diárias.

**Tipos de Séries Temporais**

1. **Univariada**

Contém apenas uma variável observada ao longo do tempo. O foco é entender e prever essa única variável com base em seus próprios valores passados. Exemplo: Preço diário de uma ação, temperatura média mensal, vendas de um produto ao longo do ano.

1. **Multivariada**

Envolve múltiplas variáveis inter-relacionadas, analisadas simultaneamente. Essas variáveis podem influenciar umas às outras, tornando a modelagem mais complexa. Exemplo: Preço de uma ação considerando também taxas de juros e volume de negociação, previsão de demanda levando em conta fatores climáticos e promoções.

**Estacionaridade**

Um conjunto de dados estacionário é aquele cujas propriedades estatísticas não mudam ao longo do tempo. Isso significa que a média, a variância e a auto covariância do processo são constantes. Em outras palavras, uma série temporal estacionária não apresenta tendências ou sazonalidades de longo prazo, tornando-a mais previsível e mais fácil de modelar.

**Modelos e Técnicas de Análise**

**Modelos Estatísticos**

1. **Média Móvel (Moving Average)**

A Média Móvel suaviza flutuações em uma série temporal, facilitando a identificação de tendências. Ela calcula a média de um número fixo de valores anteriores, eliminando variações aleatórias.

1. **ARIMA (AutoRegressive Integrated Moving Average)**

O modelo ARIMA combina três componentes principais para modelar séries temporais:

**AR (AutoRegressive)**: Usa valores passados da própria série para prever futuros.

**I (Integrated)**: Aplica diferenciação para tornar a série estacionária.

**MA (Moving Average)**: Modela os erros da previsão baseando-se em valores passados.

O ARIMA é indicado para séries temporais estacionárias ou que podem ser transformadas em estacionárias. Ele é representado como **ARIMA(p, d, q)**, onde:

**p** = Número de defasagens na parte autoregressiva.

**d** = Número de diferenciações para tornar a série estacionária.

**q** = Número de termos de média móvel.

**Exemplo de Aplicação**: Previsão de demanda por produtos ou previsão de inflação.

1. **SARIMA (Seasonal ARIMA)**

O **SARIMA** é uma extensão do ARIMA que incorpora sazonalidade, ou seja, padrões que se repetem periodicamente. Ele adiciona quatro novos parâmetros para capturar variações sazonais:

**P** = Ordem da parte autoregressiva sazonal.

**D** = Diferenciação sazonal necessária.

**Q** = Ordem da média móvel sazonal.

**S** = Período da sazonalidade (exemplo: 12 para dados mensais).

O modelo é representado como **SARIMA(p, d, q) × (P, D, Q, S)**.

**Exemplo de Aplicação**: Previsão de vendas sazonais, como consumo de energia elétrica ao longo do ano.

**Etapas Práticas para Análise de Séries Temporais**

**1. Importar e Visualizar os Dados**

A visualização de dados é uma etapa essencial na análise de séries temporais, pois permite identificar padrões, tendências, sazonalidade e anomalias antes da modelagem. Algumas das principais técnicas incluem:

**Gráfico de Linha:** A forma mais comum de visualização, exibindo a evolução dos dados ao longo do tempo. Ajuda a detectar tendências e padrões sazonais.

**Histogramas e Boxplots:** Úteis para analisar a distribuição dos dados e identificar outliers.

**Autocorrelação (ACF) e Parcial (PACF):** Mostram como os valores passados influenciam os valores futuros, auxiliando na escolha de modelos ARIMA/SARIMA.

**Decomposição de Série Temporal:** Separa os componentes da série em tendência, sazonalidade e resíduo, facilitando a interpretação.

**Heatmaps e Gráficos de Dispersão:** Podem ser usados para visualizar padrões sazonais ou relações entre múltiplas variáveis em séries temporais multivariadas.

**2. Analisar a Estacionaridade**

**Visualização Gráfica**

**Gráfico de Linha:** Se houver uma tendência crescente/decrescente ou padrões sazonais evidentes, a série pode não ser estacionária.

**Rolling Statistics:** Calcular e plotar a média e a variância em janelas móveis para ver se se mantêm constantes.

**Testes Estatísticos**

**Teste de Dickey-Fuller Aumentado (ADF)**

Hipótese nula (H0): A série tem raiz unitária (não estacionária).

Hipótese alternativa (H1​): A série é estacionária.

Se o p-valor for menor que 0,05, rejeitamos H0 e concluímos que a série é estacionária.

**Teste KPSS (Kwiatkowski-Phillips-Schmidt-Shin)**

Hipótese nula (H0​): A série é estacionária.

Hipótese alternativa (H1​): A série não é estacionária.

Se o p-valor for menor que 0,05, rejeitamos H0, indicando que a série não é estacionária.

**Teste de Phillips-Perron (PP)**

Similar ao ADF, mas mais robusto a heterocedasticidade.

**Função de Autocorrelação (ACF) e Autocorrelação Parcial (PACF)**

Se a ACF decai lentamente em vez de cair rapidamente para zero, a série pode ser não estacionária.

**Como Tornar uma Série Estacionária?**

Se a série não for estacionária, pode-se aplicar transformações para estabilizar suas propriedades:

**Diferenciação:** Subtrair o valor anterior do atual () para remover tendências.

**Transformação Logarítmica** **():** para estabilizar a variância.

**Diferenciação Sazonal:** Para séries com sazonalidade, subtrair valores do mesmo período anterior (​).

**Remover Tendência com Modelos de Regressão:** Ajustar e remover uma tendência linear ou polinomial.

**Yt​** representa o valor da série temporal no instante t. Ou seja, é o valor observado no tempo t.

**S** representa a sazonalidade da série, ou seja, o número de períodos após os quais os padrões se repetem.

**3. Decomposição**

A decomposição de séries temporais é uma técnica que separa a série em diferentes componentes para facilitar a análise e a modelagem. Isso permite entender melhor as tendências, identificar padrões sazonais e remover ruídos. Podendo ser dividida em Tendencia, Sazonalidade e Resíduo.

**1. Decomposição Aditiva**

Quando os componentes são somados:

Usada quando a **amplitude da sazonalidade é constante** ao longo do tempo.

**Exemplo:** Temperatura média diária ao longo do ano.

**2. Decomposição Multiplicativa**

Quando os componentes são multiplicados:

Aplicada quando a **sazonalidade varia em intensidade** conforme a tendência cresce ou diminui.

**Exemplo:** Receita de uma empresa, onde picos sazonais aumentam à medida que a empresa cresce.

**4. Modelagem**

Escolher e ajustar o modelo apropriado (ARIMA, SARIMA, LSTM, Prophet, etc.).

**5. Avaliação**

Ao trabalhar com séries temporais, a separação dos dados deve ser feita de forma **sequencial**, mantendo a ordem cronológica. Diferente de problemas comuns de aprendizado de máquina, onde a divisão pode ser aleatória, em séries temporais isso comprometeria a capacidade preditiva do modelo.

**Como Dividir os Dados?**

A abordagem mais utilizada é:

**Conjunto de Treino**: Parte inicial dos dados, usada para ajustar o modelo.

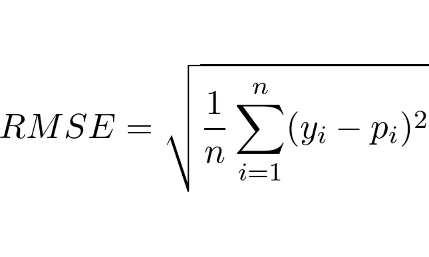
**Conjunto de Teste**: Últimos períodos da série, utilizados para avaliar a performance.

**Métricas de Avaliação para Séries Temporais**

Depois de treinar o modelo, ele precisa ser avaliado com métricas que quantificam o erro das previsões em relação aos valores reais. As mais utilizadas são:

**1. RMSE (Erro Quadrático Médio - Root Mean Squared Error)**

Mede a diferença média entre valores reais e previstos, penalizando mais os erros grandes:

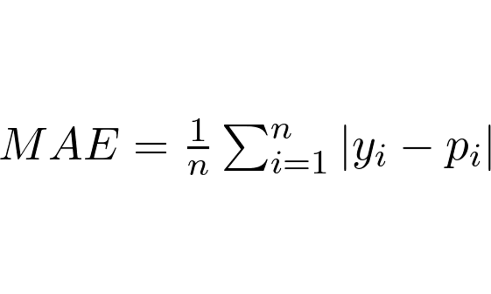


Valores menores indicam previsões mais precisas.

Penaliza fortemente erros grandes devido à elevação ao quadrado.

**2. MAE (Erro Absoluto Médio - Mean Absolute Error)**

Mede a média dos erros absolutos, sem dar peso maior para erros grandes:

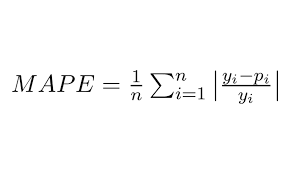


Fácil de interpretar, pois está na mesma unidade dos dados.

Não amplifica tanto os erros maiores, diferente do RMSE.

**3. MAPE (Erro Percentual Médio Absoluto - Mean Absolute Percentage Error)**

Expressa o erro médio em termos percentuais, o que facilita a interpretação:

​​​ 

Útil para comparar previsões em diferentes escalas de valores.

Problemático quando Yi​ se aproxima de zero, pois pode gerar distorções.

**Desafios em Séries Temporais**

1. **Não-estacionaridade**

Séries temporais que possuem média ou variância variando ao longo do tempo.

Solução: Transformações (diferenciação, logaritmos).

1. **Dados Perdidos (Missing Data)**

Dados ausentes podem prejudicar a análise.

Solução: Interpolação ou preenchimento com valores estimados.

1. **Séries Curta**

Poucos dados dificultam a identificação de padrões.

Solução: Adicionar variáveis externas ou usar modelos que lidem bem com pequenos conjuntos de dados.

1. **Sazonalidades Complexas**

Séries com múltiplos padrões sazonais podem ser difíceis de modelar.

**Regressão Logística**

A regressão logística é um modelo estatístico usado para classificação, especialmente em problemas binários (onde o resultado pode ser 0 ou 1, como "sim" ou "não", "spam" ou "não spam"). Diferente da regressão linear, que prevê valores contínuos, a regressão logística prevê probabilidades e as transforma em classes usando a função sigmoide.

**Fórmula Matemática**

A regressão logística modela a probabilidade como uma função logística (sigmoide) baseada em uma combinação linear das variáveis independentes.

Onde:

* p: Probabilidade estimada de uma observação pertencer à classe positiva.
* β0​: Intercepto.
* β1, β2, ..., βn ​: Coeficientes das variáveis independentes.
* x1, x2, ..., ​: Variáveis independentes.

Essa equação utiliza a função sigmoide, que transforma qualquer valor real em um número entre 0 e 1, permitindo interpretar a saída como uma probabilidade.

A decisão final é feita com base em um limiar (threshold), geralmente 0.5

Classe= {1 se p≥0.5 0 se p<0.5​}

**Assunções do Modelo**

* **Dependência linear**: A relação entre variáveis independentes e o logit (log das odds) é linear.
* **Independência das observações**: As amostras devem ser independentes umas das outras.
* **Ausência de multicolinearidade**: As variáveis independentes não devem ser altamente correlacionadas.
* **Homogeneidade da variância (não essencial)**.

**Métricas de Avaliação**

* **Acurácia**: Porcentagem de classificações corretas.
* **Precisão, Recall e F1-score**: Medem a performance em cada classe.
* **ROC e AUC**: Avaliam o desempenho do modelo em diferentes limiares.
* **Matriz de Confusão**: Analisa verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

**Limitações**

* Assume relação linear entre as variáveis independentes e o logit.
* Não é eficiente com dados altamente correlacionados (multicolinearidade).
* Pode subestimar ou superestimar quando o conjunto de dados não é balanceado.
* Pode ser inadequada para problemas não lineares.

**Aplicações Práticas**

* **Medicina**: Diagnóstico de doenças (ex.: probabilidade de ter câncer).
* **Marketing**: Previsão de churn (clientes que deixam o serviço).
* **Financeiro**: Probabilidade de inadimplência em empréstimos.
* **Recursos Humanos**: Probabilidade de um candidato ser aprovado.

**Exemplo**

**Problema:**  
Uma empresa deseja prever se um cliente comprará um seguro de vida baseado em dois fatores: **idade** e **renda anual**. A variável dependente é categórica (comprou ou não o seguro: 1 ou 0).

**1. Entendendo o Contexto**

**Variáveis:**

* **Independentes (Preditoras):**
  + **Idade:** A idade do cliente em anos.
  + **Renda anual:** A renda do cliente em milhares de dólares.
* **Dependente (Resposta):**
  + **Compra:** Se o cliente comprou o seguro (1) ou não (0).

**Objetivo:**

* Estimar a **probabilidade de compra** com base na idade e na renda anual.
* Classificar cada cliente como **comprador (1)** ou **não comprador (0)**.

**2. Como o Modelo Funciona**

1. **Modelo Conceitual:**
   * A regressão logística transforma as variáveis preditoras (x1=idade, x2=renda) em uma **combinação linear**.
   * A combinação linear (z) é mapeada para uma **probabilidade (p)** usando a função sigmoide (f(z)):
2. **Tomada de Decisão:**
   * Se p≥0, o cliente será classificado como **comprador (1)**.
   * Se p<0.5, o cliente será classificado como **não comprador (0)**.

**3. Suposições do Modelo**

* A probabilidade de compra varia **logisticamente** com a idade e a renda.
* Os efeitos de idade e renda são **aditivos** e contribuem linearmente para z.
* As observações (clientes) são **independentes**.

**4. Exemplos de Situações**

* **Cliente A:** 25 anos, renda anual de $20.000.
  + O modelo pode prever uma probabilidade de 0.20 (ou 20%), classificando-o como **não comprador (0)**.
* **Cliente B:** 45 anos, renda anual de $50.000.
  + O modelo pode prever uma probabilidade de 0.85 (ou 85%), classificando-o como **comprador (1)**.

**5. Como Interpretar os Coeficientes**

Se o modelo estimar os seguintes coeficientes:

* β0=−2.5: A probabilidade inicial sem levar em conta idade e renda.
* β1=0.03: Cada ano de idade aumenta a probabilidade de compra em e0.03=1.03, ou seja, **3% a mais na chance de compra**.
* β2=0.1: Cada $1.000 extras na renda anual aumenta a probabilidade em e0.1=1.105, ou seja, **10.5% a mais na chance de compra**.

**6. Limitações do Modelo**

* **Não-linearidade real:** Se a relação entre idade/renda e probabilidade não for linear, o modelo pode ser inadequado.
* **Classes desbalanceadas:** Se poucos clientes compram o seguro, o modelo pode ter dificuldade em prever corretamente os compradores.

**7. Benefícios Conceituais**

* A regressão logística não apenas classifica, mas também fornece uma **probabilidade estimada**.
* Permite entender **quais fatores influenciam mais** na compra.

Simplicidade e facilidade de explicação para tomadores de decisão.

**Machine Learning**

Estamos falando da capacidade de uma máquina aprender (Adquirir conhecimento)

A máquina aprende através de métodos analíticos, algoritmos, de dados históricos coletados de acordo com a necessidade da questão, assim gerando um modelo que poderá ser usado para determinar decisões futuras ou semelhante

E assim como aprendemos, conseguimos medir esse conhecimento, e é importante você ter essa informação do modelo gerado.

OBS: É importante saber da limitação referente a massa dos dados, de acordo com a problemática existe inúmeras variáveis que podem ou não podem ser medidas através desses bancos de dados, variáveis essas que tem importante determinação para o resultado do modelo, como por exemplo se determinado dia vai chover ou não. Sendo assim impossível ter uma assertividade de 100% estatisticamente falando

**Aplicações**

O objetivo e com isso a aplicação do Machine Learning é justamente obter alguma resposta através desse aprendizado, muito usado em áreas de marketing para determinar por exemplo qual produto qual cliente vai receber a promoção, ou para recursos humanos ao determinar um perfil ideal para determinada vaga, assim por diante

**Conceitos Importantes**

**Estrutura de Dados**

Atributos/dimensões: são as características de determinado problema

Classe: São os objetivos do modelo, a reposta que ele pretende obter

Instâncias: São as “linhas” de acordo com cada problemática, por exemplo um cliente, um dia de dado.

Relação: É o conjunto de dados

Dados Numéricos ou Nominais

**Tarefas**

Classificação: Quando queremos prever algo categórico (onde a classe é nominal, por exemplo descobrir a espécie de um animal ou se uma pessoa tende a pagar ou não um empréstimo)

Regressão: Quando queremos prever um número (onde a classe é numérica, por exemplo descobrir o peso de uma pessoa de acordo com sua idade e altura)

Agrupamentos: Quando queremos unir elementos com características semelhantes (Não queremos definir nem prever nada, ou seja, sem classe, por exemplo definir um grupo de clientes para determinada campanha de marketing)

Regras de associação: Quando queremos buscar uma semelhança entre diferentes elementos (Por exemplo, em uma cesta de compras, um cliente compra produto A e produto B podemos associar um produto ao outro)

As tarefas supervisionadas são aquelas que existem uma classe ou um atributo especial que nos permite comparar e validar o resultado, o modelo aprende a partir de dados rotulados, ou seja, exemplos que já possuem a resposta correta. Já as não supervisionadas não existe essa comparação/validação, o modelo aprende sem rótulos, ou seja, ele analisa padrões e estruturas nos dados sem saber previamente as respostas corretas.

OBS: Um modelo é aquilo que o machine learning produz ao estudar os dados utilizando de algum algoritmo que serve para tomar a próxima decisão ou gerar a previsão.

Tabela

Descrição gerada automaticamente

**Treinando a máquina**

Nós treinamos a máquina, como já mencionado, a partir de dados coletados, dados históricos, utilizamos uma base percentual para fazer o treinamento, por exemplo 70% desses dados, treinamos o modelo utilizando os algoritmos e então assim precisamos avaliar o resultado de previsão ao fazer testes com os outros 30% dos dados, podendo gerar a avaliação do desempenho do modelo. (Hold Out)

Também existe o método de Validação Cruzada, que parte pelo principio de repartir os dados em N partições onde o algoritmo irá treinar com cada um deles e criar um modelo fazendo testes alternadamente entre essas partições e fazendo a avaliação de cada um para definição do modelo que entrará em produção

Quando estamos criando um modelo queremos chegar num resultado “genérico” para que possa se ajustar diante o treinamento e também dentro da produção. Existindo modelos super ajustados (que são ótimos quando estamos falando de dados de treino mas em produção tem baixo desempenho) e sub ajustados (quando não conseguem boas taxas de previsão)

**Causas de Super/Sub ajuste**

* - Dados não representativos
* - Dados não significativos (poucos)
* - Forma do treinamento
* - Classe rara
* - Modelo incorreto

OBS: É muito interessante usar da tabela de confusão para determinar a eficácia do modelo (de acordo com o que você perde nos falsos negativos ou falsos positivos)

Tabela

Descrição gerada automaticamente

**Como melhorar um modelo?**

* - Testando diferentes algoritmos
* - Parametrizando algoritmos (hiper parâmetros)
* - Selecionando e tratando dados
* - Seleção/engenheira de atributos

Tabela

Descrição gerada automaticamente

**Algoritmos**

* Árvore de Decisão
* Regras
* Naive Bayes
* Redes Bayesianas
* Máquina de vetor de suporte
* Métodos de Grupos
* Aprendizado Baseado em Instância

**Codificação de Categoria**

Processo de transformar categorias em números (Label encoding ou One-hot encoding)

**Label encoding**

Definimos em ordem alfabética as categorias e denominamos os números à elas começando de 0

**One-hot Encoding**

Cada categoria é transformada em atributo (dummy variable), ou seja, uma nova coluna, tendo valores 0 ou 1 por exemplo. Existe alguns problemas como multicolinearidade, aonde para resolver excluímos alguma coluna

**Quando usar qual?**

Tabela

Descrição gerada automaticamente

**Dimensionamento de Características**

Normalmente as variáveis nos dados tem escalas diferentes, oque contribui para desbalancear o modelo, então para isso precisamos dimensionar esses dados de forma a construir uma melhor conversação entre esses dados

**Padronização (z-score)**

Transformamos os dados para algo próximo de média 0 e desvio padrão 1, ele não afeta outliers e deve ser usado na maioria dos casos

Uma imagem contendo Padrão do plano de fundo

Descrição gerada automaticamente

**Normalização (min-max)**

Transforma para escala comum entre 0 e 1, bastante usado para processamento de imagens e RNA, usamos quando não sabemos a distribuição dos dados, e eles precisam ser positivos, removendo outliers visto que impõe um limite.

OBS: Dimensionamento não necessariamente vai melhorar o seu modelo, e existe algoritmos cujo não necessita dessa tarefa

**Agrupamentos**

No agrupamento não existe classes, o objetivo é reunir grupos de acordo com características comuns.

Modelo Difuso: Cada elemento pertence a um grupo segundo uma probabilidade

Modelo Hierárquico: Permite que o grupo tenha subgrupos

* - k-Means
* - K-Medoid
* - DBScan

**Regras de Associações**

Suporte: número de transações que contêm todos os itens da transação

Confiança: indica a proporção de vezes que, em uma transação contendo o elemento A, também tem B

Lift: o quanto aumenta a frequência de B com a ocorrência de A

**Apriori e FP-Grow**

São algoritmos mais comuns na mineração de regras de associações

Apriori é baseado no principio de que se um conjunto de itens é frequente, um subconjunto destes itens também será.

FP-Grow induz árvores, e busca sobreposição destas árvores, onde os itens são frequentes

**Análise Fatorial**

É um nome genérico dado a uma classe de métodos estatísticos multivariados cujo propósito principal é definir a estrutura subjacente em uma matriz de dados. Assim, têm-se os seguintes pontos:

* Abordar o problema de analisar a estrutura das inter-relações (correlações) entre um grande número de variáveis, definindo um conjunto de dimensões latentes comuns, chamadas de fatores;
* Identificar as dimensões separadas da estrutura e então determinar o grau em que cada variável é explicada por cada dimensão; resumir e reduzir os dados em dimensões latentes interpretáveis e compreensíveis usando escores para cada dimensão e, consequentemente, substituir as variáveis originais.
* Os modelos de análise fatorial buscam explicar o comportamento das variáveis observadas em relação ao comportamento de um conjunto de variáveis não observadas (variáveis latentes (PCA) ou fatores (AFE)).

**Normalidade dos Dados**

Existem muitos testes que podem ser utilizados para avaliar o quanto a distribuição de uma determinada amostra se parece com a distribuição Gaussiana (normal). Cada um dos testes aborda diferentes pressupostos e aspectos dos dados.

Cada teste retorna ao menos dois parâmetros:

* **Estatística:** Um valor calculado que pode ser interpretado comparando-o com algum valor crítico de uma distribuição estatística de deste.
* **p-valor:** Usado para interpretar o teste. Neste caso, o quanto a distribuição se aproxima de uma distribuição normal.

Para analisar a estatística calculada, requer maior profundidade e proficiência em estatística e no teste estatístico específico. Em contrapartida, o p-valor é mais simples e prático para analisar.

O teste assume que a amostra provém de uma distribuição normal. Tecnicamente, chamada de hipótese nula ou H0. O nível de confiança, chamado alfa, tipicamente 5% (ou 0,05), é usado para interpretar o p-valor.

* Se **p > 0,05 (alfa)**, os dados **são normais** (ou seja, o teste não encontrou evidências contra a normalidade).
* Se **p < 0,05 (alfa)**, os dados **não são normais** (há evidências contra a normalidade).

Em geral, estamos procurando por p-valores maiores que alfa, confirmando que a distribuição da amostra provém de uma distribuição normal.

**Normalidade: Análise de Assimetria e Curtose – Valores Padronizados**

A assimetria (skewness) e a curtose (kurtosis) são medidas estatísticas que ajudam a entender a distribuição dos dados, avaliando se eles seguem uma distribuição normal ou apresentam desvios significativos

**Assimetria:** mede o grau de desvio da distribuição em relação à simetria perfeita

**Curtose:** mede o grau de achatamento ou concentração dos dados em relação a uma distribuição normal.

A partir da assimetria e curtose, calcula-se a assimetria e curtose padronizados e compare-se este valor com os valores críticos de z.

**Z-assimetria:** Mede o quão estatisticamente significativa é a assimetria, levando em conta o erro padrão.

**Z-curtose:** Mede o quão estatisticamente significativa é a curtose, levando em conta o erro padrão.

Ambas são para determinar se a assimetria/curtose são estatisticamente significativa

Uma imagem contendo Tabela

O conteúdo gerado por IA pode estar incorreto.

Valores críticos de 𝑧:

* 𝑧= ±2,58 (𝑠𝑖𝑔𝑛𝑖𝑓𝑖𝑐â𝑛𝑐𝑖𝑎 𝑑𝑒 0,01 – referente ao intervalo de confiança 99%)
* 𝑧= ±1,96 (𝑠𝑖𝑔𝑛𝑖𝑓𝑖𝑐â𝑛𝑐𝑖𝑎 𝑑𝑒 0,05 – referente ao intervalo de confiança 95%)

Se o valor 𝑧 calculado exceder o valor crítico especificado, então a distribuição é não-normal

**Normalidade: Teste K^2 de D’Agostino**

É um teste de normalidade que verifica se um conjunto de dados segue uma distribuição normal, com base em assimetria (skewness) e curtose (kurtosis). Melhor com amostras médias e grandes

* Hipótese nula (H0): Os dados seguem uma distribuição normal.
* Hipótese alternativa (H1​): Os dados não seguem uma distribuição normal.

A estatística final do teste é dada por:

**Normalidade: Teste KS - Kolmogorov-Smirnov**

Este teste avalia se uma dada distribuição para uma variável randômica se aproxima da distribuição da variável analisada.

* Hipótese nula (H0): Os dados seguem uma distribuição normal.
* Hipótese alternativa (H1​): Os dados não seguem uma distribuição normal.

O teste calcula a maior diferença absoluta entre a distribuição acumulada dos dados e a distribuição acumulada teórica da normal. E funciona melhor com amostras grandes

**Normalidade: Teste W - Shapiro-Wilk**

* Hipótese nula (H0): Os dados seguem uma distribuição normal.
* Hipótese alternativa (H1​): Os dados não seguem uma distribuição normal.

Funciona bem para amostras pequenas (n < 50), para maiores, pode haver pequenas diferenças e rejeitar a normalidade mesmo quando são.

--------------------------------------------------------------------------------------------------------

**Matriz de correlações**

* Examinar a matriz de correlações e verificar se existem valores significativos para justificar a utilização da técnica. Caso as correlações entre TODAS as variáveis sejam baixas, talvez a análise fatorial não seja apropriada
* Espera-se que variáveis que apresentam alta correlação tendem a compartilhar o mesmo fator
* A matriz de correlação mede a associação linear entre as variáveis por meio do coeficiente de correlação de Pearson
* Se não houver um número substancial de valores de correlação superiores a 0,30, há fortes indícios de que a utilização da técnica não é apropriada. (HAIR, et all; 2009). Contudo, cabe destacar que depende do tamanho da amostra.

**Correlação entre os Dados**

As correlações lineares de Pearson têm variação de -1 a 1, e quanto mais próximo de 0, a correlação é mais fraca, mais longe, mais forte. E quanto mais forte a relação, maior associação entre as variáveis tem entre si, e se positivo, a associação é proporcional, se negativa, é contrária.

**Teste de esfericidade de Bartlett**

(R = matriz correlação

**H0:** R = Matriz identidade) (as variáveis não estão correlacionadas)

**HA:** R != Matriz identidade (Existe pelo menos alguma **correlação significativa** entre as variáveis)

Se 𝐻0 não é rejeitada, as inter-relações entre as variáveis são iguais a 0. As variáveis não estão correlacionadas. A utilização da AF não é adequada

Se o p-valor do teste de Bartlett for menor que 0,05 (p < 0,05), rejeitamos a hipótese nula e concluímos que os dados são adequados para a Análise Fatorial.

**Teste de KMO (Kaiser-Meyer-Olkin)**

Mede o grau de adequação da amostra para uma Análise Fatorial. Ele verifica se as variáveis possuem correlação suficiente para formar fatores significativos.

Medição da associação de todas as variáveis em conjunto, o teste KMO calcula a proporção da variância dos dados que pode ser explicada por fatores latentes, comparada à variância que é devida ao erro amostral.

Variando entre 0 e 1, quanto mais próximo de 1, melhor adequação para o uso da Análise Fatorial, mais próximo de 0, sem adequação.

Se o KMO for menor que 0,50, significa que as variáveis não possuem correlação suficiente para uma análise fatorial confiável.

**Teste MSA**

Faz parte do Teste KMO (Kaiser-Meyer-Olkin) e serve para verificar se os dados são adequados para a Análise Fatorial. Ele analisa o quanto cada variável tem correlação com as outras e indica se a matriz de correlação é apropriada para extração de fatores.

Ou seja, ele é uma espécie de Teste KMO individual, e sua matriz completa, MSA geral, é o valor do teste KMO

Variando entre 0 e 1, quanto mais próximo de 1, melhor adequação dos dados para a Análise Fatorial, e quanto mais próximo de 0, pior. Abaixo de 0.5, é uma variável ruim.

**Extração de Fatores:**

**Análise de Componentes Principais (PCA - Principal Component Analysis)**

Considera a variância total dos dados. O PCA procura uma combinação linear das variáveis observadas, de maneira a maximizar a variância total explicada. Se determinadas variáveis forem altamente correlacionadas, elas serão combinadas de modo a formar um fator que explicará a maior quantidade de variância na amostra e assim, sucessivamente, para os outros fatores.

É um método estatístico usado para reduzir a dimensionalidade de um conjunto de dados enquanto mantém a maior quantidade possível de informação.

Basicamente, a PCA transforma um conjunto de variáveis originais (correlacionadas) em um novo conjunto de variáveis não correlacionadas chamadas componentes principais.

**Análise dos Fatores Comuns (AFC)**

o objetivo primário é identificar fatores ou dimensões latentes que reflitam o que as variáveis têm em comum representadas nas variáveis originais

**Componente (PCA)**

Um componente é uma variável latente (não observada) que representa uma combinação linear das variáveis originais. Ele busca resumir a variabilidade dos dados de forma mais compacta, reduzindo a dimensionalidade enquanto mantém o máximo de informação possível. Representa uma nova variável construída a partir de combinações lineares das variáveis originais, buscando maximizar a variância explicada.

**Fator (AFC):** Representa uma variável latente hipotética que explica a correlação entre as variáveis originais.

**Fatores (Dimensões latentes comuns)**

Os fatores são variáveis latentes que representam padrões ocultos nos dados. Eles ajudam a explicar a variabilidade das variáveis originais agrupando aquelas que têm correlações altas entre si. São encontradas pelas correlações entre as variáveis, ajudando a reduzir a dimensionalidade dos dados

Cada fator representa um **tema geral** que influencia as variáveis observadas.

**Critério da raiz Latente (Autovalores)**

O autovalor (também chamado de valor próprio ou eigenvalue) é um número que representa a quantidade de variância explicada por um fator em uma análise fatorial.

Ele indica o quanto um fator contribui para a estrutura dos dados, ou seja, sua importância na explicação da variabilidade total das variáveis originais.

Apenas fatores que têm autovalores (eigenvalues) maiores que 1 são considerados significantes, os demais são descartados; Isto porque, no mínimo, o componente deve explicar a variância de uma variável utilizada no modelo (média 0 e desvio 1).

**Cargas Fatoriais**

Cada variável original pode estar associada a um ou mais fatores. A carga fatorial de uma variável em um fator específico indica a importância desse fator na explicação da variabilidade dessa variável. Corresponde à correlação entre uma variável e um fator latente. Mede o quanto uma variável contribui para um fator específico.

**Nomeação dos Fatores**: As cargas fatoriais ajudam a interpretar e nomear os fatores. Para isso, você olha para as variáveis que têm as maiores cargas em cada fator e tenta identificar um padrão ou tema comum.

**Exemplo:** Se as variáveis que têm altas cargas no Fator 1 estão relacionadas à satisfação do cliente, você pode chamar esse fator de "Satisfação do Cliente".

* Em geral, considera-se cargas fatoriais maiores que 0,30.
* Cargas superiores a 0,40 são consideradas importantes
* Maiores que 0,50 são consideradas estatisticamente significativas
* Cargas fatoriais significantes dependem do tamanho da amostra
* O valor da carga fatorial representa a quantia de variância total da variável explicada pelo fator

As cargas fatoriais na **matriz rotacionada** representam a correlação entre cada variável original e os fatores extraídos em uma análise fatorial. Elas indicam o quanto cada fator contribui para a variação de uma variável e ajudam a interpretar a estrutura dos dados.

O valor da carga fatorial ao quadrado representa a quantia de variância total da variável explicada pelo fator.

**Composição dos fatores**

As variáveis são agrupadas em fatores a partir da carga fatorial. Ao analisar a linha, a variável se agrupará ao fator cuja carga fatorial seja a maior, em valor absoluto (módulo da carga fatorial).

**Matriz de Componente**

A matriz de componentes (ou matriz de cargas fatoriais) é uma tabela que mostra a correlação entre as variáveis originais e os fatores extraídos em uma Análise de Componentes Principais (PCA) ou em uma Análise Fatorial.

Ela indica o quanto cada variável contribui para cada fator, ajudando a interpretar a estrutura dos dados.

**Comunalidade**

A comunalidade de um dado refere-se à proporção da variância de uma variável que pode ser explicada pelos fatores comuns em uma análise fatorial. Em outras palavras, é o quanto de informação sobre uma variável é mantida quando se reduzem as dimensões dos dados para um conjunto menor de fatores.

**Comunalidade alta (próxima de 1):** A variável tem grande parte de sua variância explicada pelos fatores comuns.

**Comunalidade baixa (próxima de 0):** A variável não é bem explicada pelos fatores comuns, sugerindo que possui variância única ou erro.

**Especificidade**

A especificidade de um dado refere-se à parte da variância de uma variável que não é explicada pelos fatores comuns em uma análise fatorial. Em outras palavras, é a variação única da variável, composta por fatores específicos ou erro de medição. Sendo calculada por:

Esp = 1 – Comunalidade

**Quando menor que 0.30:** A variável é bem representada pelos fatores (boa adequação).

**Quando maior que 0.50:** A variável tem muita variância não explicada (pode não ser útil na análise fatorial).

ESCREVER SOBRE TRANSFORMAÇÃO DE DADOS

**Diferença entre PCA (Análise de Componentes Principais) e Análise Fatorial**

A PCA (Principal Component Analysis) e a Análise Fatorial (AF) são técnicas estatísticas usadas para reduzir a dimensionalidade dos dados e identificar padrões. Porém, elas têm propósitos e abordagens diferentes.

|  |  |  |
| --- | --- | --- |
| Característica | PCA (Análise de Componentes Principais) | Análise Fatorial |
| Objetivo | Reduzir a dimensionalidade dos dados enquanto mantém a máxima variância. | Identificar variáveis latentes (fatores) que explicam relações entre variáveis. |
| Base Matemática | Baseia-se nos autovalores e autovetores da matriz de covariância/correlação. | Baseia-se no modelo de variáveis latentes, que assume que as variáveis são influenciadas por fatores ocultos. |
| O que extrai? | Componentes principais, que são combinações lineares das variáveis originais. | Fatores, que explicam correlações entre variáveis observadas. |
| Os componentes/fatores são reais? | Os componentes são apenas transformações matemáticas e não representam uma estrutura real subjacente. | Os fatores são considerados variáveis latentes que existem na realidade. |
| Uso da rotação? | Normalmente **não** usa rotação. | A rotação (Varimax, Oblimin) é comum para facilitar a interpretação. |
| Ordem de extração | O primeiro componente explica mais variância, o segundo explica menos, etc. | Os fatores podem ser redistribuídos após rotação. |
| Correlação entre componentes/fatores? | Componentes são ortogonais (não correlacionados). | Pode ter fatores correlacionados (rotação oblíqua) ou não (rotação ortogonal). |
| Aplicações | Análise exploratória, redução de dimensionalidade, machine learning. | Psicologia, ciências sociais, questionários e testes psicológicos. |

**Exemplo Prático**

Imagine que temos um conjunto de variáveis psicológicas:

* Ansiedade, Estresse, Depressão, Extroversão, Sociabilidade.

**Se usarmos PCA:**

A PCA cria **componentes principais** que combinam essas variáveis da melhor forma possível para capturar a variação nos dados.

Exemplo de resultado:

* **Componente 1** = 0.7 × Ansiedade + 0.6 × Estresse + 0.5 × Depressão
* **Componente 2** = 0.6 × Extroversão + 0.7 × Sociabilidade

**Se usarmos Análise Fatorial:**

A análise fatorial tenta descobrir fatores latentes que explicam por que essas variáveis estão correlacionadas.

Exemplo de resultado:

* **Fator 1 (Saúde Mental)**: Ansiedade (0.8), Estresse (0.75), Depressão (0.7)
* **Fator 2 (Comportamento Social)**: Extroversão (0.85), Sociabilidade (0.90)

**Rotação dos Fatores**

A rotação tem como objetivo principal, a transformação dos coeficientes dos componentes principais retidos em uma estrutura simplificada;

A rotação dos fatores é possível, pois as cargas fatoriais podem ser representadas como pontos entre eixos (fatores).

Os eixos podem ser girados sem alterar a distância entre os pontos, contudo, as coordenadas dos pontos (cargas fatoriais) em relação aos eixos são alteradas

Com a rotação:

* As cargas fatoriais tendem a ficar mais extremas (próximas de -1, 0 ou +1), facilitando a identificação das variáveis mais fortemente associadas a cada fator.
* Cada fator fica mais bem definido, tornando a interpretação mais clara.

Os métodos para rotacionar podem ser:

* **Ortogonais:** não correlacionados entre si e interpretados a partir das cargas – loadings
  + **Varimax:** busca minimizar o número de variáveis que têm altas cargas em um fator, simplificando a interpretação dos fatores. Para cada componente, existem alguns pesos significativos e todos os outros, próximos de zero.
  + **Quartimax:** busca simplificar as linhas de uma matriz fatorial, ou seja, tornar os pesos de cada variável elevados para um pequeno número de componentes e próximos de zero para todos os demais.. Busca minimizar o número de fatores necessários para explicar uma variável
  + **Equimax:** Combinação dos dois anteriores
* **Oblíquos:** fatores correlacionados e, para interpretá-los, considera-se simultaneamente as correlações e os loadings

Gráfico

O conteúdo gerado por IA pode estar incorreto.

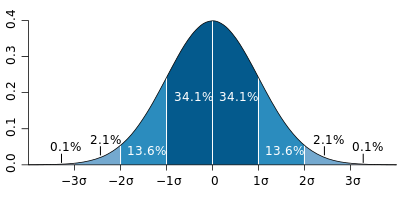
**Matriz de Componente Rotativa**

A matriz de componentes rotativa (ou matriz de cargas fatoriais rotacionada) é a matriz obtida após aplicar rotação na análise fatorial ou na Análise de Componentes Principais (PCA).

Ela ajusta a distribuição das cargas fatoriais para tornar os fatores mais interpretáveis.

**Testes de Hipoteses**

Para determinar se um dado é importante estatisticamente falando, precisamos não só olhar para o r (correlação) mas também para o p-value (significância: é um conceito que mede a probabilidade de que a diferença entre dois dados não seja causada por acaso) entre as variáveis.



Quando falamos de estatística e análise de dados, precisamos sempre ter em mente a importância de fazer os testes de hipótese para determinadas perguntas estatísticas, e para isso ter noção do gráfico de distribuição normal, com intervalo de confiança (indica a probabilidade de que o verdadeiro valor de uma população esteja dentro de um determinado intervalo) sendo 95% ou 99%, onde esses dois valores têm relação com o desvio padrão (medida estatística que indica o grau de dispersão de um conjunto de dados) dos dados

**Análise de Agrupamentos (Clustering)**

**ESCREVER SOBRE CENTROIDE E TABELA ANOVA**

**O que é**

A análise de agrupamentos (ou clustering) é uma técnica de aprendizado não supervisionado usada para identificar padrões e organizar dados em grupos (clusters) com base em suas semelhanças. O objetivo é encontrar estruturas naturais nos dados sem que haja rótulos pré-definidos.

**Objetivo**

O principal objetivo é desenvolver uma taxonomia que particione objetos em grupos com percepções similares. Ou seja, é identificar padrões e estruturas nos dados, agrupando elementos similares em conjuntos (conglomerados) distintos. Essa técnica é utilizada para entender melhor os dados, reduzir complexidade e tomar decisões estratégicas com base nos grupos identificados.

**OBS:** Os agrupamentos resultantes de objetos devem exibir elevada homogeneidade interna e elevada heterogeneidade externa

**Exemplo de Utilização:**

* Identificar grupos de investimentos de acordo com perfis de risco
* Identificar segmentos homogêneos de consumidores e estabelecer programas de marketing específicos para cada segmento
* Identificar grupos de alunos mais propensos à evasão escolar
* Buscar grupos de segurados de menor risco
* Segmentar empresas com base em indicadores financeiros (rentabilidade, liquidez, margem etc.)

**Apoio Conceitual**

A análise de agrupamento é descritiva, não-teórica e não inferencial

A análise de agrupamentos sempre criará agrupamentos, independentemente da existência real de alguma estrutura nos dados

A solução de agrupamentos não é generalizável, pois é totalmente dependente das variáveis usadas como base para medida de similaridade.

**Análise das Variáveis**

A seleção das variáveis (ou atributos) é um passo crítico, pois impacta diretamente a formação dos clusters. O processo envolve:

**Critérios de Seleção de variáveis e identificação de outliers**

Variáveis não representativas ou a presença de multicolinearidade podem distorcer os resultados do estudo

**Relevância:** As variáveis selecionadas devem ser relevantes para o objetivo do clustering. Por exemplo, em uma segmentação de clientes, características como idade, renda e preferências de compra são mais relevantes que atributos como nome ou número de telefone.

**Escala**: As variáveis devem estar em uma escala semelhante, pois muitos algoritmos de clustering (como K-Means) são s ensíveis à distância entre os pontos. Se você tiver variáveis com escalas muito diferentes (por exemplo, uma variável de renda que vai de 1000 a 100.000 e uma de idade que vai de 18 a 80), será necessário normalizar ou padronizar essas variáveis.

* **Normalização**
  + A normalização é uma técnica que ajusta os dados para um intervalo definido, geralmente entre 0 e 1. Essa técnica é útil quando você quer garantir que todas as variáveis fiquem na mesma escala e dentro de um intervalo específico.
  + Usamos quando você precisa garantir que os valores estejam em um intervalo específico, como em redes neurais que podem ter funções de ativação que limitam os valores de entrada ou quando a distribuição dos dados é não-gaussiana e você quer evitar que valores muito grandes ou pequenos dominem o modelo.
  + Seu principal método é **mini-max** que escala os valores entre um intervalo específico, geralmente [0,1], e é bom para algoritmos que requerem intervalos fixos como por exemplo redes neurais.
* **Padronização**
  + A padronização transforma os dados de forma que eles tenham média 0 e desvio padrão 1, ou seja, ela ajusta os dados para que a distribuição tenha a forma de uma distribuição normal padrão.
  + Devemos usar quando a distribuição dos dados é aproximadamente normal. Quando o modelo de aprendizado de máquina exige que os dados tenham média zero e desvio padrão unitário (como em PCA ou regressão linear). Em modelos sensíveis à escala, como Support Vector Machines (SVM) e K-Means.

**Métodos de Padronização**

* **Z-Score** (Transforma para media 0 e desvio padrão 1)
  + Útil quando os dados seguem uma distribuição normal
* Transformação Logarítmica
  + Reduz o impacto de valores extremos e torna a distribuição mais simétrica:
  + Bom para dados com crescimento exponencial.
* Método da máxima amplitude (valor máximo igual a 1)
* Método de média 1 (A variável padronizada tem média 1)
* Método de desvio padrão 1 (A variável padronizada tem desvio padrão 1)

**Mais alguns métodos e comparações:**

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

A escolha do método de padronização depende das características dos seus dados e dos algoritmos de aprendizado de máquina que você está utilizando. Normalização e padronização são as técnicas mais comuns, mas, em cenários com outliers ou dados assimétricos, o escalonamento robusto ou a transformação logarítmica podem ser mais eficazes.

**Correlações entre variáveis**: Se duas ou mais variáveis forem altamente correlacionadas (ex: renda e patrimônio), pode ser interessante remover uma delas para evitar redundância e melhorar a eficiência do algoritmo

**Métodos para Seleção de Variáveis**

* **Métodos Automáticos**: Algumas técnicas de aprendizado, como a Análise de Componentes Principais (PCA), podem ser usadas para reduzir a dimensionalidade dos dados, selecionando as variáveis mais informativas para o agrupamento.
* **Análise de Correlação**: Pode-se também usar a matriz de correlação para eliminar variáveis redundantes.

**OBS:** Outliers são pontos de dados que se distanciam significativamente do resto dos dados e podem interferir no processo de clustering. Eles podem ser identificados de diferentes maneiras, dependendo do contexto.

* **Remoção**: Se o outlier for um erro de entrada de dados ou uma anomalia sem relevância, pode-se removê-lo do conjunto de dados.
* **Transformação**: Às vezes, outliers podem ser suavizados ou transformados usando técnicas como logaritmos ou escalonamento.
* **Tratamento com Modelos**: Alguns algoritmos, como o DBSCAN, podem ser utilizados para detectar e ignorar outliers durante o processo de clustering.

**Medidas de Distância**

Para análise de clusters, as medidas de distância desempenham um papel crucial na definição da similaridade ou dissimilaridade entre os dados (a similaridade entre os pontos de dados é usada para definir quais pontos devem ser agrupados juntos), o que, por sua vez, afeta como os clusters são formados. As medidas de distância mais usadas para análise de clusters variam de acordo com a natureza dos dados e os algoritmos de clustering empregados.

**Distância Euclidiana**

A distância entre duas observações (𝑖 e 𝑗) corresponde à raiz quadrada da soma dos quadrados das diferenças entre os pares de observações (𝑖 e 𝑗) para todas as p variáveis. Em que 𝑥𝑖𝑘 é o valor da variável 𝑘 referente à observação 𝑖 e X𝑗𝑘 para a observação 𝑗. Nesta abordagem, quanto menor a distância, mais similares serão as observações. (Ideal para dados contínuos e em casos em que a distribuição dos dados é aproximadamente esférica)

Texto

O conteúdo gerado por IA pode estar incorreto.

**Distância Quadrática Euclidiana**

Recomendada para os métodos de agrupamento centroide e Ward

**Distância de Minkowski**

A distância de Minkowski é uma generalização da distância Euclidiana e de Manhattan. Ela é configurada com um parâmetro p que determina a medida de distância desejada. Usa quando você está trabalhando com dados multivariados e deseja uma flexibilidade maior ao escolher a métrica de distância, a distância de Minkowski é uma escolha poderosa.

Logotipo

O conteúdo gerado por IA pode estar incorreto.

**Distância de Mahalanobis**

A distância de Mahalanobis leva em consideração a covariância dos dados. Ela é particularmente útil quando os dados possuem correlação entre as variáveis e é muito usada quando os clusters têm diferentes variâncias. Funciona bem quando os dados têm diferentes escalas ou correlações entre variáveis, permitindo que os clusters sejam formados com base nas propriedades estatísticas dos dados.

Texto

O conteúdo gerado por IA pode estar incorreto.

Em que S é a estimativa amostral da matriz de variância covariância Σ dentro dos agrupamentos (matriz de covariância invertida dos dados.)

A distância de Mahalanobis ao quadrado (D²) é frequentemente usada em contextos estatísticos e de aprendizado de máquina quando se deseja evitar a raiz quadrada para facilitar cálculos ou quando se trabalha com distribuições estatísticas.

A distância de Mahalanobis já é baseada na matriz de covariância dos dados, e muitas análises estatísticas utilizam D² diretamente porque a raiz quadrada não adiciona significado extra ao cálculo.

Se os dados seguem uma distribuição normal multivariada, então D² segue aproximadamente uma distribuição qui-quadrado X² com p graus de liberdade (p = número de variáveis).

**Isso permite definir um limiar estatístico para identificar outliers:**

Usando a regra, consideramos o GL (número de variáveis) e multiplicamos por 2,5. E todos os números maiores que esse, possivelmente é um outlier



Use D² quando precisar de análises estatísticas rigorosas, detecção de outliers baseado em X² (distribuição qui-quadrado), ou métodos que exijam métricas quadráticas, como Ward. Para comparações diretas de distâncias, a versão normal pode ser mais interpretável

**Medidas Correlacionais**

As medidas correlacionais são usadas para medir a relação ou a associação entre duas ou mais variáveis. Em análise de dados, é fundamental entender o tipo de relação entre as variáveis, pois isso ajuda a determinar se existe uma dependência entre elas e a qual tipo de análise ou modelo se deve aplicar. A correlação pode ser positiva, negativa ou inexistente, e as medidas de correlação fornecem uma forma de quantificar essa relação.

Na análise de clusters, medir a correlação entre as variáveis pode ajudar a entender a relação entre os atributos dos dados. Se duas variáveis são altamente correlacionadas, isso pode influenciar a maneira como o algoritmo de clustering forma os grupos

Em K-Means, a alta correlação entre variáveis pode influenciar a forma do cluster. Se as variáveis estão altamente correlacionadas, elas podem ser tratadas como uma única dimensão na análise, impactando a distribuição dos pontos e a formação do centroide.

**Correlação de Pearson**

Pode ser útil para verificar se duas variáveis estão fortemente relacionadas. Se duas variáveis têm uma forte correlação positiva ou negativa, o algoritmo de clustering pode considerar essas variáveis mais similares entre si.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

**Medidas de Associação**

As medidas de associação são estatísticas utilizadas para avaliar a relação entre variáveis categóricas ou contínuas. Na análise de clusters, elas são úteis para verificar a força das conexões entre as variáveis e para entender melhor como os dados estão agrupados. Essas medidas podem ser usadas para selecionar variáveis relevantes, interpretar a composição dos clusters e validar a segmentação obtida.

São utilizadas para representar a similaridade quando se trata de variáveis nominais, baseando-se em tabelas de contingência.

A presença ou ausência de determinada característica pode ser descrita matematicamente pela introdução de variáveis binárias (1 = presença; 0 = ausência)

Tabela

O conteúdo gerado por IA pode estar incorreto.

Linha do tempo

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

**Métodos de Agrupamentos Hierárquicos**

O agrupamento hierárquico é uma técnica de clusterização que constrói uma hierarquia de clusters, organizando-os em uma árvore de relações chamada dendrograma. Diferente de métodos como K-Means, que exigem a definição prévia do número de clusters, o clustering hierárquico permite uma análise visual para determinar a melhor segmentação.

**Método Hierárquico Aglomerativos**

* No início, cada indivíduo representa um grupo.
* Para esses métodos os agrupamentos são formados a partir de uma matriz de parecença.
* A cada iteração, os clusters mais semelhantes são mesclados até que reste apenas um cluster contendo todos os dados.
* Queremos identificar os objetos que mais se parecem
* Agrupamos esses objetos e os consideramos como um único objeto
* Definimos uma nova matriz de parecença.

**Passos:**

Uma imagem contendo Linha do tempo

O conteúdo gerado por IA pode estar incorreto.

Após a formação do primeiro cluster, é preciso definir como a distância entre os dois clusters será medida. Os principais métodos são:

* Método do vizinho mais próximo (Single Linkage/Nearest Neighbor);
  + A distância entre os grupos é definida como sendo a distância entre os elementos mais próximos (menor distância) dos dois grupos
* Método do vizinho mais longe (Complete Linkage/Farthet Neighbor);
  + A distância entre dois grupos é definida como sendo a menor das mais distantes entre os indivíduos dos dois grupos (distância máxima)
* Método das médias das distâncias (Average Linkage/Between Groups);
  + A distância entre dois grupos é definida como sendo a distância média entre todos os pares de indivíduos dos dois grupos, buscando-se a menor distância média entre os grupos.
* Método da centróide;
  + Este método define a coordenada de cada grupo como sendo a média das coordenadas de seus objetos. Uma vez obtida essa coordenada, denominada centroide, a distância entre os grupos é obtida através do cálculo das distâncias entre os centroides.
* Método de Ward.
  + O método de Ward busca unir objetos que tornem os agrupamentos formados os mais homogêneos possível. A medida de homogeneidade utilizada baseia-se na partição da soma de quadrados total de uma análise de variância. O Método de Ward, é atraente por basear-se numa medida com forte apelo estatístico e por gerar grupos que, assim como os do método do vizinho mais longe, possuem alta homogeneidade interna.

**Métodos de Agrupamento Não Hierárquicos**

Os métodos de agrupamento não hierárquicos são técnicas que não dependem da criação de uma estrutura hierárquica como o dendrograma. Em vez disso, eles trabalham atribuindo diretamente os pontos a clusters e ajustando essas atribuições ao longo das iterações.

São **mais eficientes** para grandes volumes de dados, pois evitam o alto custo computacional dos métodos hierárquicos

* Não exige uma estrutura hierárquica, os grupos são formados diretamente.
* Normalmente requer a definição prévia do número de clusters.
* Os algoritmos geralmente funcionam com iterações sucessivas, refinando os agrupamentos a cada ciclo.
* Mais rápido e escalável do que os métodos hierárquicos.

**Método K-Means**

Um dos mais populares.

O número de clusters **k** deve ser definido antes do início do algoritmo.

Funciona iterativamente:

* Seleciona **k** centroides iniciais (aleatórios ou baseados em heurísticas).
* Cada ponto é atribuído ao cluster mais próximo (usando a distância Euclidiana, por exemplo).
* Os centroides dos clusters são recalculados.
* O processo continua até que os clusters não mudem significativamente.

**Vantagens**

Rápido e eficiente para grandes conjuntos de dados.  
Fácil de interpretar.  
Bom para clusters esféricos e bem separados.

**Desvantagens**

Requer definição prévia de k.  
Sensível a outliers e inicialização dos centroides.  
Não funciona bem para clusters de formas irregulares

**Método DBScan**

Baseado em densidade, não requer a definição de K

Forma Clusters baseando-se na densidade dos pontos

Identifica outliers como ruído, não os agrupando em nenhum cluster

**Vantagens**

Encontra cluster de qualquer formato  
Não exige a definição de K  
Resistente a outliers

**Desvantagens**

Requer definir parâmetros (raio de vizinhança) e minPts (Mínimo pontos para formar um cluster)  
Não funciona bem se os clusters tem densidades muito diferentes

**Tabela ANOVA**

Texto

O conteúdo gerado por IA pode estar incorreto.

**Resumindo**

**Análise de Cluster: Etapas, Conceitos e Interpretações**

A análise de cluster é uma técnica estatística de aprendizado não supervisionado utilizada para **agrupar observações** com características semelhantes, sem utilizar rótulos previamente definidos. O objetivo é identificar estruturas ou padrões ocultos nos dados, de forma exploratória. O processo envolve diversas etapas, começando pela preparação dos dados e passando por métodos como **clusterização hierárquica** e **KMeans**, além de ferramentas auxiliares como a **tabela ANOVA**, escolha de distâncias e regras de parada.

**1. Preparação dos Dados**

Antes de iniciar a clusterização, é essencial:

* Remover ou tratar valores ausentes (nulos).
* Analisar a presença de outliers.
* Selecionar apenas variáveis numéricas ou que possam ser convertidas.
* Padronizar os dados com técnicas como **StandardScaler**, garantindo que todas as variáveis estejam na mesma escala. Isso é necessário pois os algoritmos de cluster utilizam medidas de distância, que podem ser distorcidas por variáveis com ordens de grandeza diferentes.

**2. Clusterização Hierárquica (Método Exploratória Inicial)**

A clusterização hierárquica é uma técnica que agrupa os dados de forma progressiva, criando uma **árvore de agrupamentos (dendrograma)**. É frequentemente utilizada como uma **etapa exploratória antes de aplicar o KMeans**, para ajudar na escolha do número ideal de clusters.

**2.1 Escolha da Métrica de Distância**

A distância entre observações é fundamental na formação dos grupos. As principais métricas incluem:

* **Euclidiana**: distância geométrica padrão.
* **Manhattan**: soma das diferenças absolutas.
* **Minkowski**: generalização das anteriores.
* **Distância de Mahalanobis**: leva em conta a correlação entre variáveis.

A escolha da métrica depende do tipo de dado e da análise desejada, sendo a euclidiana a mais comum para variáveis padronizadas.

**2.2 Esquema de Agrupamento (Linkage)**

Define como a distância entre dois grupos é medida. Os principais métodos são:

* **Single linkage**: menor distância entre pontos de dois grupos.
* **Complete linkage**: maior distância entre pontos de dois grupos.
* **Average linkage**: média das distâncias entre todos os pares de pontos.
* **Ward**: minimiza a soma dos quadrados dentro dos grupos (ótimo para dados numéricos).

**2.3 Dendrograma e Regra da Parada**

O dendrograma é um gráfico em árvore que mostra como os grupos foram unidos em cada etapa. A **regra da parada** consiste em definir onde “cortar” a árvore para determinar o número de clusters. Isso pode ser feito de forma visual (maior distância entre fusões consecutivas) ou por meio de uma tabela auxiliar.

**3. Tabela de Incremento da Heterogeneidade**

Durante a clusterização hierárquica, a fusão de grupos aumenta a heterogeneidade (ou seja, os grupos ficam menos homogêneos). Uma tabela pode ser construída com:

* Número de agrupamentos restantes.
* Soma dos quadrados intra-grupos (ou inércia).
* Incremento de heterogeneidade em cada junção.

A **fusão que gera o maior incremento** na heterogeneidade costuma indicar um ponto ótimo para parar a formação de grupos. O ideal é **escolher o número de clusters anterior a esse ponto**.

**4. Aplicação do KMeans**

Com base na análise hierárquica, aplica-se o algoritmo **KMeans**, agora com o número de clusters escolhido previamente.

O KMeans funciona por:

* Inicializar k centroides.
* Atribuir cada ponto ao centro mais próximo.
* Recalcular os centroides com base na média dos pontos atribuídos.
* Repetir até convergência.

O resultado é um agrupamento mais preciso e ajustado, ideal para grandes bases de dados e análise quantitativa.

**5. Análise dos Resultados**

Após o agrupamento, realiza-se:

* **Avaliação dos centroides**: cada cluster é representado por um centroide (média das variáveis do grupo), que define seu perfil.
* **Tamanho dos grupos**: quantas observações cada cluster possui.
* **Tabela de Médias por Cluster**: comparar as médias de cada variável entre os grupos.

**6. ANOVA para Validação dos Grupos**

A ANOVA (Análise de Variância) é usada para verificar se **as variáveis realmente contribuem para diferenciar os grupos**. Para cada variável:

* Calcula-se a **F-estatística** comparando a variância entre grupos com a variância dentro dos grupos.
* Se o **p-valor < 0,05**, a variável é considerada estatisticamente significativa para distinguir os clusters.

**7. Interpretação Final**

Com os clusters formados e validados, pode-se:

* Nomear os grupos com base nos seus perfis (ex: “grupo de alto consumo”, “grupo conservador”).
* Visualizar os clusters com técnicas de redução de dimensionalidade (PCA, t-SNE).
* Integrar os clusters em modelos de negócio, marketing, segmentação, etc.

**Conclusão**

A análise de clusters é uma poderosa ferramenta para **descobrir padrões ocultos** em bases de dados. A combinação da clusterização hierárquica (exploratória) com KMeans (precisa) e validação por ANOVA garante robustez e interpretabilidade. Ao longo do processo, a escolha de distâncias, métodos de agrupamento, análise de heterogeneidade e perfil dos grupos são passos fundamentais para uma segmentação eficaz e aplicável.

**Redes Neurais e Deep Learning**

Redes Neurais é uma família de algoritmos que está dentro do machine learning. Assim, deep learning, é uma rede neural com camadas mais complexas.

Redes Neurais Artificiais

As redes neurais artificiais buscam simular as redes neurais biológicas do nosso cérebro no computador, e a base de uma rede, é o neurônio artificial, onde o primeiro neurônio foi o Perceptron.

A rede neural tem pesos, que irão passar dentro de uma função ativação que de acordo com esses pesos, ela irá ter um resultado.

**Banco de Dados**

**SQL (Structured Query Language)**

É uma linguagem de consulta estruturada para dados relacionais/estruturados, é também é uma linguagem declarativa, onde dizemos o que queremos fazer, não precisamos especificar como fazer (nas imperativas). Assim sendo eficiente para dados normalizados (dados que estão formatados a forma de reduzir redundâncias ou problemas de integridade)

**SGBD (Sistema Gerenciador de Banco de Dados**)

É um software que permite criar, gerenciar e manipular bancos de dados. Serve como uma interface entre o usuário, os aplicativos e o banco de dados, garantindo que os dados sejam armazenados, organizados e acessados de forma eficiente e segura.

* **Armazenamento e Organização de Dados →** Mantém os dados estruturados em tabelas.
* **Manipulação de Dados →** Permite inserir, atualizar, deletar e consultar dados via SQL.
* **Controle de Acesso →** Define permissões para usuários e protege informações.
* **Gerenciamento de Transações →** Garante consistência e integridade dos dados.
* **Otimização de Performance →** Usa índices e cache para melhorar velocidade.

**Elementos Básicos**

* **Tabelas**: Estruturas que armazenam dados organizados em linhas e colunas.
* **Registros (Linhas)**: Cada entrada na tabela.
* **Colunas (Atributos)**: Características dos dados (ex: nome, idade).
* **Esquema**: Estrutura do banco (quais tabelas e como se relacionam).

**Relacionamentos entre Tabelas**

* **1:1 (Um para Um)** → Exemplo: Pessoa e Passaporte.
* **1:N (Um para Muitos)** → Exemplo: Cliente e Pedidos.
* **N:N (Muitos para Muitos)** → Exemplo: Estudantes e Disciplinas.

**Chaves Primárias e Estrangeiras**

* **Chave Primária (Primary Key):** é um identificador único de cada linha de uma tabela de banco de dados
* **Chave Estrangeira (Foreigner Key):** é uma referência em uma tabela a uma chave primária de outra tabela.

**Dependência Funcional**

A dependência funcional (DF) ocorre quando o valor de um atributo determina o valor de outro atributo dentro de uma tabela.

Dado um conjunto de atributos X e um atributo Y, dizemos que Y é funcionalmente dependente de X se, para cada valor de X, existe um único valor correspondente de Y.

**Normalização** (**IMPORTANTE)**

A normalização é o processo de organizar um banco de dados para reduzir a redundância e melhorar a integridade dos dados. É feita por meio de formas normais (FNs).

* **1ª Forma Normal (1NF)**: Todos os atributos devem conter valores atômicos (sem listas ou conjuntos dentro de uma célula). E cada coluna deve conter apenas um único valor por registro.
* **2ª Forma Normal (2NF)**: Nenhum atributo não chave deve depender apenas de uma parte da chave primaria (caso a chave primaria seja composta) ele deve depender de ambas as partes da PK
* **3ª Forma Normal (3NF)**: Nenhum atributo não chave deve depender outro atributo não chave. Ou seja, todos os atributos que não fazem parte da PK, não devem possuir nenhuma dependência entre si.

**Tipos de Dados usados para criação dos atributos**

**Números inteiros**

* **SMALLINT**: Inteiro pequeno (2 bytes, -32.768 a 32.767)
* **INTEGER / INT**: Inteiro padrão (4 bytes, -2.147.483.648 a 2.147.483.647)
* **BIGINT**: Inteiro grande (8 bytes, para valores muito grandes)

**Números decimais**

* **NUMERIC(p, s)**: Precisão exata, usado para valores como dinheiro (ex: NUMERIC(10, 2))
* **DECIMAL(p, s)**: Igual ao NUMERIC
* **REAL**: Ponto flutuante (4 bytes), precisão aproximada
* **DOUBLE PRECISION**: Ponto flutuante (8 bytes), maior precisão

**Texto**

* **CHAR(n)**: Texto de tamanho fixo (ex: CHAR(10) ocupa sempre 10 caracteres)
* **VARCHAR(n)**: Texto de tamanho variável até n caracteres
* **TEXT**: Texto de tamanho ilimitado

**Data e Hora**

* **DATE**: Apenas data (formato: AAAA-MM-DD)
* **TIME**: Apenas hora (formato: HH:MM:SS)
* **TIMESTAMP**: Data e hora combinadas
* **TIMESTAMPTZ**: Data e hora com fuso horário

**Booleano**

* **BOOLEAN**: Pode armazenar TRUE, FALSE ou NULL

**Identidade (auto incremento)**

* **SMALLSERIAL**: Inteiro auto-incrementável pequeno (2 bytes)
* **SERIAL**: Inteiro auto-incrementável padrão (4 bytes)
* **BIGSERIAL**: Inteiro auto-incrementável grande (8 bytes)

**Binário**

* **BYTEA**: Armazena dados binários, como arquivos ou imagens

**Outros Especiais**

* **UUID**: Identificador único universal (ex: usado como chave primária)
* **JSON / JSONB**: Armazena dados em formato JSON. JSONB é otimizado para busca
* **ARRAY**: Armazena listas de valores (ex: TEXT[], INTEGER[])
* **ENUM**: Conjunto fixo de valores possíveis (ex: 'pequeno', 'médio', 'grande')

**DDL (Data Definition Language)**

Usado para definir e estruturar um banco de dados. Esses comandos não manipulam os dados, apenas criam ou alteram tabelas e outros objetos.

* **CREATE:** Cria tabelas, bancos de dados, índices etc.
* **ALTER**: Modifica a estrutura de tabelas (adicionar/remover colunas).
* **DROP**: Remove tabelas ou bancos de dados.
* **TRUNCATE**: Remove todos os dados de uma tabela sem apagar a estrutura.

**CREATE SCHEMA** (Cria um esquema de BD relacional)

|  |
| --- |
| CREATE SCHEMA vendas |

**DROP SCHEMA** (Remove um esquema de BD relacional)

* **CASCADE** (Remove o esquema BD incluindo todas suas tabelas e outros elementos)
* **RESTRICT** (Remove um esquema BD somente se não existir elementos definidos para esse esquema)

|  |
| --- |
| DROP SCHEMA vendas RESTRICT;  DROP SCHEMA vendas CASCADE; |

Obs: o esquema serve para agrupar as tabelas e outros comandos que pertencem à mesma aplicação

**CREATE TABLE** (Cria uma nova tabela – relação – no BD, ela não possui dados inicialmente)

|  |
| --- |
| CREATE TABLE clientes (  id SERIAL PRIMARY KEY,  nome VARCHAR(100) NOT NULL,  email VARCHAR(100) UNIQUE,  idade INTEGER,  criado\_em TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  ); |

**DROP TABLE** (remove uma tabela – relação – e todas suas instâncias do BD)

|  |
| --- |
| DROP TABLE clientes; |

**ALTER TABLE** (Altera a estrutura de uma tabela – relação – já existente no BD)

|  |
| --- |
| -- Adicionar uma coluna  ALTER TABLE clientes ADD COLUMN telefone VARCHAR(20);  -- Modificar o tipo de uma coluna  ALTER TABLE clientes ALTER COLUMN idade TYPE SMALLINT;  -- Renomear uma coluna  ALTER TABLE clientes RENAME COLUMN telefone TO celular;  -- Renomear a tabela  ALTER TABLE clientes RENAME TO consumidores;  -- Remover uma coluna  ALTER TABLE consumidores DROP COLUMN celular; |

**Restrição de Integridade**

**Valor nulo (NULL)**: Não precisa de um valor específico

|  |
| --- |
| CREATE TABLE produtos (  id SERIAL PRIMARY KEY,  nome VARCHAR(100),  descricao TEXT NULL  ); |

**Restrição não nula (NOT NULL):** Usamos quando não é permitido um valor nulo

|  |
| --- |
| CREATE TABLE categorias (  id SERIAL PRIMARY KEY,  nome VARCHAR(100) NOT NULL  ); |

**Comparações em consultas:** Usar “IS NULL” e “IS NOT NULL”

|  |
| --- |
| -- Buscar clientes sem email cadastrado  SELECT \* FROM clientes WHERE email IS NULL;  -- Buscar clientes com email cadastrado  SELECT \* FROM clientes WHERE email IS NOT NULL; |

**Cláusula DEFAULT:** Associa um valor predeterminado para um atributo caso nenhum seja especificado

|  |
| --- |
| CREATE TABLE pedidos (  id SERIAL PRIMARY KEY,  status VARCHAR(20) DEFAULT 'pendente'  ); |

**Cláusula CHECK:** Especifica um predicado que precisa ser satisfeito por todas as tuplas de uma relação

|  |
| --- |
| CREATE TABLE funcionarios (  id SERIAL PRIMARY KEY,  nome VARCHAR(100),  salario NUMERIC CHECK (salario > 0)  ); |

**Cláusula PRIMARY KEY:** Identifica os atributos da relação que formam a sua chave primária (obrigatoriamente NOT NULL)

|  |
| --- |
| CREATE TABLE departamentos (  id INTEGER PRIMARY KEY,  nome VARCHAR(50) ); |

**Cláusula UNIQUE:** Não permite valores duplicados para um determinado atributo

|  |
| --- |
| CREATE TABLE usuarios (  id SERIAL PRIMARY KEY,  nome\_usuario VARCHAR(50) UNIQUE  ); |

**Cláusula FOREIGN KEY:** Integridade referencial (dependência existente entre a chave estrangeira de uma relação e a chave primária da relação relacionada). Com essa cláusula eliminamos a possibilidade de violação da integridade referencial.

|  |
| --- |
| CREATE TABLE pedidos (  id SERIAL PRIMARY KEY,  cliente\_id INTEGER,  FOREIGN KEY (cliente\_id) REFERENCES clientes(id)  ); |

**CREATE DOMAIN** (Domínio: Conjunto de valores válidos que um atributo pode assumir, podendo ser usados dentro da criação de tabelas)

|  |
| --- |
| CREATE DOMAIN idade\_valida AS INT  CHECK (VALUE >= 0 AND VALUE <= 120);  CREATE TABLE pessoa (  nome TEXT,  idade idade\_valida  ); |

**DROP DOMAIN**

|  |
| --- |
| DROP DOMAIN idade\_valida; |

**ALTER DOMAIN**

|  |
| --- |
| -- renomear domínio  ALTER DOMAIN idade\_valida RENAME TO idade\_limitada;  -- adicionar uma restrição  ALTER DOMAIN idade\_limitada ADD CONSTRAINT idade\_minima CHECK (VALUE >= 18);  -- remover uma restrição  ALTER DOMAIN idade\_limitada DROP CONSTRAINT idade\_minima; |

Obs: As características de um domínio são globais ao BD

**CREATE INDEX** (Índice: Estrutura de dados usada para acelerar a busca de informações dentro de uma tabela)

|  |
| --- |
| -- Cria índice simples na coluna 'nome' da tabela 'cliente'  CREATE INDEX idx\_nome ON cliente(nome);  -- Cria um índice único na coluna 'email' da tabela 'cliente'  CREATE UNIQUE INDEX idx\_email\_unico ON cliente(email);  -- Cria índice em múltiplas colunas  CREATE INDEX idx\_nome\_sobrenome ON cliente(nome, sobrenome);  -- Cria índice ordenado de forma descendente  CREATE INDEX idx\_data\_desc ON pedidos(data\_compra DESC);  -- Cria índice parcial (somente para clientes ativos)  CREATE INDEX idx\_ativos ON cliente(situacao) WHERE situacao = 'ativo';  -- Cria índice com função (para buscas case-insensitive)  CREATE INDEX idx\_lower\_email ON cliente(LOWER(email)); |

**DROP INDEX**

|  |
| --- |
| DROP INDEX idx\_nome;  -- Remove índice apenas se existir (evita erro)  DROP INDEX IF EXISTS idx\_nome; |

**ALTER INDEX**

|  |
| --- |
| -- Renomeia o índice  ALTER INDEX idx\_email\_unico RENAME TO idx\_cliente\_email\_unico;  -- Altera o proprietário do índice  ALTER INDEX idx\_cliente\_email\_unico OWNER TO novo\_usuario;  -- Reconstrói o índice (útil após muitas mudanças na tabela)  REINDEX INDEX idx\_cliente\_email\_unico; |

**DML (Data Manipulation Language)**

Usado para inserir, atualizar e excluir dados dentro das tabelas.

* **INSERT:** Insere novos dados na tabela.
* **UPDATE:** Atualiza dados existentes.
* **DELETE:** Remove registros de uma tabela.
* **SELECT:** (embora seja mais considerado DQL, está dentro da manipulação).

**INSERT INTO** (Para inserir uma linha na relação – tabela)

|  |
| --- |
| -- Inserção completa (todos os campos)  INSERT INTO clientes (nome, email, cidade, ativo)  VALUES ('Ana Souza', 'ana@email.com', 'São Paulo', TRUE);  -- Inserção parcial (colunas não incluídas usarão o valor DEFAULT ou NULL)  INSERT INTO clientes (nome, email)  VALUES ('Carlos Lima', 'carlos@email.com'); |

**UPDATE** (Atualizar atributos de linhas)

|  |
| --- |
| -- Atualiza o email e a cidade do cliente com id = 1  UPDATE clientes  SET email = 'ana.souza@email.com',  cidade = 'Rio de Janeiro'  WHERE id = 1;  -- Desativa todos os clientes da cidade de São Paulo  UPDATE clientes  SET ativo = FALSE  WHERE cidade = 'São Paulo'; |

**DELETE** (Excluir linhas da relação)

|  |
| --- |
| -- Remove o cliente com id = 2  DELETE FROM clientes  WHERE id = 2;  -- Remove todos os clientes inativos  DELETE FROM clientes  WHERE ativo = FALSE; |

**DQL (Data Query Language)**

A DQL é uma subcategoria do SQL usada exclusivamente para consultas. O principal comando aqui é o SELECT, que recupera dados de uma ou mais tabelas

* **WHERE:** filtra os dados com base em condições.
* **ORDER BY:** ordena os resultados.
* **GROUP BY:** agrupa dados (usado com funções de agregação).
* **HAVING:** filtra grupos após um GROUP BY.
* **JOIN:** permite combinar dados de várias tabelas.

**Formas de fazer pesquisa/seleções de linhas e ou colunas**

**Cláusula SELECT:** Lista os atributos e/ou funções a serem exibidas no resultado da consulta

**Cláusula FROM:** Especifica as relações (tabelas) a serem examinadas na avaliação da consulta

**Cláusula WHERE**: Especifica as condições para a seleção das tuplas no resultado da consulta, as condições devem ser definidas sobre os atributos das relações que aparecem na cláusula FROM. Inclui condições de junções

* Operadores: AND, OR e NOT

Tabela

O conteúdo gerado por IA pode estar incorreto.

* **Operador LIKE:** 
  + Caractere coringa: “%” substitui qualquer string e “\_” substitui qualquer caractere
  + Sensível a letra maiúscula e minúsculas

|  |
| --- |
| -- Seleciona todos os dados da tabela  SELECT \* FROM clientes;  -- Seleciona apenas os nomes e e-mails de todos os clientes  SELECT nome, email FROM clientes;  -- Seleciona clientes que moram em 'São Paulo'  SELECT \* FROM clientes  WHERE cidade = 'São Paulo';  -- Seleciona clientes com mais de 30 anos  SELECT \* FROM clientes  WHERE idade > 30;  -- Seleciona clientes entre 18 e 25 anos  SELECT \* FROM clientes  WHERE idade BETWEEN 18 AND 25;  -- Seleciona clientes inativos  SELECT \* FROM clientes  WHERE ativo = FALSE;  -- Clientes com idade diferente de 40  SELECT \* FROM clientes  WHERE idade != 40;  -- Clientes com idade maior ou igual a 60  SELECT \* FROM clientes  WHERE idade >= 60;  -- Clientes com cidade 'São Paulo' ou 'Rio de Janeiro'  SELECT \* FROM clientes  WHERE cidade IN ('São Paulo', 'Rio de Janeiro');  -- Clientes que não estão nessas cidades  SELECT \* FROM clientes  WHERE cidade NOT IN ('Curitiba', 'Belo Horizonte');  -- Clientes cujo nome começa com 'A'  SELECT \* FROM clientes  WHERE nome LIKE 'A%';  -- Clientes cujo nome termina com 's'  SELECT \* FROM clientes  WHERE nome LIKE '%s';  -- Clientes cujo nome contém 'silva'  SELECT \* FROM clientes  WHERE nome LIKE '%silva%';  -- Clientes cujo email é do domínio gmail  SELECT \* FROM clientes  WHERE email LIKE '%@gmail.com';  -- Clientes cujo nome tem exatamente 5 letras  SELECT \* FROM clientes  WHERE nome LIKE '\_\_\_\_\_'; -- 5 underlines = 5 letras  -- Clientes ativos com mais de 25 anos  SELECT \* FROM clientes  WHERE ativo = TRUE AND idade > 25;  -- Clientes inativos ou com idade abaixo de 18  SELECT \* FROM clientes  WHERE ativo = FALSE OR idade < 18;  -- Clientes com nome contendo 'joão' e morando no RJ  SELECT \* FROM clientes  WHERE LOWER(nome) LIKE '%joão%' AND cidade = 'Rio de Janeiro'; |

**Cláusula ORDER BY:** Ordena as tuplas que aparecem no resultado da pesquisa, ASC (ascendente) ou DESC(descendente)

|  |
| --- |
| -- Ordena por valor em ordem crescente  SELECT \* FROM pedidos  ORDER BY valor;  -- Ordena por valor em ordem decrescente  SELECT \* FROM pedidos  ORDER BY valor DESC;  -- Ordena por cidade (A-Z) e, dentro da cidade, pelo valor (maior para menor)  SELECT \* FROM pedidos  ORDER BY cidade ASC, valor DESC; |

**SELECT DISTINCT / ALL:** Distinct não considera duplas duplicas e All considera todas tuplas

|  |
| --- |
| -- Lista todas as cidades únicas dos pedidos (sem repetições)  SELECT DISTINCT cidade FROM pedidos;  -- Clientes únicos que fizeram pedidos  SELECT DISTINCT cliente FROM pedidos;  -- Esse comando é igual a SELECT padrão, pois ALL é implícito  SELECT ALL cliente, cidade FROM pedidos; |

**Cláusula GROUP BY:** Permite aplicar uma função de agregação não somente a um conjunto de tuplas, mas também a um grupo de conjunto de tuplas

|  |
| --- |
| -- Soma dos pedidos por cliente  SELECT cliente, SUM(valor) AS total\_gasto  FROM pedidos  GROUP BY cliente;  -- Quantidade de pedidos por cidade  SELECT cidade, COUNT(\*) AS total\_pedidos  FROM pedidos  GROUP BY cidade; |

**Cláusula HAVING:** Permite especificar uma condição de seleção para grupos

|  |
| --- |
| -- Clientes que gastaram mais de 500 no total  SELECT cliente, SUM(valor) AS total\_gasto  FROM pedidos  GROUP BY cliente  HAVING SUM(valor) > 500;  -- Cidades com mais de 2 pedidos  SELECT cidade, COUNT(\*) AS total\_pedidos  FROM pedidos  GROUP BY cidade  HAVING COUNT(\*) > 2; |

**Funções de Agregação**

**Média:** AVG()

**Mínimo:** MIN()

**Máximo:** MAX()

**Soma:** SUM()

**Contagem:** COUNT()

|  |
| --- |
| -- Média de valores de todos os pedidos  SELECT AVG(valor) AS media\_pedidos  FROM pedidos;  -- Média de valor por cidade  SELECT cidade, AVG(valor) AS media\_por\_cidade  FROM pedidos  GROUP BY cidade;  -- Pedido mais barato  SELECT MIN(valor) AS menor\_valor  FROM pedidos;  -- Menor pedido por status  SELECT status, MIN(valor) AS pedido\_minimo  FROM pedidos  GROUP BY status;  -- Pedido mais caro  SELECT MAX(valor) AS maior\_valor  FROM pedidos;  -- Maior valor de pedido por cliente  SELECT cliente, MAX(valor) AS pedido\_mais\_caro  FROM pedidos  GROUP BY cliente;  -- Soma total de todos os pedidos  SELECT SUM(valor) AS total\_vendas  FROM pedidos;  -- Total vendido por cidade  SELECT cidade, SUM(valor) AS total\_por\_cidade  FROM pedidos  GROUP BY cidade;  -- Total de pedidos  SELECT COUNT(\*) AS total\_pedidos  FROM pedidos;  -- Total de clientes únicos  SELECT COUNT(DISTINCT cliente) AS total\_clientes  FROM pedidos;  -- Número de pedidos por status  SELECT status, COUNT(\*) AS total\_por\_status  FROM pedidos  GROUP BY status; |

**Junções**

**Cláusula ON:** usada em JOINs para especificar a condição de junção entre duas tabelas — ou seja, qual coluna em uma tabela se relaciona com qual coluna da outra.

**CROSS JOIN:** Faz o produto cartesiano: cada linha da primeira tabela é combinada com todas as linhas da segunda.

|  |
| --- |
| SELECT clientes.nome, pedidos.valor  FROM clientes  CROSS JOIN pedidos; |

**INNER JOIN:** Retorna apenas as linhas que têm correspondência em ambas as tabelas.

|  |
| --- |
| -- Mostra só os clientes que fizeram pedidos  SELECT clientes.nome, pedidos.valor  FROM clientes  INNER JOIN pedidos ON clientes.id = pedidos.cliente\_id; |

**LEFT JOIN:** Retorna todas as linhas da tabela da esquerda e os dados correspondentes da tabela da direita. Se não houver correspondência, os campos da direita vêm como NULL.

|  |
| --- |
| -- Mostra todos os clientes, mesmo os que não fizeram pedidos  SELECT clientes.nome, pedidos.valor  FROM clientes  LEFT JOIN pedidos ON clientes.id = pedidos.cliente\_id; |

**RIGHT JOIN:** Retorna todas as linhas da tabela da direita, e os dados correspondentes da esquerda. Se não houver correspondência, os campos da esquerda vêm como NULL.

|  |
| --- |
| -- Mostra todos os pedidos, mesmo os que estão sem clientes associados  SELECT clientes.nome, pedidos.valor  FROM clientes  RIGHT JOIN pedidos ON clientes.id = pedidos.cliente\_id; |

**FULL JOIN**: Retorna todas as linhas de ambas as tabelas, com correspondência onde houver. Onde não houver, preenche com NULL.

|  |
| --- |
| -- Junta todos os clientes e todos os pedidos, inclusive os que não tem  -- correspondência entre si  SELECT clientes.nome, pedidos.valor  FROM clientes  FULL JOIN pedidos ON clientes.id = pedidos.cliente\_id; |

**SELF JOIN:** É quando uma tabela faz JOIN com ela mesma.

|  |
| --- |
| SELECT A.nome AS cliente1, B.nome AS cliente2  FROM clientes A  JOIN clientes B ON A.id != B.id; |

**SELECTs aninhados**

são consultas dentro de outras consultas. Ou seja, você coloca um comando SELECT dentro de outro para utilizar o resultado de uma consulta como parte de outra. Essas subconsultas podem ser usadas em diversas partes da consulta principal, como no SELECT, FROM, WHERE, e até em HAVING.

**Exemplos:**

|  |
| --- |
| -- EXEMPLOS DE SELECTS ANINHADOS (SUBCONSULTAS)  -- Exemplo 1: Subconsulta no WHERE  -- Objetivo: Buscar os clientes que fizeram pedidos com valor maior que o maior valor do cliente com id = 1  SELECT nome  FROM clientes  WHERE id IN (  SELECT cliente\_id  FROM pedidos  WHERE valor > (  SELECT MAX(valor)  FROM pedidos  WHERE cliente\_id = 1  )  );  -- Exemplo 2: Subconsulta no FROM  -- Objetivo: Mostrar o total de pedidos apenas para clientes que fizeram mais de 2 pedidos  SELECT cliente\_id, SUM(valor) AS total\_pedidos  FROM (  SELECT cliente\_id, valor  FROM pedidos  WHERE cliente\_id IN (  SELECT cliente\_id  FROM pedidos  GROUP BY cliente\_id  HAVING COUNT(\*) > 2  )  ) AS pedidos\_filtrados  GROUP BY cliente\_id;  -- Exemplo 3: Subconsulta no SELECT  -- Objetivo: Mostrar o nome de cada cliente e o total que ele já gastou (soma dos valores de seus pedidos)  SELECT nome, (  SELECT SUM(valor)  FROM pedidos  WHERE cliente\_id = clientes.id  ) AS total\_gasto  FROM clientes;  -- Exemplo 4: Subconsulta correlacionada  -- Objetivo: Para cada pedido, mostrar se ele foi o maior feito por aquele cliente  SELECT id, cliente\_id, valor,  CASE  WHEN valor = (  SELECT MAX(valor)  FROM pedidos AS p2  WHERE p2.cliente\_id = p1.cliente\_id  )  THEN 'MAIOR PEDIDO'  ELSE 'NORMAL'  END AS status\_pedido  FROM pedidos AS p1;  -- Exemplo 5: Subconsulta simples com comparação  -- Objetivo: Mostrar os pedidos que têm valor maior que a média geral  SELECT \*  FROM pedidos  WHERE valor > (  SELECT AVG(valor)  FROM pedidos  ); |