Sumário

[Tratamento de Valores Ausentes 1](#_Toc190772895)

[Codificação de Variáveis Categóricas 1](#_Toc190772896)

[Normalização ou Padronização 1](#_Toc190772897)

[Conversão de Tipos de Dados 2](#_Toc190772898)

[Remoção de Dados Duplicados 2](#_Toc190772899)

[Agregação de Dados 2](#_Toc190772900)

[Criação de Novas Variáveis (Feature Engineering) 2](#_Toc190772901)

[Divisão de Dados 2](#_Toc190772902)

[Renomeação de Colunas 2](#_Toc190772903)

[Mudança do Formato do DataFrame 2](#_Toc190772904)

[Expressão Regular para Limpeza de Dados 2](#_Toc190772905)

[get\_dummies e sep 3](#_Toc190772906)

[Pipelines do Scikit-learn 3](#_Toc190772907)

[Pipeline de Dados 3](#_Toc190772908)

[Diferença entre Data Wrangling e Data Pipeline 3](#_Toc190772909)

[ETL (Extract, Transform, Load) 3](#_Toc190772910)

[O Que é? 3](#_Toc190772911)

[Por Que Limpar Dados? 3](#_Toc190772912)

[Manipulando o DataFrame para Limpeza 3](#_Toc190772913)

[Lidando com Dados Faltosos 4](#_Toc190772914)

# Tratamento de Valores Ausentes

Valores ausentes podem causar problemas significativos durante a análise de dados. É essencial preencher (imputar) ou remover esses valores para garantir a integridade dos resultados. Métodos comuns incluem a imputação com a média, mediana ou moda, ou a remoção de linhas ou colunas inteiras que contêm valores ausentes.

# Codificação de Variáveis Categóricas

Variáveis categóricas precisam ser convertidas em uma forma numérica para serem utilizadas em algoritmos de machine learning. Uma técnica popular é o one-hot encoding, que cria colunas binárias para cada categoria, permitindo que os algoritmos interpretem corretamente os dados categóricos.

# Normalização ou Padronização

Para garantir que os valores numéricos estejam em uma escala comum, utilizamos técnicas de normalização ou padronização. A normalização escala os valores para um intervalo de 0 a 1, enquanto a padronização ajusta os dados para que tenham média 0 e desvio padrão 1. Isso é crucial para algoritmos que são sensíveis à escala dos dados.

# Conversão de Tipos de Dados

Garantir que os tipos de dados das colunas estão corretos é fundamental para a análise. Isso pode incluir a conversão de strings em datas, números inteiros, floats, entre outros. Ferramentas como pandas facilitam essas conversões com métodos como pd.to\_datetime() e pd.to\_numeric().

# Remoção de Dados Duplicados

Dados duplicados podem distorcer os resultados da análise. Identificar e remover essas entradas é essencial para manter a precisão. Em pandas, o método drop\_duplicates() é frequentemente utilizado para essa finalidade.

# Agregação de Dados

Agrupar dados com base em determinadas características e calcular estatísticas agregadas, como média, soma, mínimo e máximo, é uma prática comum. Isso ajuda a resumir e entender grandes conjuntos de dados. Em pandas, métodos como groupby() e agg() são utilizados para essas operações.

# Criação de Novas Variáveis (Feature Engineering)

Criar novas variáveis que possam ser mais informativas para a análise é uma técnica poderosa. Por exemplo, calcular a idade a partir da data de nascimento ou criar indicadores binários a partir de variáveis categóricas. Isso pode melhorar significativamente a performance dos modelos de machine learning.

# Divisão de Dados

Separar conjuntos de dados em subconjuntos com base em condições específicas é útil para análises segmentadas. Isso pode incluir a divisão de dados em conjuntos de treino e teste, ou a segmentação por categorias específicas.

# Renomeação de Colunas

Modificar os nomes das colunas para torná-los mais descritivos e compreensíveis facilita a interpretação dos dados. Em pandas, o método rename() é utilizado para renomear colunas de forma eficiente.

# Mudança do Formato do DataFrame

Reorganizar ou pivotar o DataFrame pode facilitar a análise ou a criação de visualizações. Ferramentas como pivot\_table() e melt() em pandas são úteis para essas transformações.

# Expressão Regular para Limpeza de Dados

Expressões regulares são poderosas para remover caracteres especiais de nomes e telefones, garantindo que os dados estejam limpos e padronizados. Em pandas, o método str.replace() pode ser utilizado em conjunto com expressões regulares.

# get\_dummies e sep

O método get\_dummies() em pandas é utilizado para realizar one-hot encoding de variáveis categóricas. O parâmetro sep pode ser utilizado para especificar o separador em operações de leitura de arquivos, como pd.read\_csv().

# Pipelines do Scikit-learn

Pipelines no scikit-learn permitem a automação de fluxos de trabalho de machine learning, integrando etapas de pré-processamento, transformação e modelagem em uma única sequência. Isso facilita a reprodução e a manutenção dos modelos.

# Pipeline de Dados

Um pipeline de dados envolve as etapas de coleta, transformação, carregamento, armazenamento e monitoramento dos dados. Isso garante que os dados estejam sempre prontos para análise e que qualquer problema seja rapidamente identificado e corrigido.

# Diferença entre Data Wrangling e Data Pipeline

Data wrangling refere-se ao processo de limpeza e transformação de dados brutos em um formato utilizável. Já um data pipeline é uma sequência de etapas automatizadas que movem e transformam dados de uma fonte para um destino, garantindo a integridade e a disponibilidade dos dados.

# ETL (Extract, Transform, Load)

ETL é um processo fundamental em data engineering que envolve a extração de dados de várias fontes, a transformação desses dados em um formato adequado para análise e o carregamento dos dados transformados em um sistema de armazenamento ou banco de dados.

## O Que é?

O processo de manipulação de dados geralmente segue as etapas de exploração, limpeza, transformação e análise. Durante a exploração, os dados são inspecionados para entender sua estrutura e conteúdo. A limpeza envolve a remoção de inconsistências e valores ausentes. A transformação ajusta os dados para o formato necessário para análise, que pode incluir a aplicação de modelos de machine learning ou a criação de dashboards.

## Por Que Limpar Dados?

A limpeza de dados é crucial para garantir a qualidade, consistência, integridade e padronização dos dados. Dados limpos resultam em análises mais precisas e eficientes, previnem vieses e garantem conformidade com regulamentações. Isso melhora a precisão dos modelos de machine learning e a eficiência das operações de análise.

## Manipulando o DataFrame para Limpeza

* Removendo Colunas e Linhas: O pandas oferece uma maneira prática de remover colunas ou linhas indesejadas usando a função drop().
* Mudando o Índice: Utilizar um campo de identificação com valores únicos como índice pode ser útil para a organização dos dados.
* Removendo Duplicados: Identificar e remover entradas duplicadas é essencial para manter a precisão da análise.

## Lidando com Dados Faltosos

* Identificar Dados Faltosos: Utilizar df.isna() para identificar valores ausentes no DataFrame.
* Remover Dados Ausentes: Remover linhas ou colunas com valores ausentes pode ser feito com dropna().
* Preencher Dados Ausentes: A função fillna() permite preencher valores ausentes com uma constante ou com métodos de imputação.
* Imputação Estatística: Utilizar modelos estatísticos para imputar valores ausentes pode melhorar a qualidade dos dados.