



Pré-processamento (Preparação e limpeza de dados)

Higor Alexandre Duarte Mascarenhas



Pré-processamento dos dados

- O pré processamento é uma fase que antecede ao uso dos modelos de machine learning, entretanto para que ele possa ser executado com clareza e também com assertividade, é de fundamental importância que se conheça o processo completo de um projeto de DS.
- Não há a possibilidade de fazer o pré processamento sem saber onde se deseja chegar, se não conhece os modelos computacionais a serem testados e o tipo de resposta que se deseja chegar ao projeto.
- Importante que se entenda o que é possível fazer de modo global, entretanto é fato dizer que somente a experiência de fazer e refazer projetos é que vai tornar a fase de pré processamento assertiva e eficaz.



Pré processamento de dados

- Conjuntos de dados podem apresentar diferentes características, dimensões ou formatos.
- Os dados podem conter ruídos, imperfeições, valores incorretos ou inconsistentes, podem ser duplicados ou ausentes; os atributos podem ser independentes ou correlacionados; os conjuntos de dados podem apresentar poucos ou muitos objetos, que podem ter uma pequena ou grande quantidade de atributos.



Pré processamento de dados

- Técnicas de pré-processamento tem como principal objetivo melhorar a qualidade dos dados e também procurar eliminar elementos que podem criar um falso resultado no processamento dos dados
- Às vezes a fase de pré-processamento tem como objetivo ajustar os dados para um uso mais adequado, modelando-o para que possa ser processado.



Pré processamento de dados

- Por isso é importante conhecer os tipos de dados, as grandezas, para que seja possível identificar as necessidades de ajustes aos quais os dados precisam ser submetidos.
- Um conjunto de técnicas podem ser aplicadas e elas não tem regras, nem sequência, sendo o olhar do cientista de dados e sua experiência que determinam o que precisa ser feito.



Porque fazer a limpeza dos dados

- Em geral em ciência de dados lidamos com um volume de dados muito grande (milhões ou às vezes até bilhões de registros)
- **As bases de dados que lidamos nem sempre estão 100% corretas**
- Isso pode acontecer por diversos motivos:
 - Registro incorreto
 - Erros no sistema que guarda os dados
 - Interações manuais nos dados
 - Perda de registros
 - Regras de negócio (Exemplo: tenho uma cidade dos EUA cadastrado, porém a empresa somente atende no Brasil)



Porque fazer a limpeza dos dados

- Exemplo:
- Temos uma base de dados com o censo realizado pelo IBGE, e em um dos registros temos um habitante de "São Paulo" e em outro registro um habitante de "Sao Paulo".
- Eu deveria considerar que são duas cidades diferentes ou não?
- E se eu tenho um habitante com 143 anos de idade?
- Ou uma família com -1 filhos?
- Teremos situações onde erros são fáceis de identificar, outras nem tanto e estarão atrelados a uma regra de negócio.
- Por exemplo, tenho um registro na minha base de dados de um cliente localizado em Nova Iorque. Porém não sei dizer se é Nova Iorque (EUA) ou Nova Iorque (Maranhão).



Porque fazer a limpeza dos dados

- Saber interpretar os dados da base, e garantir que temos o menor índice de erros possível garantirá a qualidade da nossa análise de dados e do modelo que será criado.
- Dados imprecisos passados adiante na análise ou na etapa de modelagem, podem gerar resultados com alto índice de erros.
- Data Cleaning é o processo de ajuste da base de dados garantindo que ela tenha qualidade e faça sentido para o problema que se pretende resolver.



Integração dos dados

- Dados podem ser oriundos de diversas fontes, de diversos conjuntos de dados e em determinada situação precisam ser integrados;
- Imagine que dados podem ser oriundos de uma API, com informações sobre investimento em marketing e seja preciso integrar com dados de vendas feitas em uma outra plataforma digital.
- Aspectos como: atributos correspondentes, com nomes diferentes em bases distintas; informações correspondentes em bases numéricas diferentes ou moedas (ou idiomas) diferentes.
- Em muitos casos, na integração é necessário compreender os atributos necessários de cada objeto. Lembrando sempre que elevado número de atributos pode comprometer o desempenho dos algoritmos de machine learning.



Eliminação manual de atributos

- Muitas vezes, ao observar um conjunto de dados, fica claro que alguns atributos podem ser eliminados manualmente.
- Retirar os atributos pode estar relacionado, por exemplo, a anonimização de uma base (nome de pessoas muitas das vezes não é necessário).
- Em análises preditivas, quando um atributo não contribui para a estimativa de um valor, ele é irrelevante para a análise e deve ser eliminado.
- Atributos que contém o mesmo valor para todos os objetos também devem ser eliminados, por exemplo o campo cidade em uma base que analisa dados de uma determinada cidade (todos os registros no nosso dataset são pertencentes a mesma cidade).



Alguns dos principais problemas encontrados

- Em geral, os principais problemas que encontramos nas bases de dados são:
 - Dados nulos
 - Dados incorretos (errados ou que não respeitam regras de negócio)
 - Valores repetidos (duplicados, triplicados, etc)
 - Textos escritos de forma não padronizada



Lidando com dados nulos

- Os dados nulos ou ausentes no Python serão identificados como "None" ou "NaN"
- São valores que não tem significado específico e devem sempre ser tratados ou removidos.
- O não tratamento deste tipo de dado, pode gerar inconsistências nos resultados da análise ou modelo ou até mesmo impedir o código de rodar gerando erros de inconsistência.



Lidando com dados nulos

- Para lidar com dados nulos temos algumas estratégias:
- 1. Eliminar os dados (**devemos analisar se a remoção dos registros pode prejudicar a base de dados**)
- 2. Definir e preencher manualmente valores para atributos com valores ausentes;
- 3. Substituir os dados (**para isso é importante ter uma boa interpretação dos dados para entender qual substituição irá descrever melhor os registros**)
 - Nesse caso é importante definir um valor onde saiba-se que era um valor ausente anteriormente
 - Utilizar média, moda ou mediana dos valores conhecidos;
 - Definir indutor baseado em outros atributos.

Dados incompletos

Tabela 3.2 Conjunto de dados com atributos com valores ausentes

Idade	Sexo	Peso	Manchas	Temp.	# Int.	Diagnóstico
—	M	79	—	38,0	—	Doente
18	F	67	Inexistentes	39,5	4	Doente
49	M	92	Espalhadas	38,0	2	Saudável
18	—	43	Inexistentes	38,5	8	Doente
21	F	52	Uniformes	37,6	1	Saudável
22	F	72	Inexistentes	38,0	3	Doente
—	F	87	Espalhadas	39,0	6	Doente



Dados incompletos

- A ausência de valores em alguns atributos pode ter diferentes causas:
 - O atributo não foi considerado importante (ou não era obrigatório) quando os dados foram coletados;
 - Desconhecimento do valor do atributo no preenchimento dos valores do objeto;
 - Distração no preenchimento
 - Inexistência de valor para o atributo em alguns registros (quantidade de partos para o gênero masculino);
 - Problema com o equipamento utilizado na coleta;



Dados inconsistentes

- São dados que possuem valores conflitantes em seus atributos:
- Exemplo: Idade 3, peso 150; Volta de 5 s em um circuito de Formula 1, com 3,5Km;
- Outro exemplo bastante comum é o uso de escalas diferentes para fazer referência a uma mesma medida (Metros e centímetros);
- Inconsistências também podem ser reconhecidos quando relações entre atributos são claramente conhecidas (valores correlacionados direta ou indiretamente).
- Algoritmos simples podem verificar existência de inconsistências, em caso de conjuntos de dados não muito grandes, dados inconsistentes podem ser removidos manualmente.



Lidando com dados duplicados

- Para lidar com dados duplicados, devemos sempre avaliar o impacto de remoção das duplicatas, **e remover sempre que possível.**
- Quando trabalhamos com Big Data, lidamos com alto volume de dados, redução de duplicatas e processamento de dados desnecessários significa mais rapidez para que seu código rode do início até o fim.
- Isso significa menos processamento exigido da sua máquina, e em caso de estar usando máquinas alugadas em nuvem, um menor custo para seu projeto de Data Science.

Dados redundantes

Tabela 3.5 Conjunto de dados com objetos redundantes

Idade	Sexo	Peso	Manchas	Temp.	# Int.	Diagnóstico
28	M	79	Concentradas	38,0	2	Doente
18	F	67	Inexistentes	39,5	4	Doente
49	M	92	Espalhadas	38,0	2	Saudável
18	F	67	Inexistentes	39,5	4	Doente
18	M	43	Inexistentes	38,5	8	Doente
21	F	52	Uniformes	37,6	1	Saudável
22	F	72	Inexistentes	38,0	3	Doente

Tabela 3.6 Conjunto de dados com atributos redundantes

Idade	Sexo	Peso	Manchas	Temp.	# Int.	# Vis.	Diagnóstico
28	M	79	Concentradas	38,0	2	2	Doente
18	F	67	Inexistentes	39,5	4	4	Doente
49	M	92	Espalhadas	38,0	2	2	Saudável
18	M	43	Inexistentes	38,5	8	8	Doente
21	F	52	Uniformes	37,6	1	1	Saudável
22	F	72	Inexistentes	38,0	3	3	Doente
19	F	87	Espalhadas	39,0	6	6	Doente



Dados redundantes

- Um objeto redundante é um objeto muito semelhante a outro no mesmo conjunto de dados.
- Também é considerado um atributo redundante quando ele pode ser deduzido a partir do valor de um ou mais atributos. Dois ou mais atributos estão correlacionados quando apresentam um perfil de variação semelhante para os diferentes objetos.
- Dados redundantes podem criar a falsa sensação de que o perfil de objeto é mais importante que os demais, induzindo o modelo de análise.
- É importante identificar e eliminar as redundâncias, que podem ser feitas pela eliminação dos objetos semelhantes ou pela combinação dos valores dos atributos dos objetos semelhantes.



Lidando com variáveis do tipo String

- No caso de variáveis do tipo string, não é necessário nenhum tipo de tratamento obrigatório porém é necessário analisar caso a caso para entender a real necessidade de padronizar estes dados.
- Como Python é uma linguagem de tipagem forte, variações de letras (maiúscula para minúscula ou vice versa), variações no número de espaços entre duas palavras, acentos e outras diferenças que podem parecer irrelevantes podem impactar na manipulação dos dados e nos resultados da sua análise
- Boas práticas na manipulação de dados string podem ser:
 - Evitar uso de espaços (ou removê-los caso existam)
 - Evitar uso de letras maiúsculas (ou transformar todo o texto para minúsculo)
 - Remoção de caracteres especiais



Padronização de variáveis

- A medida que suas análises passam a utilizar modelagem estatística ou machine learning, é importante se atentar a padronização das suas variáveis.
- Padronizar as variáveis significa trazê-las para uma escala comum para todas.
- Se temos modelos onde o valor da variável pode aumentar seu peso, uma variável com um número maior (como por exemplo salário) pode impactar mais no modelo que outra com número menor (como por exemplo idade).
- Para evitar este viés nos modelos, é importante trazer as variáveis para a mesma escala, e assim evitamos influências incorretas nos resultados.

Padronização de variáveis

➤ Exemplo:

