

# Fecomércio Sesc

Data Science – Princípios e Técnicas

Setembro

2024



Onde me encontrar:

https://www.linkedin.com/in/marco-mialaret-junior/

e

https://github.com/MatmJr





# Projeto de Ciência de dados



Em geral os projetos de data science envolvem as seguintes etapas:

- 1. Entendimento do negócio: aqui são definidas as perguntas, o objetivo da análise de dados e o plano a ser seguido;
- 2. Compreensão dos dados: etapa utilizada para coletar e explorar os dados, aumentando a compreensão sobre sua estrutura, atributos e contexto;



- 3. **Preparação dos dados:** após a análise exploratória, inicia-se o processo de limpeza, filtragem, estruturação, redução e integração dos dados;
- 4. **Modelagem dos dados:** envolve as tarefas de seleção dos dados, definição e construção do modelo;



- 5. **Validação do modelo:** os resultados gerados pelo modelo são avaliados, para verificar se a precisão obtida está satisfatória e coesa;
- 6. **Utilização do modelo:** após serem validados, os resultados dos modelos são utilizados e monitorados.



# Também vimos que: Os dados utilizados estão normalmente "sujos"

Muitas vezes, as bases contêm dados incompletos, inconsistentes, corrompidos, duplicados ou em formatos inadequados, entre outros problemas. Portanto, é essencial a intervenção de um profissional capacitado para tratar esses dados antes de iniciar a análise propriamente dita.



A crescente geração de dados amplifica a necessidade de limpeza de dados eficiente, visto que a má qualidade dos dados implica custos significativos para as empresas, chegando a US\$ 12,9 milhões anualmente, conforme indicações da Gartner.



Além do impacto financeiro, dados imprecisos demandam dos cientistas de dados até 60% do seu tempo em tarefas de limpeza e organização, ressaltando a importância de adotar ferramentas e estratégias proativas para assegurar a qualidade dos dados desde sua coleta.



Mas como resolvemos esse problema?

Com o processo de limpeza de dados!



#### O que é Limpeza de dados?

A limpeza de dados, essencial no gerenciamento de qualidade, envolve identificar e corrigir erros ou inconsistências nos dados para garantir sua precisão, consistência e utilidade.



Esse processo é crucial pois o uso de dados brutos, sem tratamento, pode levar a decisões baseadas em informações imprecisas, prejudicando a eficácia operacional e estratégica das organizações.



Mas por que não podemos usar dados brutos em vez de gastar tanto tempo na limpeza de dados?



- Entradas com erros ortográficos: Erros de digitação e ortografia podem levar a erros de categorização.
- Formatos inconsistentes: Datas, números ou categorias podem ser representados de forma diferente no mesmo conjunto de dados.
- Valores discrepantes e erros: Entradas incomuns ou erradas podem levar a análises imprecisas.

- Registros duplicados: Dados redundantes podem levar a estatísticas e conclusões imprecisas.
- Valores nulos ou ausentes: Dados incompletos podem levar a lacunas na análise e a insights imprecisos e/ou limitados.
- Dados imprecisos: Informações incorretas ou desatualizadas podem levar a decisões imprecisas.



- Unidades não padronizadas: Diferentes unidades de medida podem criar problemas de inconsistência de dados, especialmente ao comparar ou agregar dados.
- Dados incompatíveis: Dados conflitantes de fontes diferentes podem causar discrepâncias integração de dados e análise.



#### Principais técnicas de limpeza de dados

#### Removendo duplicatas:

Como fazer: Utilize algoritmos para identificar e remover linhas duplicadas com base em atributos vitais selecionados.



#### Tratamento de dados ausentes:

Como fazer: As opções incluem imputação, exclusão ou uso de algoritmos que podem lidar com valores ausentes. A imputação pode usar estratégias baseadas em média, mediana ou modelo, como k-NN.



#### **Corrigindo dados incorretos:**

Como fazer: verificações de consistência e revisão manual, se necessário. <u>Ferramentas de preparação de dados</u> podem ajudar na correspondência de padrões e correções.



#### Tratamento de valores discrepantes:

Como fazer: identifique valores discrepantes por meio de métodos estatísticos, como pontuação Z ou IQR, e decida se deseja limitá-los, transformá-los ou removê-los.



#### **Normalizando Dados:**

Como fazer: aplique técnicas como dimensionamento mínimo-máximo, normalização de pontuação Z ou transformações de log.



#### Validando a consistência dos dados:

Como fazer: Crie regras de validação para verificar relacionamentos e consistência entre atributos.



#### **Transformando Dados:**

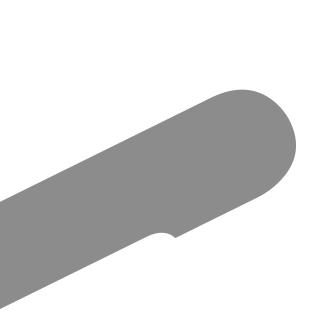
Como fazer: usar <u>transformações de dados</u> como codificação de dados categóricos ou criação de termos de interação com base em necessidades analíticas.



#### Mais detalhes:

https://www.astera.com/pt/type/blog/data-cleansing/





# Estudo de caso



Uma empresa de e-commerce especializada em produtos eletrônicos busca extrair insights de seu banco de dados de clientes. Os produtos, armazenados na sede, são entregues globalmente via navio, avião ou caminhão, com descontos baseados no peso do produto. Os clientes têm acesso ao suporte para dúvidas ou problemas e podem avaliar a experiência de compra. A única informação pessoal disponível é o gênero do cliente.

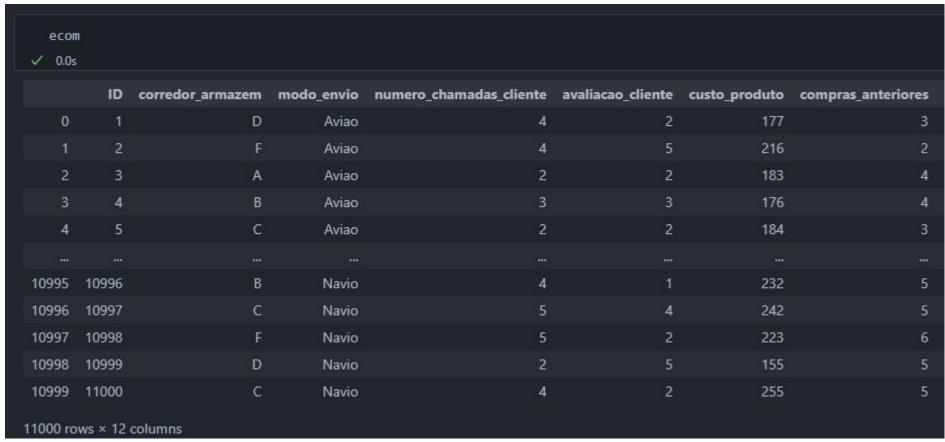


O objetivo do estudo de caso é explorar e organizar os dados, identificar problemas e analisá-los sob a perspectiva da entrega do produto.



Nome da Coluna	Tipo de Dado	Descrição  ID exclusivo de cada transação.  Letra do corredor do armazém onde o produto está armazenado.		
ID	Int			
corredor_armazem	Character(1)			
modo_envio	String	Modo de envio do produto.		
numero_chamadas_cliente	Int	Número de vezes que o cliente acionou o suporte da empresa.		
avaliacao_cliente	Int	Avaliação do cliente sobre a experiência de compra.  Custo do produto.  Número de vezes que o cliente fez uma compra na empresa.  Prioridade de entrega do produto.  Gênero do cliente (F ou M).		
custo_produto	Int			
compras_anteriores	Int			
prioridade_produto	String			
genero	Character(1)			
desconto	Int	Desconto concedido na compra do produto.		
peso_gramas	Int	Peso do produto.  Se o produto foi entregue no prazo, sendo 0 (não foi entregue no prazo) ou 1 (foi entregue no prazo).		
entregue_no_prazo	Character(1)			







#### Verificar as informações do conjunto de dados

```
ecom.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11000 entries, 0 to 10999
Data columns (total 12 columns):
    Column
                            Non-Null Count Dtype
                            11000 non-null int64
    ID
                     11000 non-null object
    corredor armazem
    modo_envio
                            11000 non-null object
    numero_chamadas_cliente 11000 non-null int64
    avaliacao cliente
                            11000 non-null int64
    custo_produto
                            11000 non-null int64
                            11000 non-null int64
    compras anteriores
    prioridade produto
                            11000 non-null object
                            11000 non-null object
    genero
    desconto
                            11000 non-null int64
                            11000 non-null int64
    peso gramas
11 entregue no prazo
                            11000 non-null int64
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```



Separação de Variáveis numéricas das variáveis categóricas:



Para as variáveis numéricas podemos usar o método describe para ver um resumo dos dados.

ecom[nums].describe()								
	numero_chamadas_cliente	custo_produto	compras_anteriores	desconto	peso_gramas	entregue_no_prazo		
count	11000.000000	11000.000000	11000.000000	11000.000000	11000.000000	11000.000000		
mean	4.054455	210.200909	3.567727	13.372545	3633.844455	0.596636		
std	1.141438	48.062985	1.522852	16.204943	1635.402722	0.490595		
min	2.000000	96.000000	2.000000	1.000000	1001.000000	0.000000		
25%	3.000000	169.000000	3.000000	4.000000	1839.000000	0.000000		
50%	4.000000	214.000000	3.000000	7.000000	4149.000000	1.000000		
75%	5.000000	251.000000	4.000000	10.000000	5050.000000	1.000000		
max	7.000000	310.000000	10.000000	65.000000	7846.000000	1.000000		



Para as variáveis categóricas isso não faz muito sentido, então devemos procurar outras formas de descrever o conjunto de dados.

```
for coluna in categ:
    print(f"Contagem de valores para {coluna}:")
    print(ecom[coluna].value_counts())
    print("\n")
```



Obtendo informações do nosso target

```
ecom[target].value_counts()
```



#### Vamos começar checando os Null e NaN

```
ecom.isnull().sum()
```

```
ecom.isna().sum()
```



Não existem valores ausentes e 'Not a Number' no conjunto de dados, mas será que nossos problemas estão resolvidos ?

Também podemos checar registros duplicados.

```
ecom.duplicated().sum()
```



Outliers são pontos de dados que se distanciam significativamente dos demais em um conjunto, representando valores incomuns. Esses valores atípicos podem afetar análises estatísticas, possivelmente ocultando descobertas importantes ou distorcendo resultados. A identificação de outliers não segue regras estatísticas rígidas, dependendo mais do conhecimento específico da área e da análise do processo de coleta de dados.

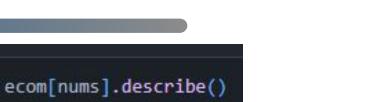


Embora não haja uma definição matemática precisa, métodos e testes estatísticos ajudam a detectar esses pontos discrepantes, que são cruciais por seu potencial de impactar análises estatísticas adversamente.

Vamos começar a análise olhando outra vez para as variáveis numéricas



## Dicionário de dados:



	numero_chamadas_cliente	custo_produto	compras_anteriores	desconto	peso_gramas	entregue_no_prazo
count	11000.000000	11000.000000	11000.000000	11000.000000	11000.000000	11000.000000
mean	4.054455	210.200909	3.567727	13.372545	3633.844455	0.596636
std	1.141438	48.062985	1.522852	16.204943	1635.402722	0.490595
min	2.000000	96.000000	2.000000	1.000000	1001.000000	0.000000
25%	3.000000	169.000000	3.000000	4.000000	1839.000000	0.000000
50%	4.000000	214.000000	3.000000	7.000000	4149.000000	1.000000
75%	5.000000	251.000000	4.000000	10.000000	5050.000000	1.000000
max	7.000000	310.000000	10.000000	65.000000	7846.000000	1.000000



O atributo desconto está com características interessantes:

- a média(mean): 13.372545
- o desvio-padrão(std): 16.204943
- o menor desconto (min): 1.000000
- o maior desconto (max): 65.00000



Além disso 75% dos produtos estão com desconto menor do que 10%, ou seja, a maioria dos produtos está com desconto de até 10%, assim podemos concluir que descontos muito maiores que 10% podem sem cutliars

podem sem outliers.

ecom.desconto.describe()					
count	11000.000000				
mean	13.372545				
std	16.204943				
min	1.000000				
25%	4.000000				
50%	7.000000				
75%	10.000000				
max	65.000000				
Name: d	esconto, dtype: float64				



Para aprimorar a análise de outliers, uma técnica é calcular o limite em que observações podem ser consideradas atípicas usando a fórmula:

Limite = média ± 3\*desvio padrão.



Esse método baseia-se na regra empírica, \*\*presumindo que os dados seguem uma distribuição normal\*\*, onde aproximadamente 99,7% dos dados encontram-se dentro de três desvios padrão da média. Observações fora desse limite são potencialmente outliers, sugerindo variações atípicas que podem necessitar de investigação adicional para determinar suas causas ou validade.





#### Determinando os Outliers:

<pre>outliers = ecom[(ecom.desconto &lt;= limiteInferior)   (ecom.desconto &gt;= limiteSuperior)] outliers.head() </pre> <pre>0.0s</pre>								
	ID	corredor_armazem	modo_envio	numero_chamadas_cliente	avaliacao_cliente	custo_produto		
36	37	D	Navio	3	5	137		
60	61	D	Navio	3	1	221		
62	63	А	Navio	5	1	105		
111	112	В	Caminhao	4	2	239		
122	123	A	Caminhao	4	2	160		



```
print("O tamanho do dataset original: ", len(ecom))
  print("Os outliers do atributo desconto: ", len(outliers))
  ✓ 0.0s

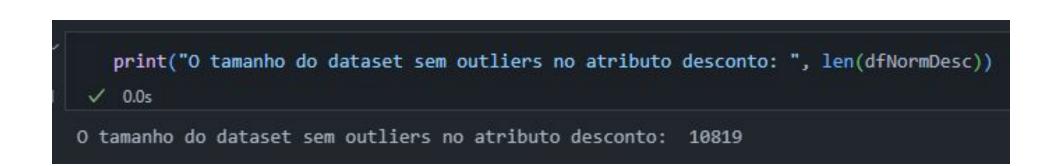
O tamanho do dataset original: 11000
Os outliers do atributo desconto: 181
```



Podemos determinar o conjunto dos valores sem outliers com a negação da operação acima:

4	<pre>dfNormDesc = ecom[(ecom.desconto &gt; limiteInferior) &amp; (ecom.desconto &lt; limiteSuperior)]   dfNormDesc.head()</pre>							
	ID	corredor_armazem	modo_envio	numero_chamadas_cliente	avaliacao_cliente	custo_produto		
0	1	D	Aviao	4	2	177		
1	2	F	Aviao	4	5	216		
2	3	A	Aviao	2	2	183		
3	4	В	Aviao	3	3	176		
4	5	С	Aviao	2	2	184		







Agora podemos olhar para o describe da variável desconto

```
dfNormDesc.desconto.describe()

√ 0.0s

         10819.000000
count
            12.536186
mean
std
            14.981972
min
            1.000000
25%
             4.000000
50%
            7.000000
75%
            10.000000
            61.000000
max
Name: desconto, dtype: float64
```



Exercício: Usando outras ferramentas para resolver o mesmo problema:

Vamos usar a biblioteca SciPy para resolver esse problema



```
from scipy.stats import zscore

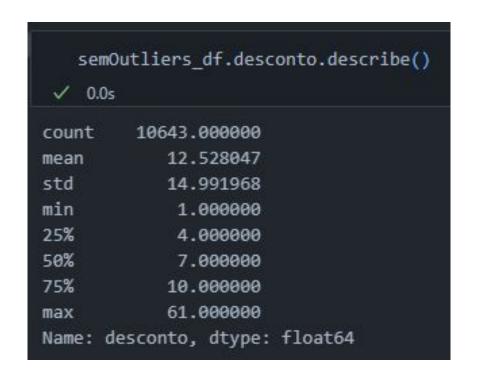
zscore_df = dfNormDesc[nums].apply(zscore)

semOutliers_df = dfNormDesc[(zscore_df.abs()<3).all(axis=1)]

</pre>

0.0s
```







# **Dúvidas?**









#### **Marco Mialaret, MSc**

**Telefone:** 

81 98160 7018

E-mail:

marcomialaret@gmail.com

