



apresenta



Data Science & Machine Learning



Data Wrangling

Consultor: Carolina Bez

Olá, eu sou a Carol

- Graduação: Engenharia Eletrônica e de Computação na UFRJ (diploma *Magna Cum Laude*)
- Extensão em Data Science em Harvard Extension School (em andamento)
- Consultora de Data Science na IBM há mais de 3 anos, atuando como *Tech Lead* há 1 ano
- Principais Projetos: Text and Speech Analytics, Propensão de Venda, *Credit Scoring*, Detecção de Fraude

O que veremos neste módulo

1. Intro *Data Wrangling*
2. Descoberta dos dados
 - a. Recapitulação de modelos
 - b. Análise dos dados
3. Estruturação dos dados
 - a. *Pivot table* e *one-hot encoding*
4. Limpeza dos dados
 - a. Seleção e filtragem
 - b. Tratamento de nulos e *outliers*
 - c. Tratamento de ruídos
5. Enriquecimento dos dados
 - a. Tratamento de textos
6. Validação dos dados
7. Produtização
 - a. Estruturação do *pipeline*
 - b. Produtização do *pipeline*

Alinhamento de Expectativas

O que o aluno vai sair aprendendo?

Aprender a limpar, normalizar, combinar, estruturar e organizar os diversos tipos de dados a fim de poder realizar análises e modelagens corretas, confiáveis e conclusivas



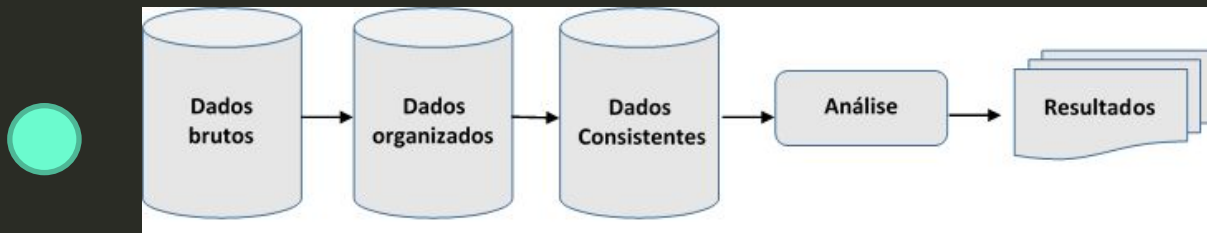
Intro *Data Wrangling*

Consultor: Carolina Bez

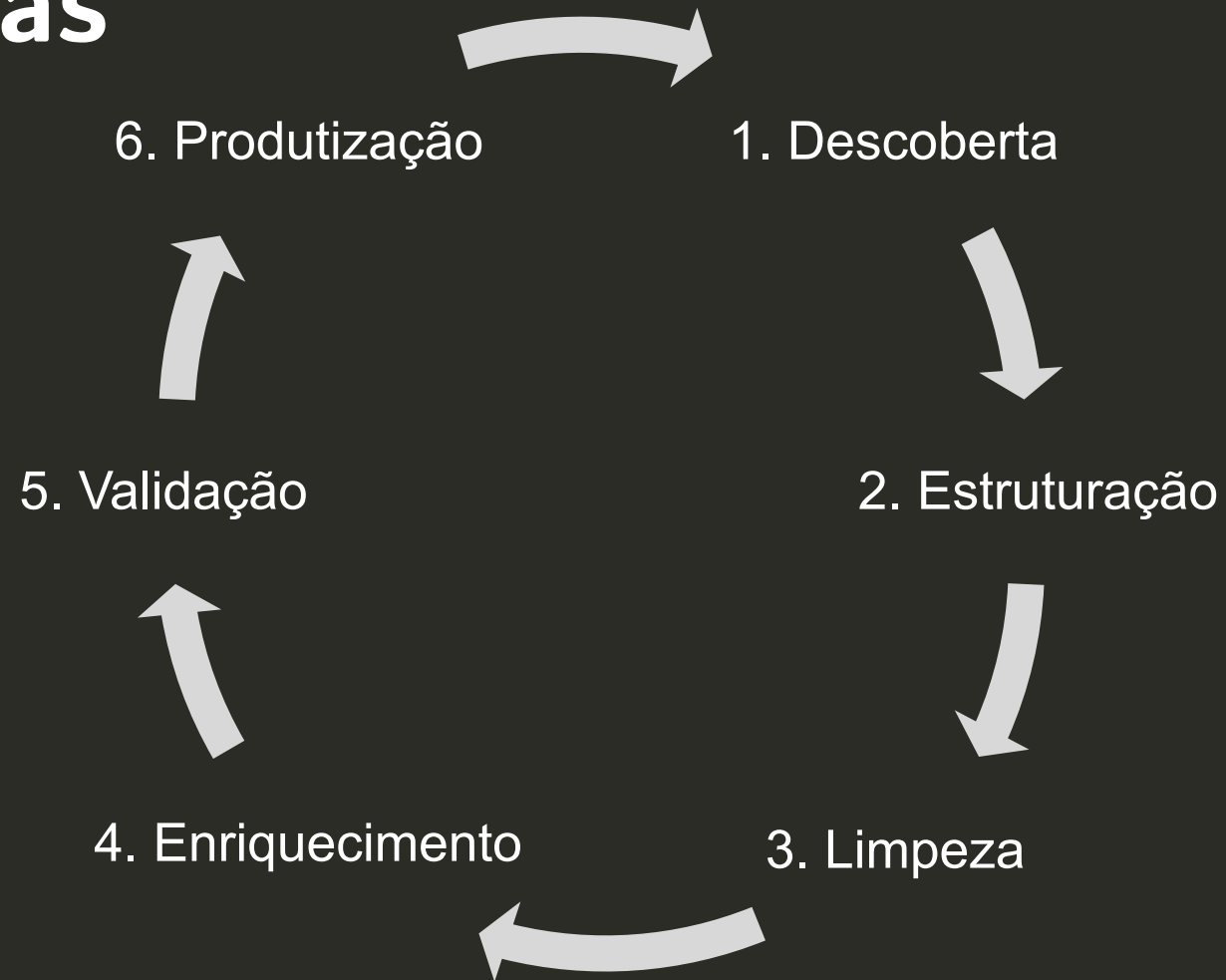
Abril de 2021

O que é *Data Wrangling*?

- É o processo de transformar e mapear dados que estão em uma forma mais “bruta” para um formato de maior qualidade e usabilidade para futuras análises
- Pré-requisito para o processo de visualização de dados, agregação, modelagem estatística e machine learning
- Data Wrangling consome cerca de 80% do tempo dos cientistas de dados, sobrando apenas 20% para exploração e modelagem



Etapas



Descoberta

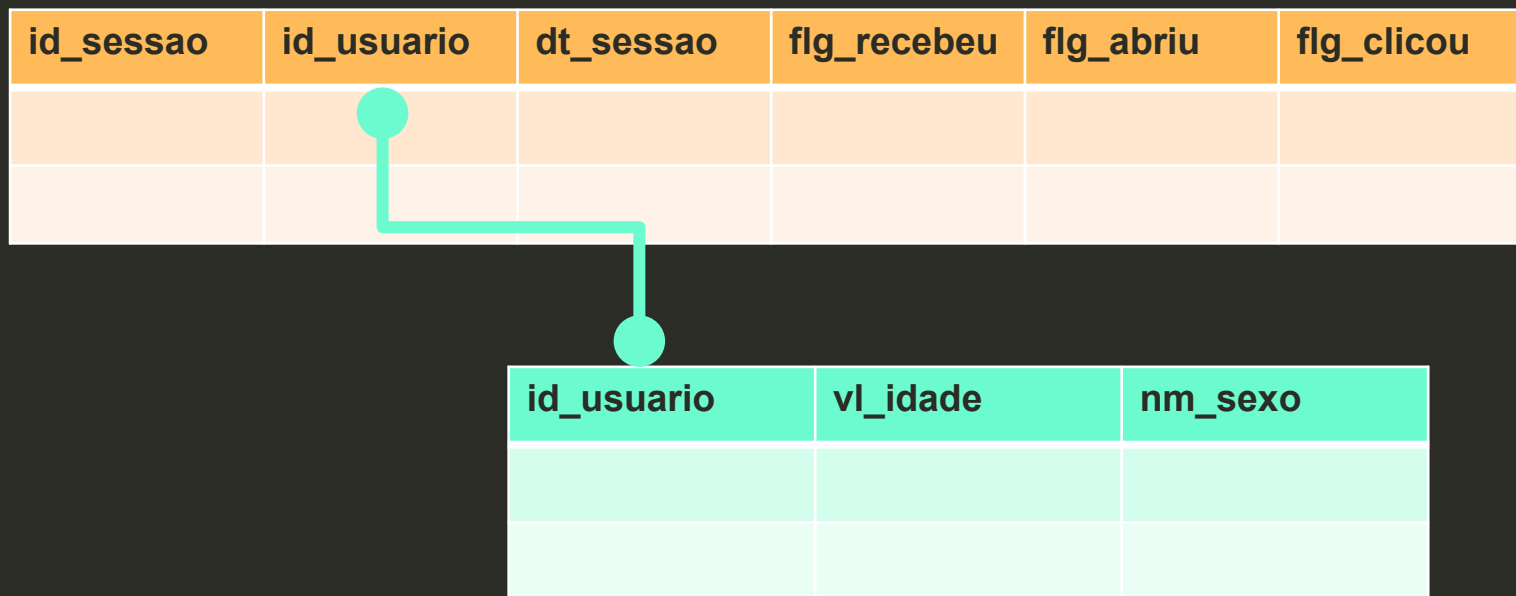
- Entendimento do problema de negócio
- Desenho da modelagem
- Mapeamento da necessidade de dados
- Entendimento das dimensões, chaves e relacionamentos

Estruturação

Estruturar os dados de acordo com a modelagem definida:

- Merges
- Agrupamentos
- Pivot Tables
- One Hot Encoding

Estruturação



Estruturação

ID	DT_CONSUMO	VL_CONSUMO
XX1	2020-11-03	50
XX1	2020-11-15	21
XX2	2020-11-18	33
XX2	2020-12-16	23
XX3	2020-11-06	45
XX4	2020-12-18	46
XX4	2020-11-18	12
XX4	2020-11-04	34



ID	TOTAL_CONSUMO_ANTES	DT_COMPRA	COMPROU (MES REF: DEZ)
XX1	71	2020-12-01	1
XX2	33	2020-12-15	1
XX3	45		0
XX4	46		0

Limpeza

- O famoso “*Data Cleaning*”
- Seleção dos dados relevantes e filtros necessários para evitar *bias*
- Tratamento de nulos, erros e *outliers*
- Padronização de valores
- Tratamento de ruídos

Limpeza

	a	b	c	d
0	0.0	NaN	-1.0	1.0
1	NaN	2.0	NaN	NaN
2	2.0	3.0	NaN	9.0
3	NaN	4.0	-4.0	16.0

Enriquecimento

- “*Feature Engineering*”: derivar novos dados a partir dos que já existem, podendo ser feito de diversas formas:
 - combinar duas ou mais colunas
 - realizar um cálculo a partir de uma coluna
 - criar uma dimensão para extrair métricas agregadas
- Necessário principalmente quando os dados não estão em formato estruturado. Ex: textos
- Entendimento das dimensões, chaves e relacionamentos

Validação

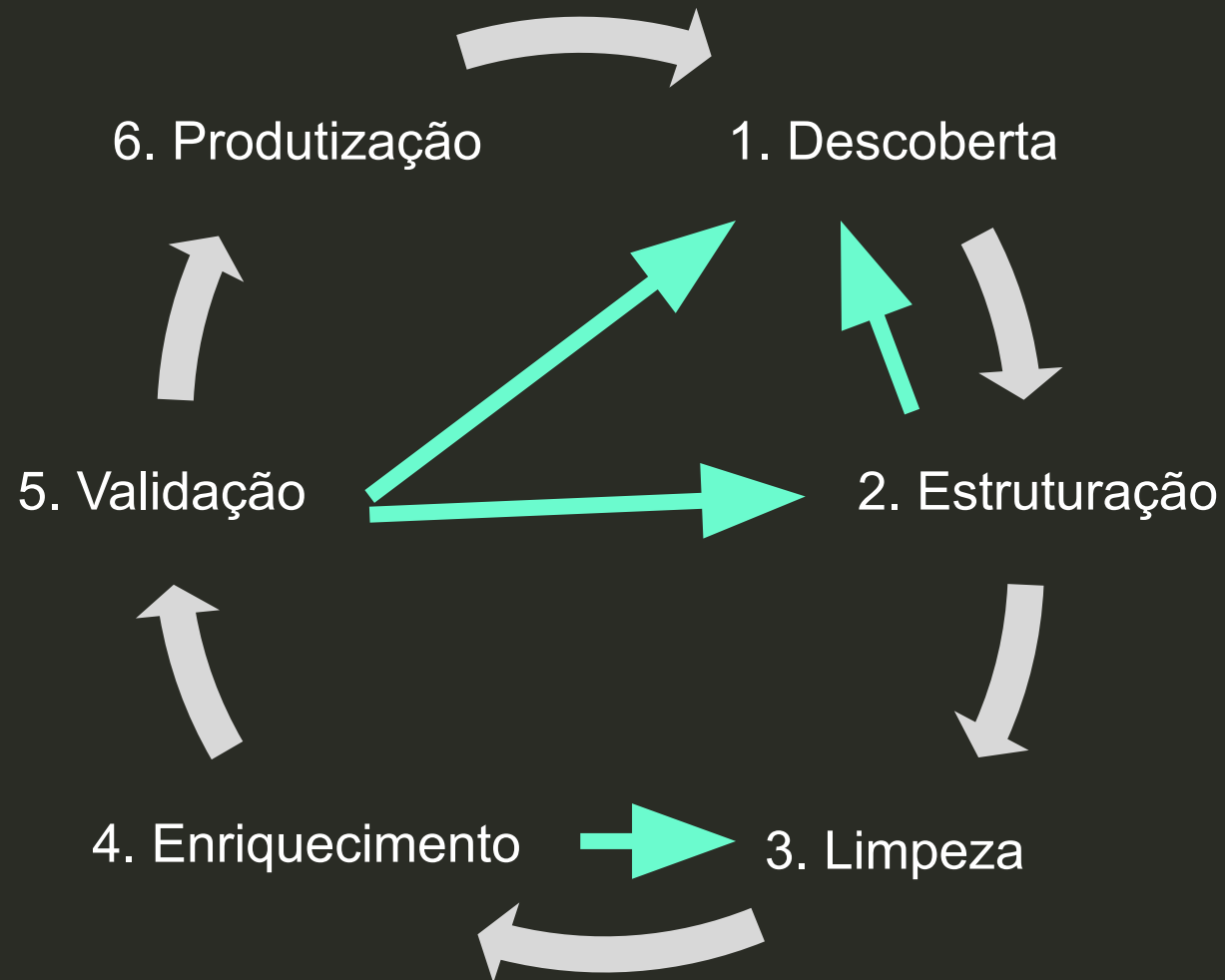
É preciso garantir que os dados apresentam **consistência**, **qualidade** e **segurança** após os tratamentos. Isso pode ser feito avaliando a:

- Distribuição dos dados
- Correlação (com o *target* e entre variáveis)

Produtização

- É bem provável que todo o tratamento feito para esses dados tenha de ser replicado para outros. Será que o fluxo montado para esse dataset funcionará para todos os outros?
- Além disso, em modelos preditivos, o dataset de teste sempre precisa de uma preparação um pouco diferente do de treino
- Por isso, é importante estruturar um pipeline que agregue todas as etapas realizadas e seja escalável para outros dados

É iterativo!



Descoberta dos dados

Consultor: Carolina Bez

O que veremos nessa aula:

1. Entendimento do problema de negócio
2. Desenho da modelagem e necessidade de dados
3. Entendimento das dimensões, chaves e relacionamentos

Entendimento do problema de negócio

- “Quero aumentar o engajamento das minhas campanhas de marketing digital”
- “Quero diminuir a quantidade de clientes inadimplentes”
- “Quero melhorar o atendimento ao meu cliente”

Entendimento do problema de negócio

Antes de partir para a solução, o primeiro passo é definir bem o problema.

	S	M	A	R	T
	SPECIFIC	MEASURABLE	ACTION-ORIENTED	RELEVANT	TIME-BOUND
5 W 2 H	WHAT, WHERE Especifique, limite o escopo	HOW MUCH / HOW MANY KPI's, como mensurar se alcançou?	HOW / WHO Está no seu alcance resolver esse problema? O que pode ser feito?	WHY É relevante para o negócio? É orientado a resultados?	WHEN Quanto tempo preciso para resolver?

Entendimento do problema de negócio

Exemplo: “Quero aumentar o engajamento das minhas campanhas de marketing digital”

S	Qual campanha? Em qual canal?	Venda do produto A para clientes novos no canal de email
M	Como medir engajamento? Qual KPI? Número de clicks no email? Número de compras finalizadas?	Taxa de clicks sobre views (Click to open)
A	O que posso fazer para melhorar? <i>Banners</i> melhores, <i>headers</i> mais chamativos, ofertas personalizadas?	<i>Banners</i> com ofertas personalizadas (até que ponto posso personalizar?)
R	Relevante para o negócio?	Sim, o produto A é chave para a empresa e o canal de email é um canal em potencial
T	Em quanto tempo?	As ofertas tem ciclo de vida de 6 meses e os banners demora 1 mês para ser feito e aprovado, então em no máximo 2 meses

Desenho da modelagem e necessidade de dados

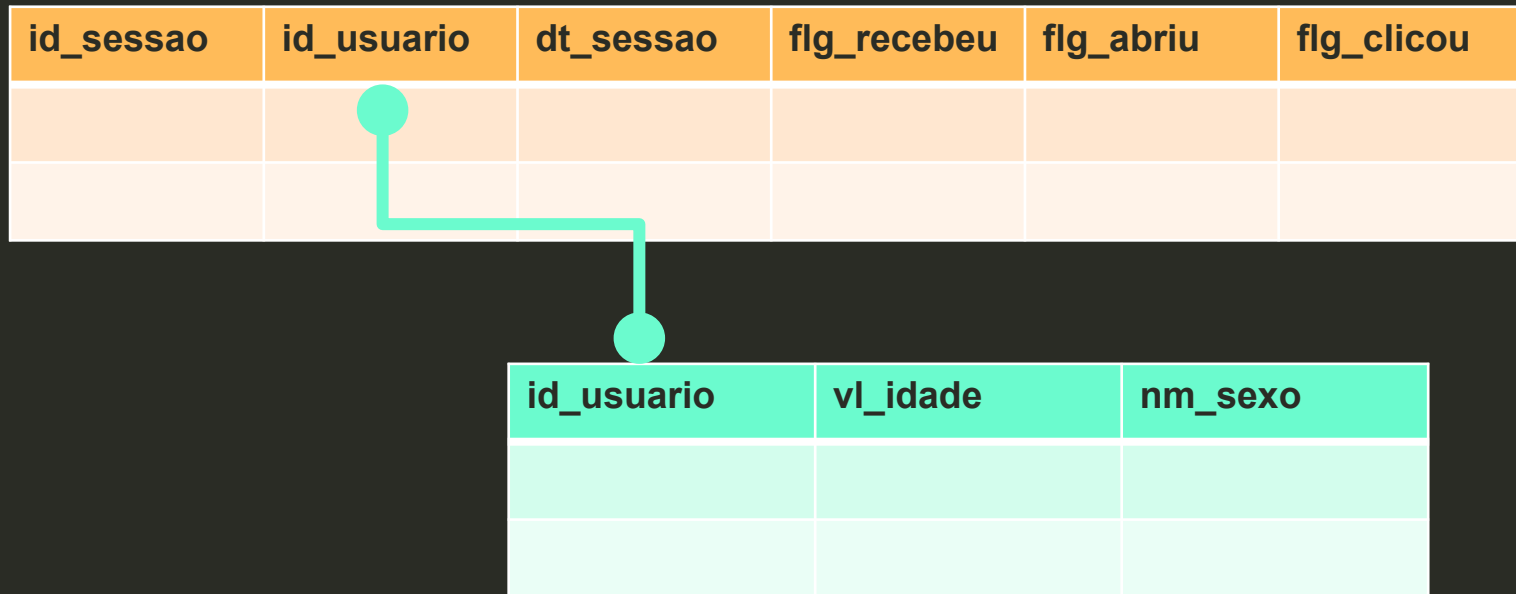
1. Reestruturar sua pergunta de forma mais “técnica”
2. Definir o tipo de modelo:
 - Aprendizado supervisionado (Classificação / Regressão): Qual *target*? Qual identificador do registro?
 - Aprendizado não-supervisionado (Clusterização / Redução de dimensionalidade?)
 - Aprendizado por reforço (Reinforcement Learning)
3. Mapear dados necessários:
 - Faça um brainstorming, pense em exemplos da vida real
 - Busque casos de uso semelhantes na academia e no mercado
 - Investigue dados existentes na empresa

Desenho da modelagem e necessidade de dados

Exemplo:

1. Definir o tipo de modelo:
 - Aprendizado supervisionado: prever quais clientes têm maior propensão a clicar
 - Aprendizado não-supervisionado: clusterizar clientes de acordo com seu perfil para o desenho de peças e ofertas mais assertivas
2. Mapear dados necessários:
 - Aprendizado supervisionado: prever quais clientes têm maior propensão a clicar
 - Aprendizado não-supervisionado: clusterizar clientes de acordo com seu perfil para o desenho de peças e ofertas mais assertivas

Entendimento das dimensões, chaves e relacionamentos



Recapitulação de modelos

Consultor: Carolina Bez

O que veremos nessa aula:

1. Tipos de Modelos
2. Algoritmo Supervisionado
3. Algoritmo Não-Supervisionado

Tipos de Modelos

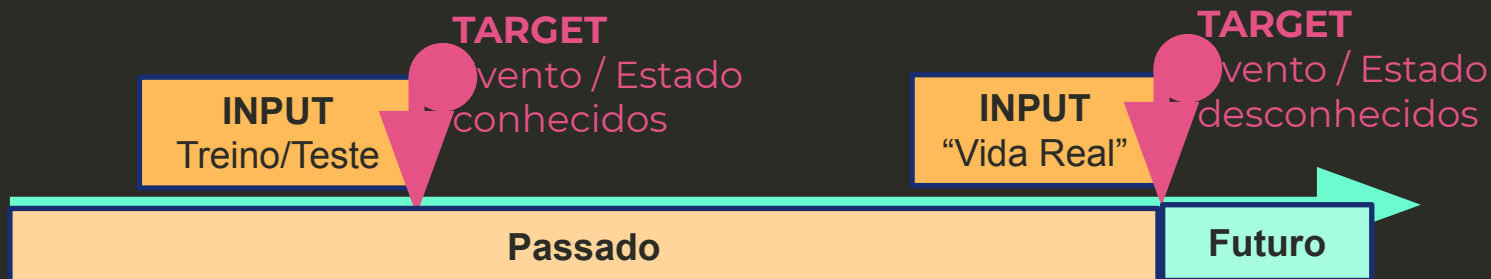
1. Aprendizado Supervisionado (*Supervised learning*)
2. Aprendizado Não-Supervisionado (*Unsupervised Learning*)
3. Aprendizado Semi-Supervisionado (*Semi-supervised Learning*)
4. Aprendizado Por Reforço (*Reinforcement Learning*)

Aprendizado Supervisionado



I1	I2	I3	I4	T
XXX	XXX	XXX	XXX	1
XXX	XXX	XXX	XXX	0
XXX	XXX	XXX	XXX	?
XXX	XXX	XXX	XXX	?

Modelam relacionamentos entre variável *label / target* e variáveis de *input* para que seja possível replicar o modelo para prever a *label* em dados de *label* desconhecida

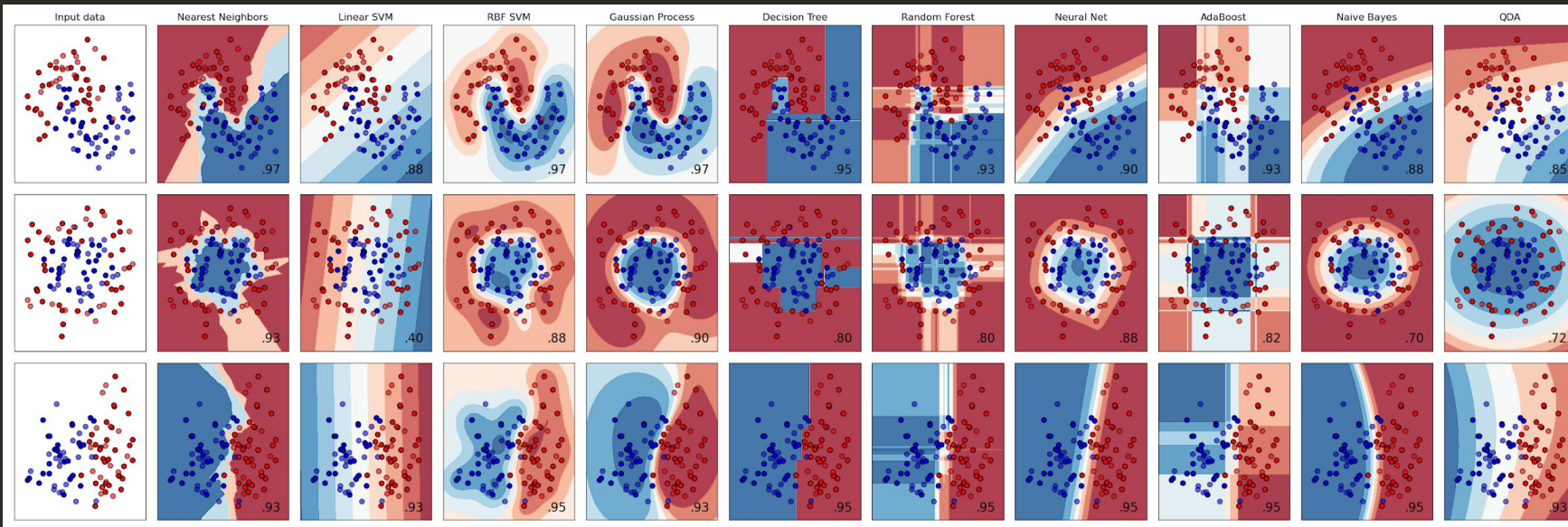


Aprendizado Supervisionado

Se valor associado ao evento / estado conhecidos (“label” / “target”) for:

- Flag / Categoria □ **Classificação**
Ex: Flag cliente clicou ou não, Categoria do produto do cliente
- Valor Numérico □ **Regressão**
Ex: Valor da quantidade de vendas no mês

Aprendizado Supervisionado



Aprendizado Não-Supervisionado



I1	I2	I3	I4
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX

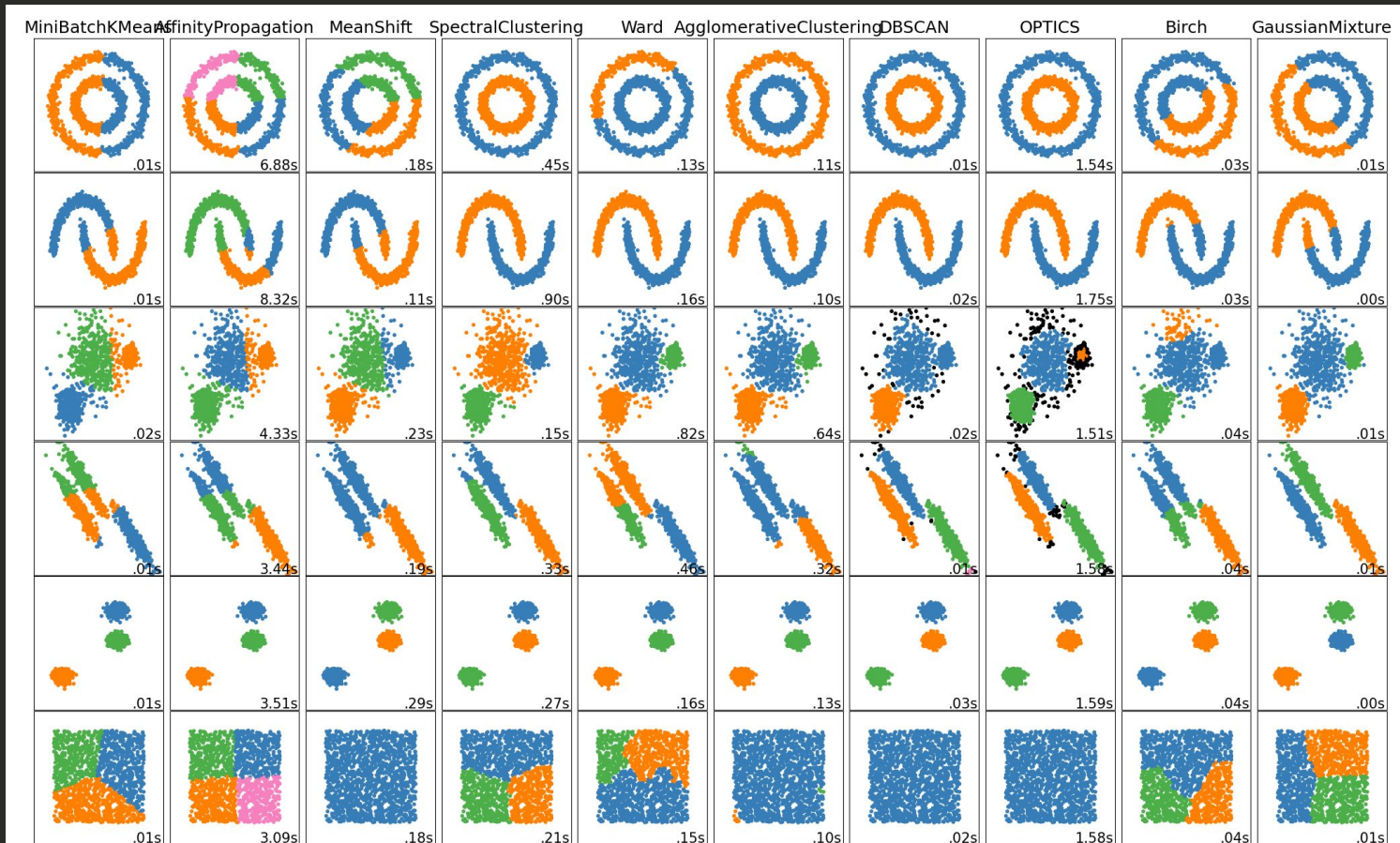
- Não possuem variável *label* / *target*
- Modelos tentam encontrar regras, detectar padrões, sumarizar e agrupar dados de modo a facilitar a descoberta de *insights* revelantes a partir dos dados

Aprendizado Não-Supervisionado

Se o objetivo for:

- Agrupar / Segmentar □ **Clusterização**
Ex: Perfilamento de clientes
- Encontrar anomalias □ **Detector de Anomalias**
Ex: Detectar falhas sistêmicas
- Reduzir a dimensão dos dados □ **Redução de dimensionalidade**
Ex: Plotar gráficos 2D em datasets de N variáveis
- Encontrar regras e padrões □ **Sistemas de recomendação, Association Rules**
Ex: Recomendação de produtos

Aprendizado Não-Supervisionado



Análise dos dados

Consultor: Carolina Bez

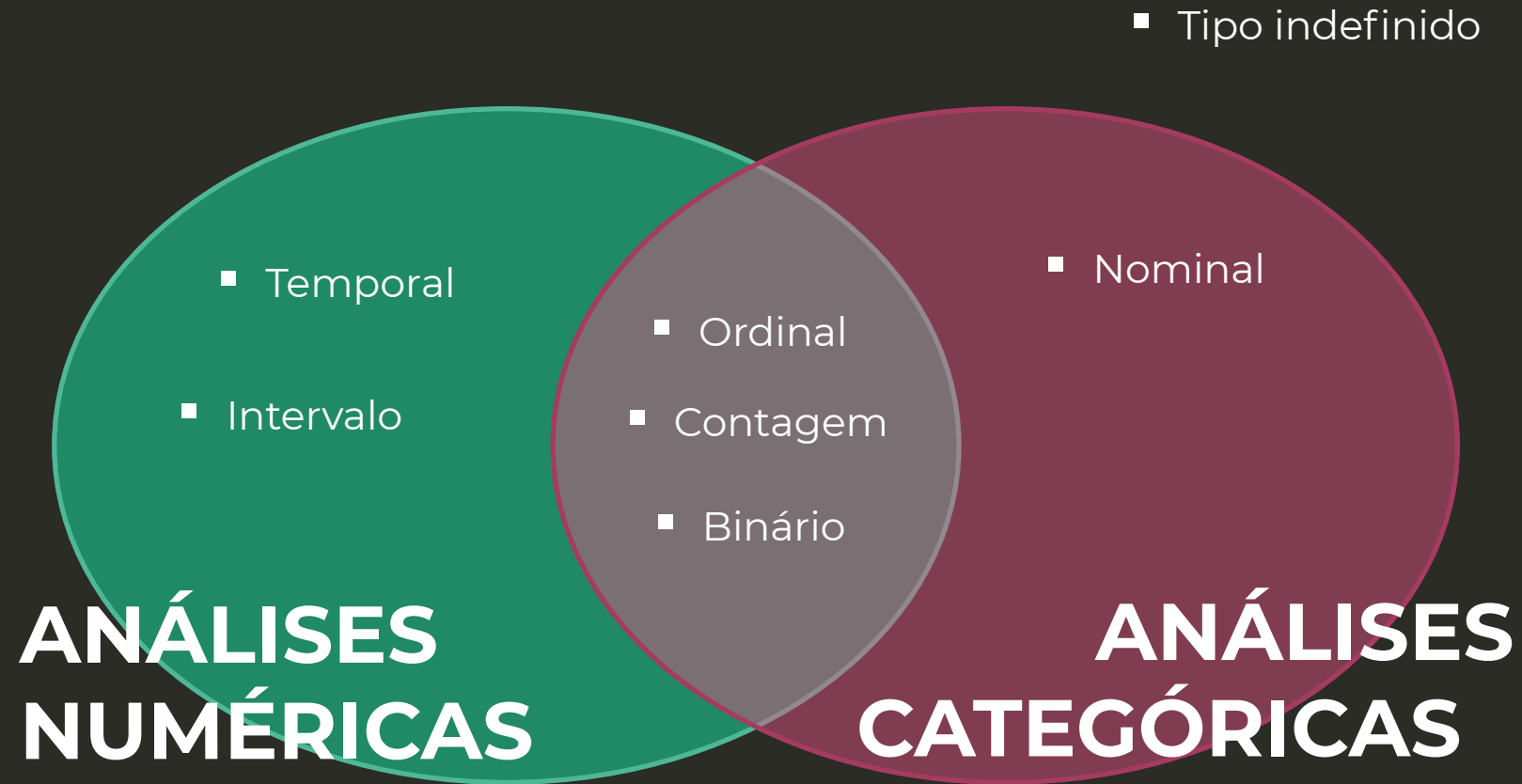
O que veremos nessa aula:

1. Tipos de Dados
2. Análise Univariada
3. Aula Prática (Colab)

Tipos de Dados

1. **Tipo indefinido** (ex: ID, CPF, número de série, etc.)
2. **Nominal**: sem relações numéricas (ex: Cachorro, Gato, Papagaio)
3. **Binário**: só podem existir 2 categorias: 0 ou 1 / True ou False / Valor A ou valor B
4. **Ordinal**: números inteiros, podem ser ordenados, mas a distância entre um e outro não é conhecida
5. **Contagem**: números inteiros positivos
6. **Temporal**: dados cíclicos e contínuos, podendo estar representados em dias, semanas, meses, anos, etc.
7. **Intervalos**: possui intervalos iguais entre os números e não expressam tempo (ex: percentuais, valores fracionados, etc.)

Tipos de Dados



Tipos de Dados

- Comando `pandas.DataFrame.dtypes`:

```
In [18]: df.dtypes
Out[18]: _id                object
         attendantCharacters  int64
         audio_length        float64
         callReason          object
         channel             object
         conversationId       object
         createdAt           datetime64[ns]
         dependant           bool
         feedback            int64
         initiatedAt         datetime64[ns]
         lastUpdatedAt        datetime64[ns]
```

- Porém, atenção! É sempre importante validar coluna por coluna pois ele pode errar em caso de anomalias no formato dos dados.
- Use `df.select_dtypes(include="float64").columns` para filtrar somente colunas de um tipo específico

Análise Univariada

Geral

- Percentual de nulos

Numéricas

- Média e Desvio Padrão
- Distribuição / Histograma
- Distribuição por percentis
- Verificação de outliers
- Check de Normalidade

Catóricas

- Distribuição por categoria
- Quantidade de categorias distintas

Percentual de nulos

- `pandas.DataFrame.isnull().sum(axis=0)`

```
In [62]: pd.DataFrame(df.isnull().sum(axis=0)).sort_values(by=0,ascending=False)/df.shape[0]*100
```

Out[62]:

	0
C69	100.0
C53	100.0
C16	100.0
C68	90.0
C288	2.0
C302	2.0
C315	2.0
C314	2.0
C313	2.0
C312	2.0
C311	2.0

- **Atenção!** Validar colunas com alto percentual de nulo!
- Há colunas não-nulas com nulos?

Método *Describe*

- Comando `pandas.DataFrame.describe`:

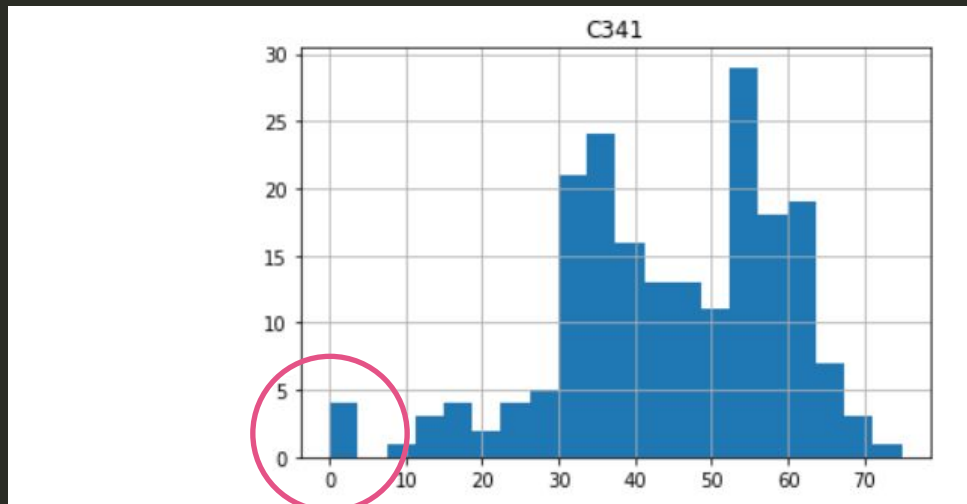
```
In [26]: df["silenceDuration"].describe(percentiles=[.001, .01, .1, .25, .5, .75, .9, .99, .999])
```

```
Out[26]: count      50.000000  
         mean       47.693700  
         std        35.573071  
         min         1.898000  
         0.1%        1.927351  
         1%          2.191510  
         10%         8.463100  
         25%        20.914750  
         50%        41.030500  
         75%        62.728750  
         90%       100.180700  
         99%       141.894660  
         99.9%     157.623366  
         max        159.371000  
         Name: silenceDuration, dtype: float64
```

- Útil para avaliar media, std, mín, max, percentis e outliers

Histograma

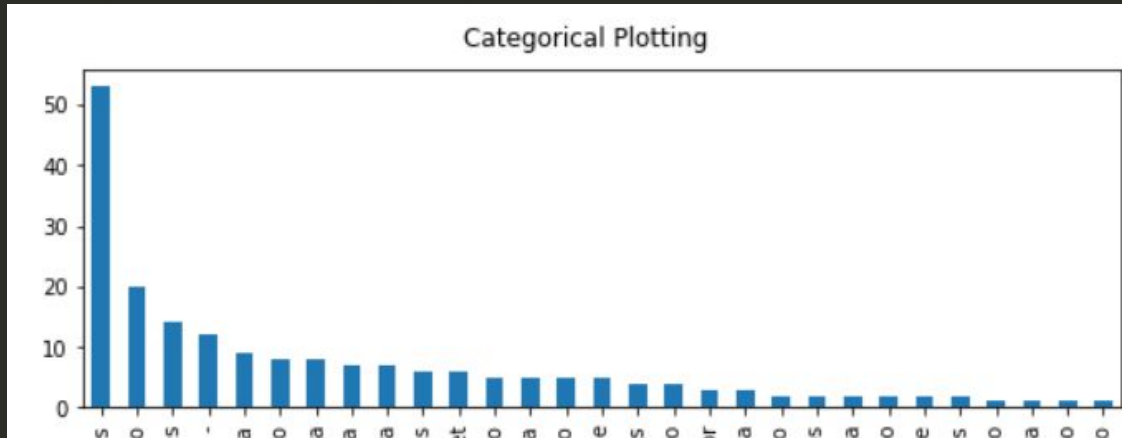
- `pandas.DataFrame.hist`



- Nesse caso, não parece seguir uma distribuição padrão, possui dois “picos”
- Pico isolado no zero: vale investigar!

Distribuição

- `pandas.DataFrame.value_counts(dropna=False).plot(kind='bar')`



- Há alguma categoria pouco representativa?
- A quantidade de categorias está aceitável?

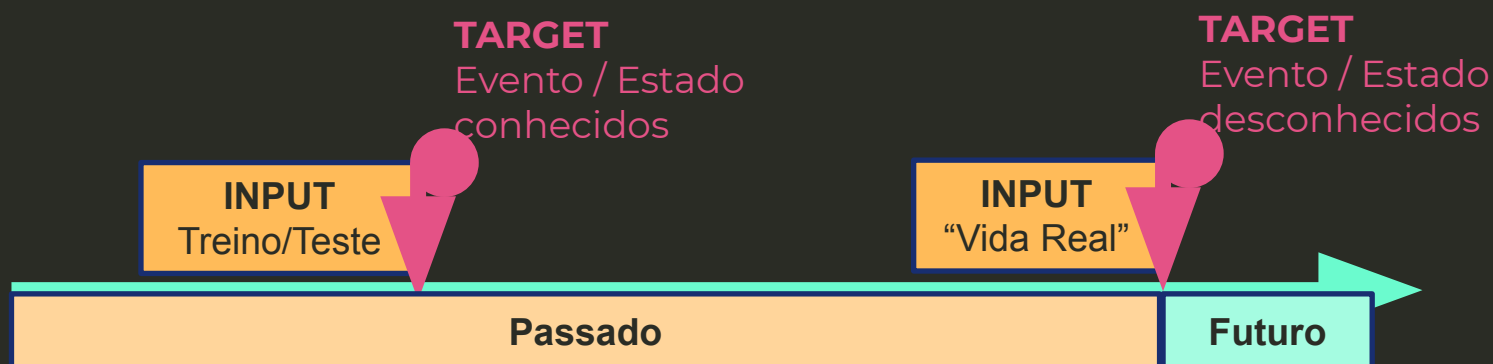
Estruturação dos dados

Consultor: Carolina Bez

O que veremos nessa aula:

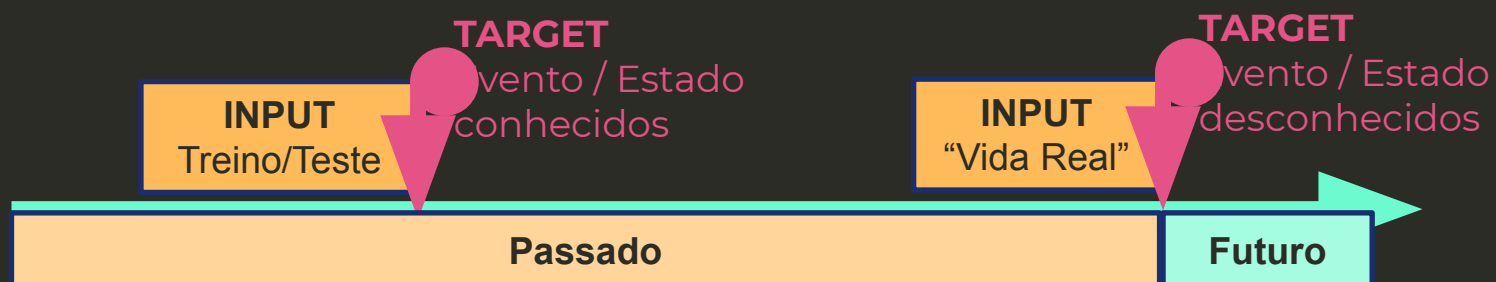
1. Estruturação para classificação de evento no tempo
2. Merges
3. Group By
4. Group by e Aggregate

Estruturação para classificação de evento no tempo



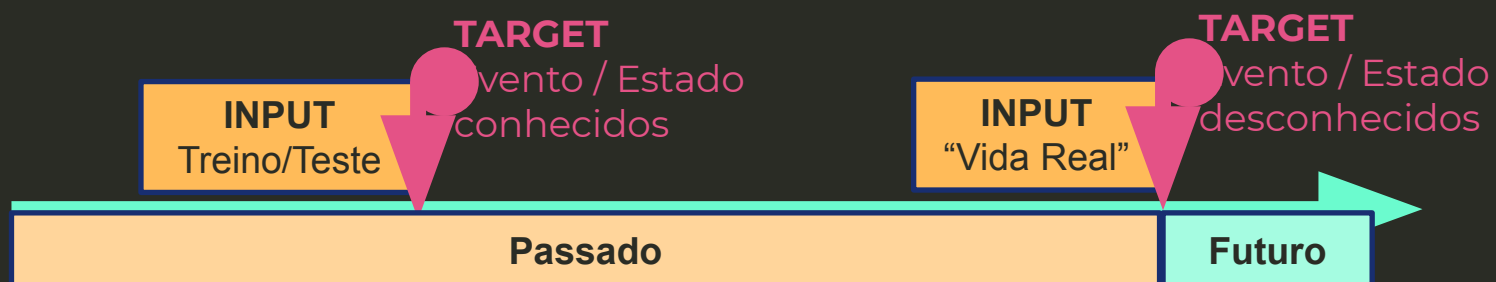
ID	DT_CONSUMO	VL_CONSUMO
XX1	2020-11-03	50
XX1	2020-11-15	21
XX2	2020-11-18	33
XX2	2020-12-16	23
XX3	2020-11-06	45
XX4	2020-12-18	46
XX4	2020-11-18	12
XX4	2020-11-04	34

Select



ID	DT_CONSUMO	VL_CONSUMO
XX1	2020-11-03	50
XX1	2020-11-15	21
XX2	2020-11-18	33
XX2	2020-12-16	23
XX3	2020-11-06	45
XX4	2020-12-18	46
XX4	2020-11-18	12
XX4	2020-11-04	34

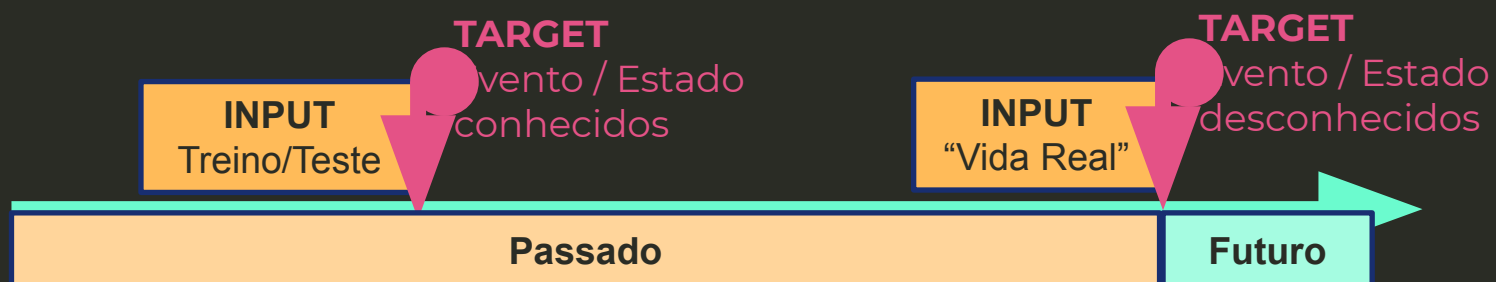
Group By e Aggregate



ID	DT_CONSUMO	VL_CONSUMO
XX1	2020-11-03	50
XX1	2020-11-15	21
XX2	2020-11-18	33
XX2	2020-12-16	23
XX3	2020-11-06	45
XX4	2020-12-18	46
XX4	2020-11-18	12
XX4	2020-11-04	34

ID	TOTAL_CONSUMO_ANTES
XX1	71
XX2	33
XX3	45
XX4	46

Merge



ID	TOTAL_CONSUMO_ANTES
XX1	71
XX2	33
XX3	45
XX4	46

ID	DT_COMPRA	COMPROU (MES REF: DEZ)
XX1	2020-12-01	1
XX2	2020-12-15	1
XX3		0
XX4		0

Merge



ID	DT_CONSUMO	VL_CONSUMO
XX1	2020-11-03	50
XX1	2020-11-15	21
XX2	2020-11-18	33
XX2	2020-12-16	23
XX3	2020-11-06	45
XX4	2020-12-18	46
XX4	2020-11-18	12
XX4	2020-11-04	34

ID	TOTAL_CONSUMO_ANTES	DT_COMPRA	COMPROU (MES REF: DEZ)
XX1	71	2020-12-01	1
XX2	33	2020-12-15	1
XX3	45		0
XX4	46		0

Pivot table e one-hot encoding

Consultor: Carolina Bez
Abril de 2021

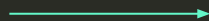
O que veremos nessa aula:

1. Pivot table
2. One-hot Encoding
3. Cat columns

Pivot Table

- `pandas.DataFrame.pivot_table`
- **Caso 1:** coluna categórica em que cada categoria vira uma nova coluna preenchida com o valor de outra variável

	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9



		C	large	small
	A	B		
bar	one		4.0	5.0
	two		7.0	6.0
foo	one		4.0	1.0
	two		NaN	6.0

Pivot Table

- **Caso 2:** sem transposição de coluna categórica, valores são agrupados pelo index

	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9

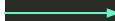


		D		E	
		sum	max	mean	min
A	B				
bar	one	9	8.0	7.000000	6.0
	two	13	9.0	9.000000	9.0
foo	one	5	5.0	3.666667	2.0
	two	6	6.0	5.500000	5.0

One-Hot Encoding

- `pandas.get_dummies`
- **Caso 1:** cada categoria vira uma nova variável booleana


	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3



	idade	animal_cat	animal_dog	sexo_femea	sexo_macho
0	1	0	1	0	1
1	2	1	0	1	0
2	3	0	1	1	0

One-Hot Encoding

- `pandas.get_dummies(..., drop_first=True)`
- **Caso 2:** cada categoria vira uma nova variável booleana exceto a variável de 1o nível



	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3

	idade	animal_dog	sexo_macho
0	1	1	1
1	2	0	0
2	3	1	0

Cat Codes

- `pandas.Series.cat.codes`
- Mapeia cada categoria em um valor numérico
- Útil para algoritmos que aceitam variáveis categóricas (ex: LightGBM)

	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3



	animal	sexo	idade
0	1	1	1
1	0	0	2
2	1	0	3

Seleção e filtragem de dados

Consultor: Carolina Bez
Abril de 2021

O que veremos nessa aula:

1. Filtros de tempo
2. Filtros de segmentos
3. Seleção de colunas: Qualidade
4. Seleção de colunas: Relevância

Filtros de tempo

1. Avaliar período histórico a ser analisado para a base principal
Ex: Previsão de fraude: coletar dados referentes aos últimos 3 anos de fraude na empresa
2. Avaliar período histórico a ser coletado para cálculo das variáveis
Ex: quero criar a variável qt_compras para saber quantas compras o cliente realizou nos últimos 6 meses

Atenção! Sempre avaliar junto à área de negócio!

Filtros de segmento

Avaliar grupos de interesse para a análise, de acordo com a relevância do grupo e a complexidade da análise.

Melhor começar pequeno, fazer um bom estudo e depois expandir o raciocínio do que começar grande e não chegar a lugar nenhum!

Exemplos de segmentação: filtros de produto / região / comportamentos

Atenção! Sempre avaliar junto à área de negócio!

Seleção de colunas: Qualidade

- Eliminar colunas com muitos valores nulos
- Eliminar colunas com distribuição irregular (ex: 97% dos registros possuem o mesmo valor)

Atenção! É uma excelente hora para rever se houve algum erro na geração do dataset!

Seleção de colunas: Relevância

- Eliminar colunas de baixíssima (ou inexistente) correlação com a variável target
- Eliminar colunas de altíssima correlação entre si (eliminar só uma das duas)

```
for col in df.select_dtypes(include=np.number).columns:  
    print(col, df[col].corr(df['nps']))
```

```
audio_length 0.042987001583190125  
messages 0.06403421980232389  
polarity 0.027060411946898243  
silenceDuration 0.028552807892433436  
userCharacters 0.06176253361256131  
attendantId -0.08000138848536215  
callId -0.0035366764051235136  
callday_activate_data_package nan  
callday_ask_invoice nan
```

Atenção! É uma excelente hora para rever se houve algum erro na geração do dataset!

Tratamento de nulos

Consultor: Carolina Bez
Abril de 2021

O que veremos nessa aula:

1. Tipos de nulos
2. Substituir por valor escolhido
3. Substituir por mode/median/average
4. Deletar o registro inteiro
5. Interpolação / Extrapolação
6. Forward filling / Backward filling — Hot Deck
7. Outros métodos

Tipos de Nulos

A primeira etapa é entender a origem do valor nulo:

- **Missing Completely at Random (MCAR)** – valores nulos independem de outras variáveis e deles mesmo
Ex: dados perdidos de forma accidental
- **Missing at Random (MAR)** - valores nulos dependem de outras variáveis e mas não dependem deles mesmo
Ex: Sensor de Temperatura / Perda de pacotes devido à queda de conexão
- **Not Missing at Random (NMAR)** – valores nulos dependem deles mesmo
Ex: Sensor de Temperatura não funciona abaixo de 5°C

Substituir por valor escolhido

- Depende do cenário
- Frequentemente podemos substituir por zero (principalmente após merges ou pivot-tables)
- Ex: Quero criar variável vl_total_compra fazendo um merge com o histórico de compras do cliente. Se o cliente não tiver compra, ficará como nulo □ posso substituir por zero
- Podemos substituir por máximo / mínimo / valor default:
- Ex: Sensor de temperatura quebrado a 5°C: Substituir por 4°C para dizer que é um número abaixo (**Cuidado!** Isso para o modelo pode funcionar, mas não mostre isso ao cliente. Na visualização, melhor colocar uma tag “Menor que 5°C”)

Substituir por moda/mediana/média

Depende do cenário e do algoritmo, a ideia é substituir pelo valor que menos afete o cálculo do algoritmo

- Ex: Se for variável categórica / ordinal / count, é preferível substituir por moda (valor mais frequente)
- Se for um algoritmo que use média para a tomada de decisão / calcular os pesos, é normalmente preferível usar a média (ex; kmeans, regressão linear)
- Se for um algoritmo que use a distribuição / percentis, é normalmente preferível usar a mediana (ex: árvore de decisão)
- Se a variável tiver muitos outliers, melhor mediana ou invés de média

Deletar o registro inteiro

- **Deletar linha:** deletar linhas que tenham tenham valores nulos em qualquer variável
- **Deletar par:** deletar linhas quem tenham valores nulos somente em variáveis que serão usadas na análise (recomendado somente se tipo de nulo for MCAR)
- **Deletar coluna:** deletar uma coluna inteira caso ela tenha um % de nulos acima de um limiar (ex: 70%)

Interpolação

- Útil para dados em série (ex: série temporal)
- Existem vários métodos para interpolação
- `pandas.DataFrame.interpolate`

df

	a	b	c	d
0	0.0	NaN	-1.0	1.0
1	NaN	2.0	NaN	NaN
2	2.0	3.0	NaN	9.0
3	NaN	4.0	-4.0	16.0

```
df.interpolate(method='linear', limit_direction='forward', axis=0)
```

	a	b	c	d
0	0.0	NaN	-1.0	1.0
1	1.0	2.0	-2.0	5.0
2	2.0	3.0	-3.0	9.0
3	2.0	4.0	-4.0	16.0

Forward filling / Backward filling

- `pandas.DataFrame.fillna(..., method={'backfill', 'bfill', 'pad', 'ffill', None}, ...)`
- `'pad', 'ffill'`: propaga a última observação válida para preencher o próximo valor nulo
- `'bfill', 'pad'`: propaga a próxima observação válida para preencher o último valor nulo

	A	B	C	D
0	NaN	2.0	NaN	0
1	3.0	4.0	NaN	1
2	NaN	NaN	NaN	5
3	NaN	3.0	NaN	4



```
>>> df.fillna(method='ffill')
   A    B    C    D
0  NaN  2.0 NaN    0
1  3.0  4.0 NaN    1
2  3.0  4.0 NaN    5
3  3.0  3.0 NaN    4
```

Outros métodos

- Treinar um modelo para o valor nulo (regressão, k-nearest neighbors etc.)
- Multiple imputation

O que veremos nessa aula:

1. Tipos de nulos
2. Substituir por valor escolhido
3. Substituir por mode/median/average
4. Deletar o registro inteiro
5. Interpolação / Extrapolação
6. Forward filling / Backward filling — Hot Deck
7. Outros métodos

Tratamento de *outliers*

Consultor: Carolina Bez
Abril de 2021

O que veremos nessa aula:

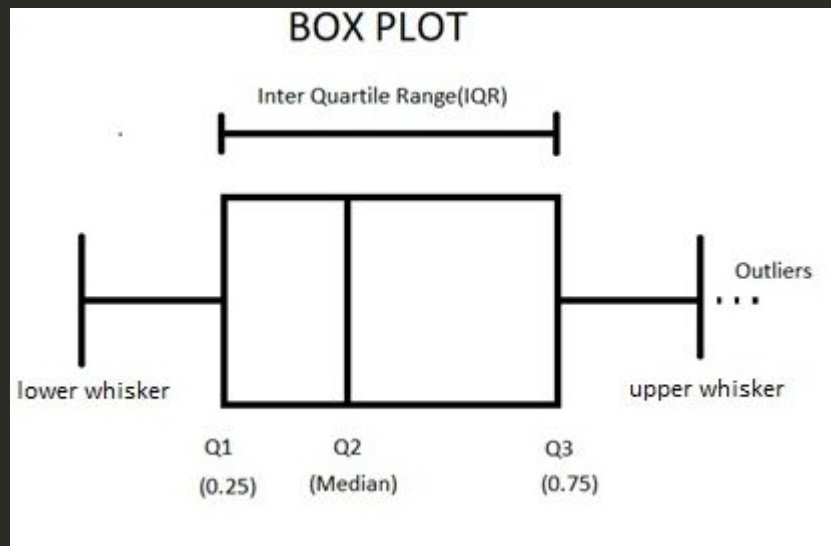
1. Tipos de Outlier
2. Identificar Outliers – Interquartile Range (IQR)
3. Identificar Outliers – Z Score
4. Corrigir Outliers
5. Demo

Tipos de Outlier

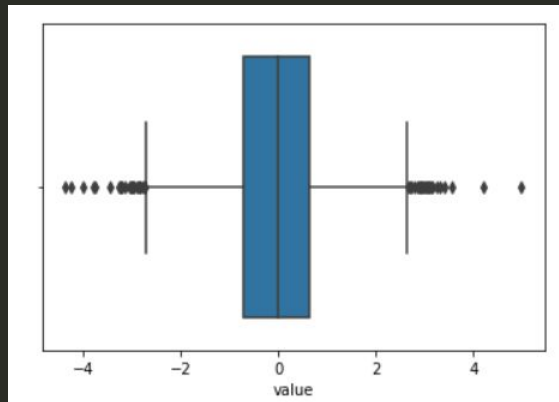
- Natural / Variabilidade dos dados
- Erro na mensuração / geração / gravação

Identificar Outliers – Interquartile Range (IQR)

- O registro possui valor acima de $Q3 + 1.5 \cdot IQR$ ou abaixo de $Q1 - 1.5 \cdot IQR$



Identificar Outliers – Interquartile Range (IQR)



```
Q1=df['value'].quantile(0.25)
Q3=df['value'].quantile(0.75)
IQR=Q3-Q1
```

```
print(IQR, IQR)
lower_bound = Q1-1.5*IQR
upper_bound = Q3+1.5*IQR
print("Normal Range", lower_bound, "-", upper_bound)
```

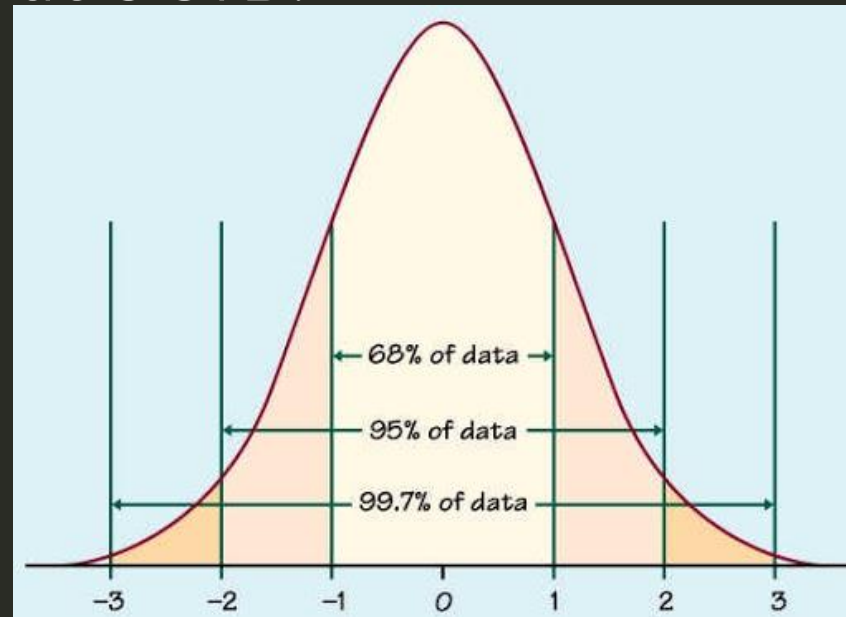
```
Q1 -0.7076964791646516
Q3 0.6417878224894843
IQR 1.3494843016541358
Normal Range -2.7319229316458555 - 2.666014274970688
```


Identificar Outliers – Z Score

- O registro possui valor acima de média + 3*STD ou abaixo de média - 3*STD
- Podemos usar o z-score: o registro possui z score fora da faixa de 3 STD.

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean
 σ = Standard Deviation



Corrigir Outliers

- Remoção
- Cap (min/max aceitáveis) ou fill com média / mediana / moda / valor default

```
df['value'].describe(percentiles=[0.01,0.1,0.25,0.50,0.75,0.90,0.99])
```

```
count    5000.000000
mean      -0.034785
std        1.014606
min       -4.381693
1%        -2.413475
10%       -1.295180
25%       -0.707696
50%       -0.024717
75%        0.641788
90%        1.239164
99%        2.307456
max        4.980069
Name: value, dtype: float64
```

Tratamento de datas

Consultor: Carolina Bez
Abril de 2021

O que veremos nessa aula:

1. Formatação de datas
2. Cálculo de datas
3. Demo

Formatação de datas

- `pandas.to_datetime()`

df

dt_compra

0 26/1/2016

1 5/11/2016

```
pd.to_datetime(df['dt_compra'].astype(str), format='%d/%m/%Y')
```

0 2016-01-26

1 2016-11-05

Name: dt_compra, dtype: datetime64[ns]

Formatação de datas

%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31
%b	Month as locale's abbreviated name.	Jan, Feb, ..., Dec (en_US); Jan, Feb, ..., Dez (de_DE)
%B	Month as locale's full name.	January, February, ..., December (en_US); Januar, Februar, ..., Dezember (de_DE)
%m	Month as a zero-padded decimal number.	01, 02, ..., 12
%y	Year without century as a zero-padded decimal number.	00, 01, ..., 99
%Y	Year with century as a decimal number.	0001, 0002, ..., 2013, 2014, ..., 9998, 9999

<https://docs.python.org/3/library/datetime.html#strftime-and-strptime-format-codes>

Cálculos de data

Calcular diferença (em dias, horas, minutos, meses, etc.)

```
df['dt_compra'] = pd.to_datetime(df['dt_compra'].astype(str), format='%d/%m/%y')
```

```
df['dt_venda'] = pd.to_datetime(df['dt_venda'].astype(str), format='%d/%m/%y')
```

```
df['qt_dias_dif'] = (df.dt_venda - df.dt_compra)
```

```
df['qt_dias_dif']
```

```
0] 0    31 days  
    1     4 days  
    Name: qt_dias_dif, dtype: timedelta64[ns]
```

```
df['qt_dias_dif'].map(lambda x: x.components.days)
```

```
0] 0    31  
    1     4  
    Name: qt dias dif, dtype: int64
```



Rolling Filters

Consultor: Carolina Bez

Abril de 2021

O que veremos nessa aula:

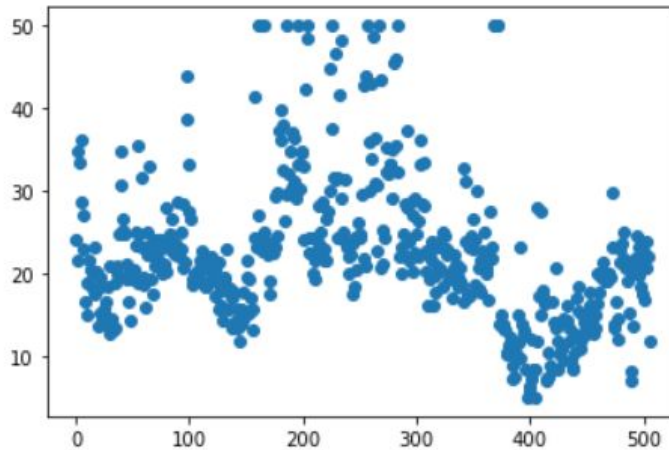
1. Rolling Filter com Mediana
2. Rolling Filter com Média

Rolling Filter com Média

- pandas.DataFrame.rolling

```
import matplotlib.pyplot as plt

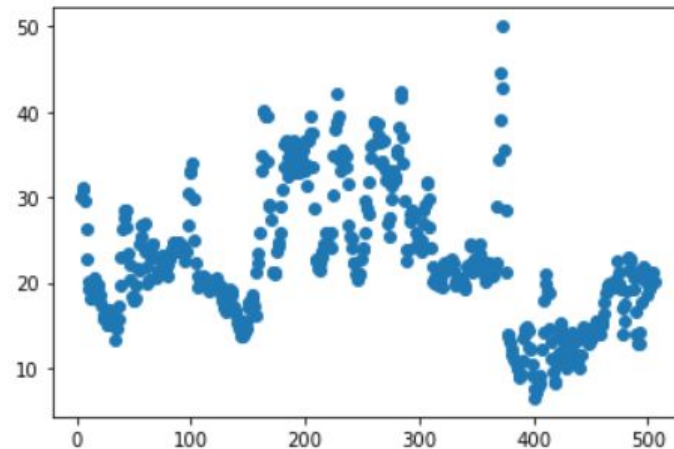
plt.scatter(x=df.index, y=df["value"])
plt.show()
```



```
df["less_noise_value"] = df.value.rolling(window=5).mean()
```

```
import matplotlib.pyplot as plt

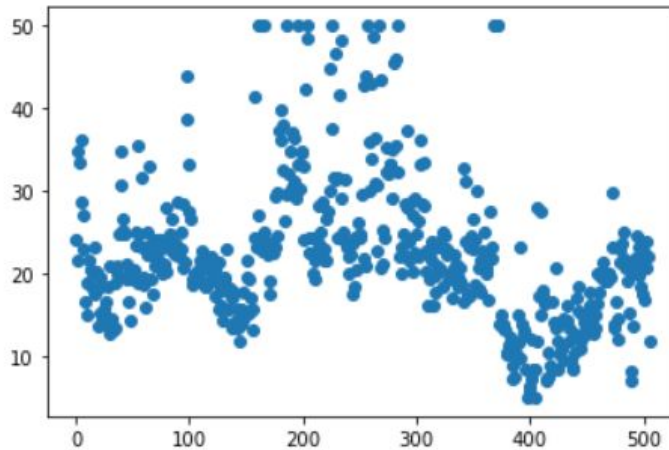
plt.scatter(x=df.index, y=df["less_noise_value"])
plt.show()
```



Rolling Filter com Média

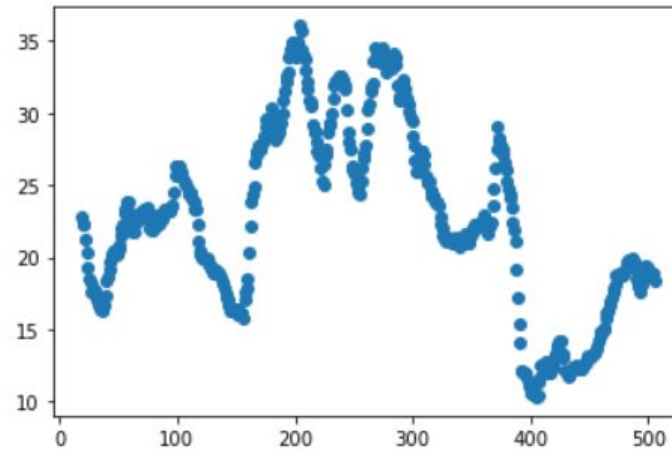
- pandas.DataFrame.rolling

```
import matplotlib.pyplot as plt  
  
plt.scatter(x=df.index,y=df["value"])  
plt.show()
```



```
df["less_noise_value"] = df.value.rolling(window=20).mean()
```

```
import matplotlib.pyplot as plt  
  
plt.scatter(x=df.index,y=df["less_noise_value"])  
plt.show()
```

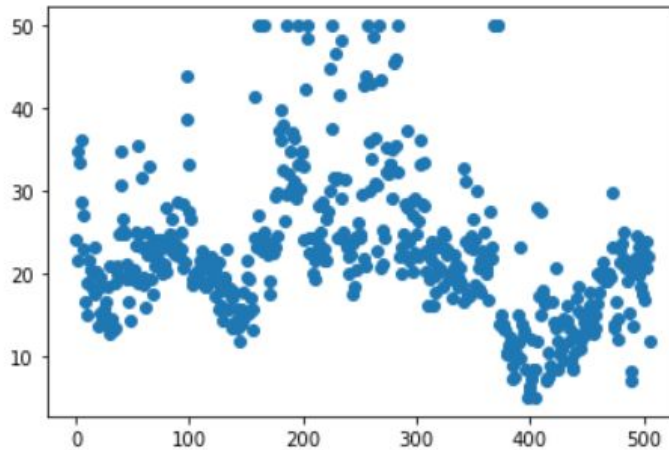


Rolling Filter com Mediana

- pandas.DataFrame.rolling

```
import matplotlib.pyplot as plt
```

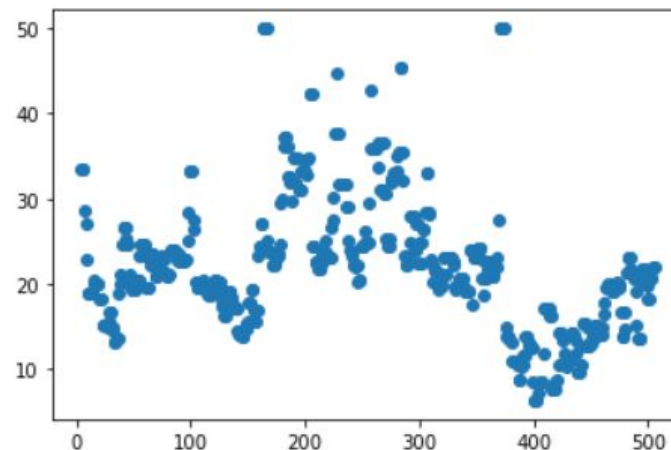
```
plt.scatter(x=df.index, y=df["value"])  
plt.show()
```



```
df["less_noise_value"] = df.value.rolling(window=5).median()
```

```
import matplotlib.pyplot as plt
```

```
plt.scatter(x=df.index, y=df["less_noise_value"])  
plt.show()
```



Enriquecimento dos dados – Operações Matemáticas

Consultor: Carolina Bez

O que veremos nessa aula:

1. Tipos de Operações
2. Inverso
3. Potências / Exponencial
4. Square-root / Log
5. Transformação Box-Cox
6. Somas / Diferenças / Multiplicação / Divisão
7. Demo

Operações

1. Univariadas
2. Bivariadas

Inverso

$$f(x) = \frac{1}{x}$$

Potências / Exponencial

$$f(x) = x^2$$

$$g(x) = x^3$$

$$h(x) = x^n$$

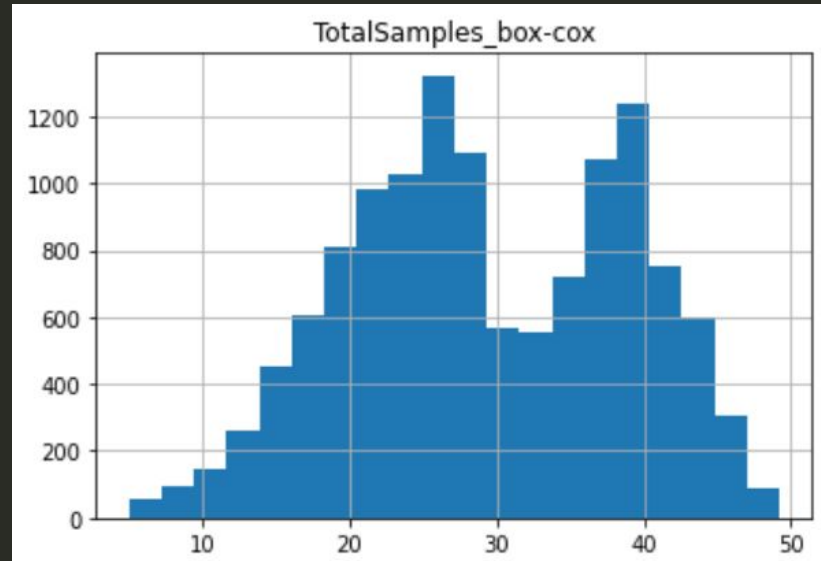
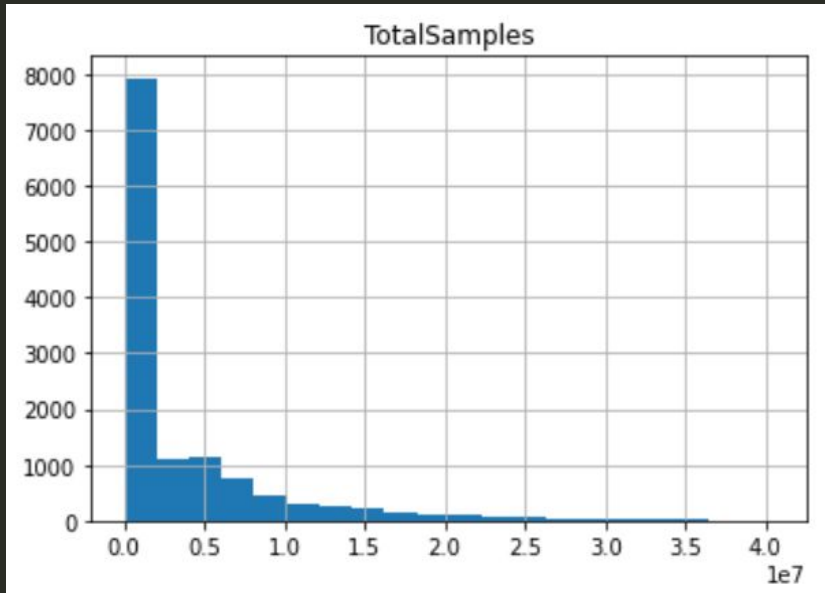
$$k(x) = \exp(x)$$

Square-root / Log

$$f(x) = \ln(x)$$

$$f(x) = \sqrt{x}$$

Transformação Box-Cox



Transformação Box-Cox

$$x_i^{(\lambda)} = \begin{cases} \frac{x_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(x_i) & \text{if } \lambda = 0, \end{cases}$$

Operações Matemáticas - Bivariadas

$$c = a + b$$

$$c = a - b$$

$$c = a * b$$

$$c = a / b$$

Enriquecimento dos dados – Operações Categóricas

Consultor: Carolina Bez

O que veremos nessa aula:

1. Recap: One-hot encoding
2. Recap: Cat Codes
3. Count / Frequency Mapping
4. Demo

Recap: One-Hot Encoding

	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3

Original

	idade	animal_dog	sexo_macho
0	1	1	1
1	2	0	0
2	3	1	0

One-Hot

Recap: Cat Codes

	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3

Original

	animal	sexo	idade
0	1	1	1
1	0	0	2
2	1	0	3

Cat Codes

Count / Frequency Mapping

	animal	sexo	idade
0	dog	macho	1
1	cat	femea	2
2	dog	femea	3

Original

	animal	sexo	idade
0	2	1	1
1	1	2	2
2	2	2	3

Count
Mapping

	animal	sexo	idade
0	0.666667	0.333333	1
1	0.333333	0.666667	2
2	0.666667	0.666667	3

Frequency
Mapping

Enriquecimento dos dados – Discretização

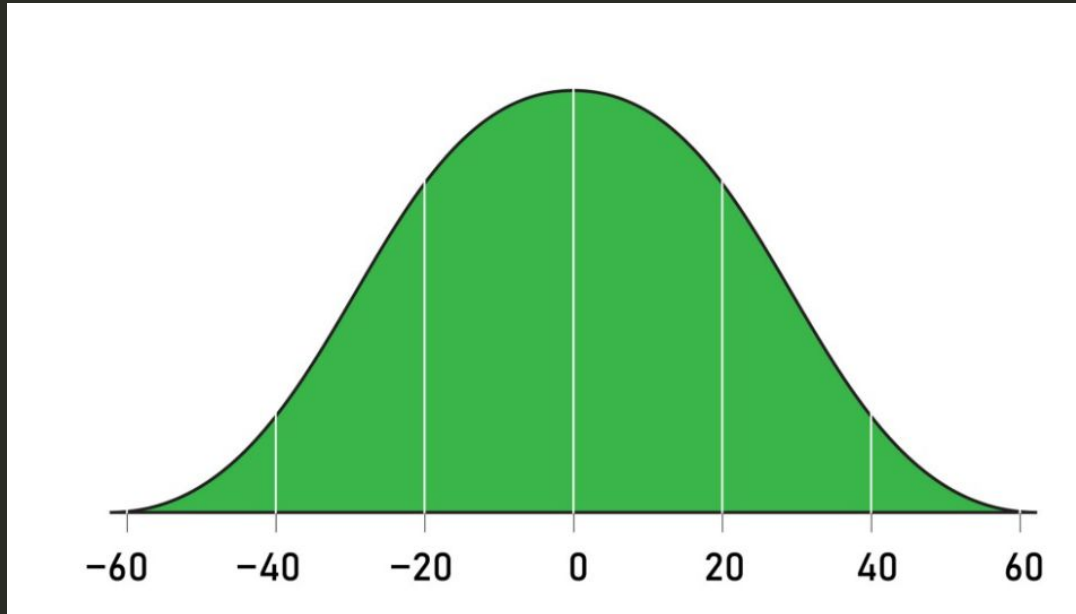
Consultor: Carolina Bez

O que veremos nessa aula:

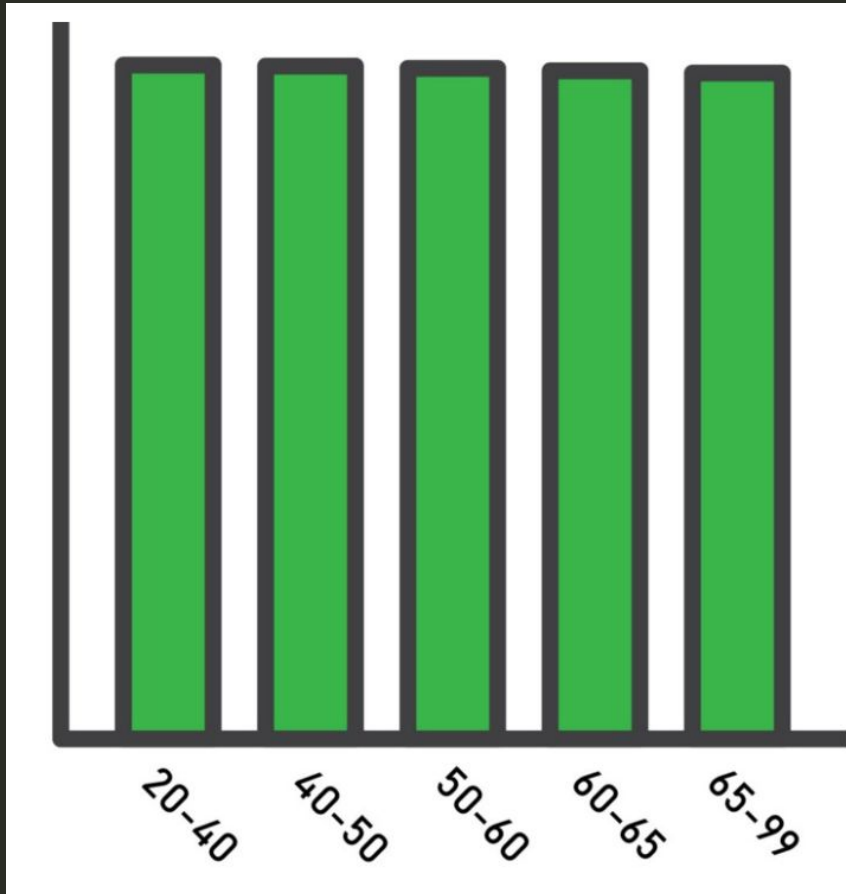
1. Equal-width discretization
2. Equal-frequency discretization
3. Demo

Equal-width discretization

$$width = \frac{maxvalue - minvalue}{N}$$



Equal-frequency discretization



Enriquecimento dos dados – Dimensões

Consultor: Carolina Bez

O que veremos nessa aula:

1. Novas Dimensões
2. Exemplo: ID Vendedor
3. Exemplo: CEP

Novas Dimensões

Em variáveis categóricas, principalmente Ids e nomes, podemos criar **métricas associadas a esse campo** ao invés desse campo em si.

ID	DT_CONSUMO	VL_CONSUMO	ID_VENDEDOR	NOTA_NPS
XX1	2020-11-03	50	X123	9
XX1	2020-11-15	21	X234	6
XX2	2020-11-18	33	X123	8
XX2	2020-12-16	23	X123	7
XX3	2020-11-06	45	X234	7
XX4	2020-12-18	46	X536	10
XX4	2020-11-18	12	X234	6
XX4	2020-11-04	34	X123	8

Novas Dimensões

Em variáveis categóricas, principalmente Ids e nomes, podemos criar **métricas associadas a esse campo** ao invés desse campo em si.

Exemplo: ID Vendedor

ID	DT_CONSUMO	VL_CONSUMO	ID_VENDEDOR	NOTA_NPS
XX1	2020-11-03	50	X123	9
XX1	2020-11-15	21	X234	6
XX2	2020-11-18	33	X123	8
XX2	2020-12-16	23	X123	7
XX3	2020-11-06	45	X234	7
XX4	2020-12-18	46	X536	10
XX4	2020-11-18	12	X234	6
XX4	2020-11-04	34	X123	8

Exemplo: ID Vendedor

Criação de uma dimensão Vendedor

ID_VENDEDOR	NOTA_NPS
X123	9
X123	8
X123	7
X123	8
X234	6
X234	7
X234	6
X536	10

ID_VENDEDOR	NOTA_NPS
X123	9
X123	8
X123	7
X123	8
X234	6
X234	7
X234	6
X536	10

ID_VENDEDOR	MÉDIA NOTA_NPS
X123	8.0
X234	6.3

Group by



Exemplo: ID Vendedor

ID	DT_CONSUMO	VL_CONSUMO	Média_NPS_VEN DEDOR	NOTA_NPS
XX1	2020-11-03	50	8.0	9
XX1	2020-11-15	21	6.3	6
XX2	2020-11-18	33	8.0	8
XX2	2020-12-16	23	8.0	7
XX3	2020-11-06	45	6.3	7
XX4	2020-12-18	46		10
XX4	2020-11-18	12	6.3	6
XX4	2020-11-04	34	8.0	8

Exemplo: ID Vendedor

ID	DT_CONSUMO	VL_CONSUMO	Média_NPS_VEN DEDOR	NOTA_NPS
XX1	2020-11-03	50	8.0	9
XX1	2020-11-15	21	6.3	6
XX2	2020-11-18	33	8.0	8
XX2	2020-12-16	23	8.0	7
XX3	2020-11-06	45	6.3	7
XX4	2020-12-18	46	7.6	10
XX4	2020-11-18	12	6.3	6
XX4	2020-11-04	34	8.0	8

Exemplo: CEP

ID	DT_CONSUMO	VL_CONSUMO	CEP_MORADIA
XX1	2020-11-03	50	22430095
XX1	2020-11-15	21	22430095
XX2	2020-11-18	33	22432240
XX2	2020-12-16	23	22432240
XX3	2020-11-06	45	21678190
XX4	2020-12-18	46	21654570
XX4	2020-11-18	12	21654570
XX4	2020-11-04	34	21654570

Exemplo: CEP

ID	DT_CONSUMO	VL_CONSUMO	CEP_MORADIA	CEP_3DIG
XX1	2020-11-03	50	22430095	224
XX1	2020-11-15	21	22430095	224
XX2	2020-11-18	33	22432240	224
XX2	2020-12-16	23	22432240	224
XX3	2020-11-06	45	21678190	216
XX4	2020-12-18	46	21654570	216
XX4	2020-11-18	12	21654570	216
XX4	2020-11-04	34	21654570	216

Exemplo: CEP

Criação de uma dimensão CEP_3Dig

CEP_3DIG	Média_VL_CONSUMO
224	31.75
216	34.25

Exemplo: CEP

ID	DT_CONSUMO	VL_CONSUMO	Média_VI_Consumo _CEP_3Dig
XX1	2020-11-03	50	31.75
XX1	2020-11-15	21	31.75
XX2	2020-11-18	33	31.75
XX2	2020-12-16	23	31.75
XX3	2020-11-06	45	34.25
XX4	2020-12-18	46	34.25
XX4	2020-11-18	12	34.25
XX4	2020-11-04	34	34.25

Enriquecimento dos dados – Operações de texto

Consultor: Carolina Bez

O que veremos nessa aula:

1. Word Tokenization
2. Word Stemming
3. Tratamento de Caixa e Dígitos
4. Remoção de Stop Words

Word Tokenization

"Eu estou estudando ciência de dados na
Dinâmica, estou amando!!!"



['Eu', 'estou', 'estudando', 'ciência', 'de', 'dados',
'na', 'Dinâmica', ',', 'estou', 'amando', '!', '!', '!']

Tratamento de Caixa e Dígitos

- 1- Colocar tudo em minúsculo
- 2- Remover pontuações e alguns caracteres especiais
- 3- Se irrelevante, remover números

Word Stemming

Seleciona o radical da palavra

estudando

estudei

estudar

estud

Remoção de Stop Words

Stop Words: palavras com pouco significado, não agregam informação semântica

Ex: de, a, eu, você, assim, porque

Validação dos dados

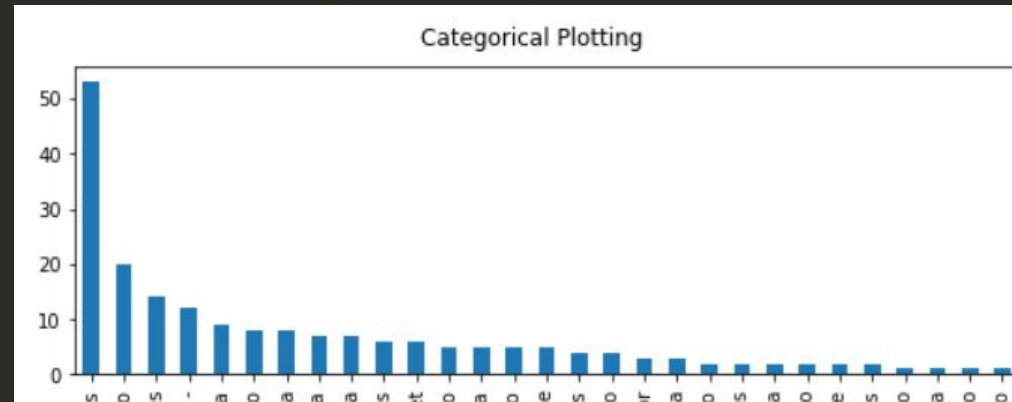
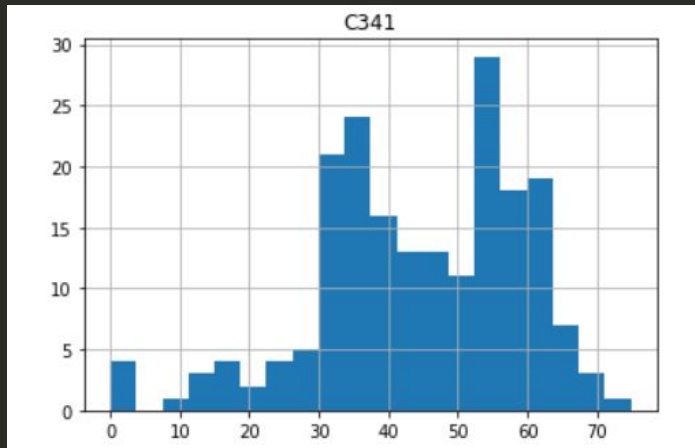
Consultor: Carolina Bez

O que veremos nessa aula:

1. Visualização
2. Checklist

Visualização

- Verificar novamente a distribuição de cada variável
- Em especial, as novas variáveis



Checklist

- Checar presença de nulos para cada variável
- Checar presença de outliers para cada variável
- Checar faixa de valores
- Checar variabilidade dos valores

```
In [26]: df["silenceDuration"].describe(percentiles=[.001, .01, .1, .25, .5, .75, .9, .99, .999])
```

```
Out[26]: count      50.000000  
         mean       47.693700  
         std        35.573071  
         min         1.898000  
         0.1%        1.927351  
         1%          2.191510  
         10%         8.463100  
         25%        20.914750  
         50%        41.030500  
         75%        62.728750  
         90%       100.180700  
         99%       141.894660  
         99.9%     157.623366  
         max        159.371000  
         Name: silenceDuration, dtype: float64
```

Estruturação do *pipeline*

Consultor: Carolina Bez

O que veremos nessa aula:

1. Organizar ordem do fluxo
2. Transformar validações em testes formais
3. Criar funções para generalizar o que foi feito
4. Organizar arquivos de código
5. Estruturar um pipeline

Organizar ordem do fluxo

- A ordem em que você gerou o código não necessariamente é a ordem mais lógica
- Agrupar trechos de códigos parecidos
(tratamento de nulo com tratamento de nulo, agrupamentos com agrupamentos)

Transformar validações em testes formais

Ex: assert

```
df['value'].describe(percentiles=[0.01,0.1,0.25,0.50,0.75,0.90,0.99])
```

```
count    5000.000000
mean      0.007052
std       1.021281
min      -4.395224
1%       -2.367227
10%      -1.286918
25%      -0.679073
50%       0.000656
75%       0.710529
90%       1.282575
99%       2.414630
max       3.908931
Name: value, dtype: float64
```



```
assert df['value'].describe(percentiles=[0.01,0.1,0.25,0.50,0.75,0.90,0.99])['max']<4
```


Criar funções para generalizar o que foi feito

	a	b	c	d
0	0.0	NaN	-1.0	1.0
1	NaN	2.0	NaN	NaN
2	2.0	3.0	NaN	9.0
3	NaN	4.0	-4.0	16.0

```
df.b = df.b.fillna(0)  
df.d = df.d.fillna(df.d.mean())
```

```
def treat_na(df, select_columns, fill_values):  
    for col in select_columns:  
        df[col] = df[col].fillna(fill_values[col])  
    return df
```

```
treat_na(df, ['b', 'd'], {'b': 0, 'd': df.d.mean()})
```

Organizar arquivos de código

- 1- Transformar Notebooks em Códigos Python
- 2- Criar módulos à parte se for necessário

Estruturar Pipeline

```
pandas.DataFrame.pipe()
```

DEMO

Produtização do *pipeline*

Consultor: Carolina Bez

O que veremos nessa aula:

1. Treino x Previsão
2. Treino x Previsão: Datas
3. Treino x Previsão: Categóricas
4. Treino x Previsão: Métricas Estatísticas

Treino x Previsão

ID	DT_ULTIM A_COMPRA	TIPO_ITEM_ MAIS_COMP RADO	ID_VENDEDO R	TOTAL_CON SUMO_ANTI S	FLG_COM PROU_NO VAMENTE	DT_COMP RA	NPS_COM PRA
XX1	20210201	COMPUTAD OR	X123	71	1	20210401	8
XX2	20201204	CELULAR	X234	33	1	20210404	8
XX3	20210330	CELULAR	X123	45	0	20210430	7
XX4	20210113	COMPUTAD OR	X234	46	0	20210413	5

ID	DT_ULTIM A_COMPRA	TIPO_ITEM_ MAIS_COMP RADO	ID_VENDEDO R	TOTAL_CON SUMO_ANTI S	FLG_COM PROU_NO VAMENTE	DT_COMP RA	NPS_COM PRA
XX5	20210401	COMPUTAD OR	X123	54			

Treino x Previsão: Data

ID	DT_ULTIMA_COMPRA	QT_MESES_ULTIMA_COMPRA	TIPO_ITEM_MAIS_COMPRADO	ID_VENDEDOR	TOTAL_CONSUMO_ANNOS	FLG_COMPROU_NOVAMENTE	DT_COMPRA	NPS_COMPRA
XX1	20210201	2	COMPUTADOR	X123	71	1	20210401	8
XX2	20201204	5	CELULAR	X234	33	1	20210404	8
XX3	20210330	1	CELULAR	X123	45	0	20210430	7
XX4	20210113	3	COMPUTADOR	X234	46	0	20210413	5

$$QT_MESES_ULTIMA_COMPRA = DT_COMPRA - DT_ULTIMA_COMPRA$$

ID	DT_ULTIMA_COMPRA	QT_MESES_ULTIMA_COMPRA	TIPO_ITEM_MAIS_COMPRADO	ID_VENDEDOR	TOTAL_CONSUMO_ANNOS	FLG_COMPROU_NOVAMENTE	DT_COMPRA	NPS_COMPRA
XX5	20210401	?	COMPUTADOR	X123	54			

$$QT_MESES_ULTIMA_COMPRA = <CURRENT_DATE> - DT_ULTIMA_COMPRA$$

Treino x Previsão: Categorias

ID	DT_ULTI MA_CO MPRA	TIPO_ITE M_MAI S_COM PRADO	FL_COMP UTADOR	FL_CELU LAR	ID_VEND EDOR	TOTAL_C ONSUMO _ANTES	FLG_C OMPRO U_NOV AMENT E	DT_CO MPRA	NPS_C OMPRA
XX1	2021020 1	COMPUT ADOR	1	0	X123	71	1	202104 01	8
XX2	2020120 4	CELULAR	0	1	X234	33	1	202104 04	8
XX3	2021033 0	CELULAR	0	1	X123	45	0	202104 30	7
XX4	2021011 3	COMPUT ADOR	1	0	X234	46	0	202104 13	5

SURGIMENTO DE OUTRA CATEGORIA FORA DO DATASET DE TREINO:

ID	DT_ULTI MA_CO MPRA	TIPO_ITE M_MAI S_COM PRADO	FL_COMP UTADOR	FL_CELU LAR	ID_VEND EDOR	TOTAL_C ONSUMO _ANTES	FLG_C OMPRO U_NOV AMENT E	DT_CO MPRA	NPS_C OMPRA
XX5	2021040 1	TECLADO	0	0	X123	54			

Treino x Previsão: Métricas Estatísticas

ID	DT_ULTI MA_COM PRA	TIPO_ITEM _MAIS_CO MPRADO	ID_VENDE DOR	Média_NPS _VENDEDO R	TOTAL_CO NSUMO_A NTES	FLG_CO MPROU_ NOVAME NTE	DT_COM PRA	NPS_CO MPRA
XX1	20210201	COMPUTA DOR	X123	7.5	71	1	2021040 1	8
XX2	20201204	CELULAR	X234	6.5	33	1	2021040 4	8
XX3	20210330	CELULAR	X123	7.5	45	0	2021043 0	7
XX4	20210113	COMPUTA DOR	X234	6.5	46	0	2021041 3	5

ID	DT_ULTI MA_COM PRA	TIPO_ITEM _MAIS_CO MPRADO	ID_VENDE DOR	Média_NPS _VENDEDO R	TOTAL_CO NSUMO_A NTES	FLG_CO MPROU_ NOVAME NTE	DT_COM PRA	NPS_CO MPRA
XX5	20210401	COMPUTA DOR	X123	7.5	54			