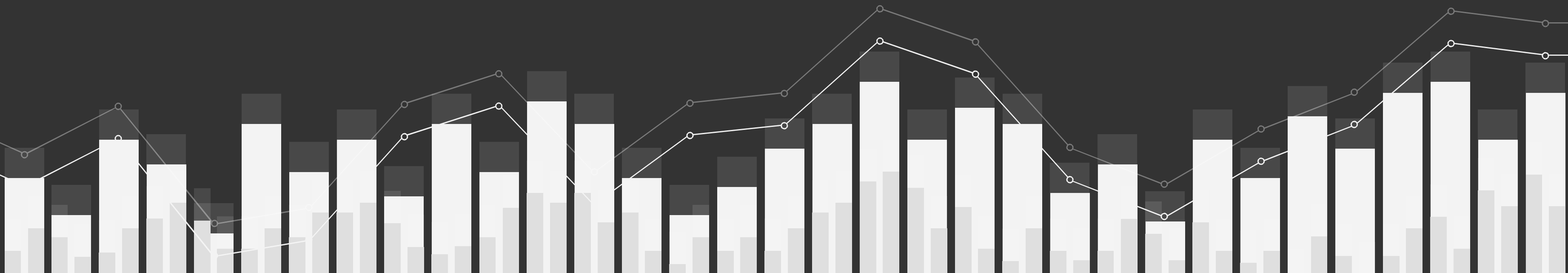
Trading com Dados



Sócios Fundadores

Victor Gomes, CEO

Engenheiro Mecânico pela UFRN, Engenheiro Industrial pela Northeastern University e MBA em Investimentos e Private Banking pelo Ibmec, com passagens por Itaú e XP Inc.

Gustavo Abud, CMO

Pós-Graduado em análise de dados e métodos quantitativos pela FIA, com solida experiência em autorregulação no mercado de capitais, tesouraria e corretora de valores, trabalhou em projetos de PLD, Treasury Business Control e Op Risk passando por empresas como B3, IBBA e XP Inc.

Lucas Corrêa, CFO

Profissional da área de dados e analytics, com passagens pela Embraer e Itaú Unibanco. Atualmente faz parte do time de data science da B3 e faz parte do quadro de professores do LABDATA na FIA.

Yago Luz, UX/Designer

Graduando em Engenharia de Gestão pela UFABC. Atualmente está na XP Investimentos, onde trabalha com Gestão de Metas. Possui experiência com UX, Marketing e Gestão de Mídias Sociais.

Nossos produtos educacionais

Cursos e workshops presenciais e online

Academy

TD Academy, nosso portal de cursos online

E-books

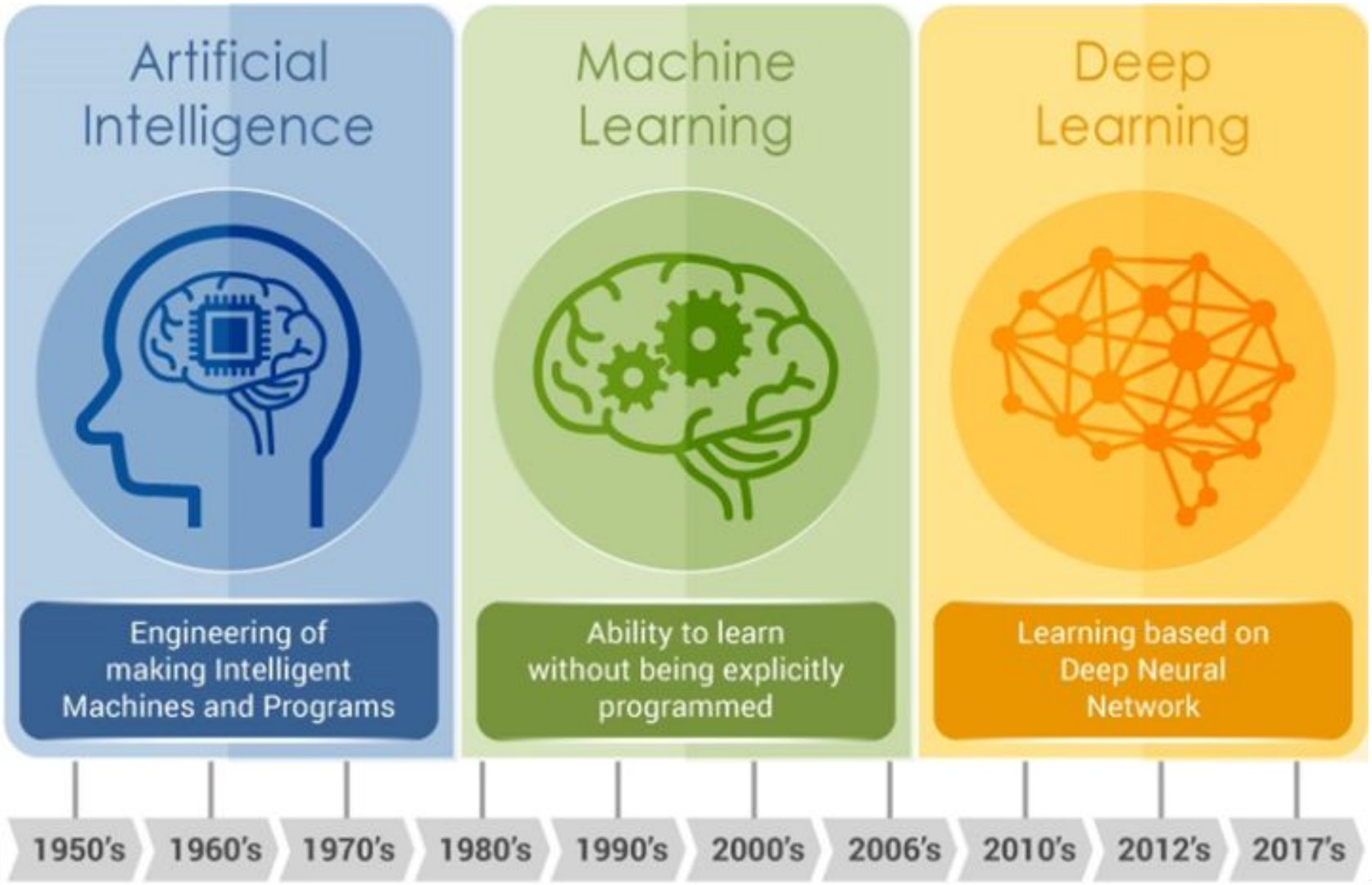
Artigos Premium

Mentoria

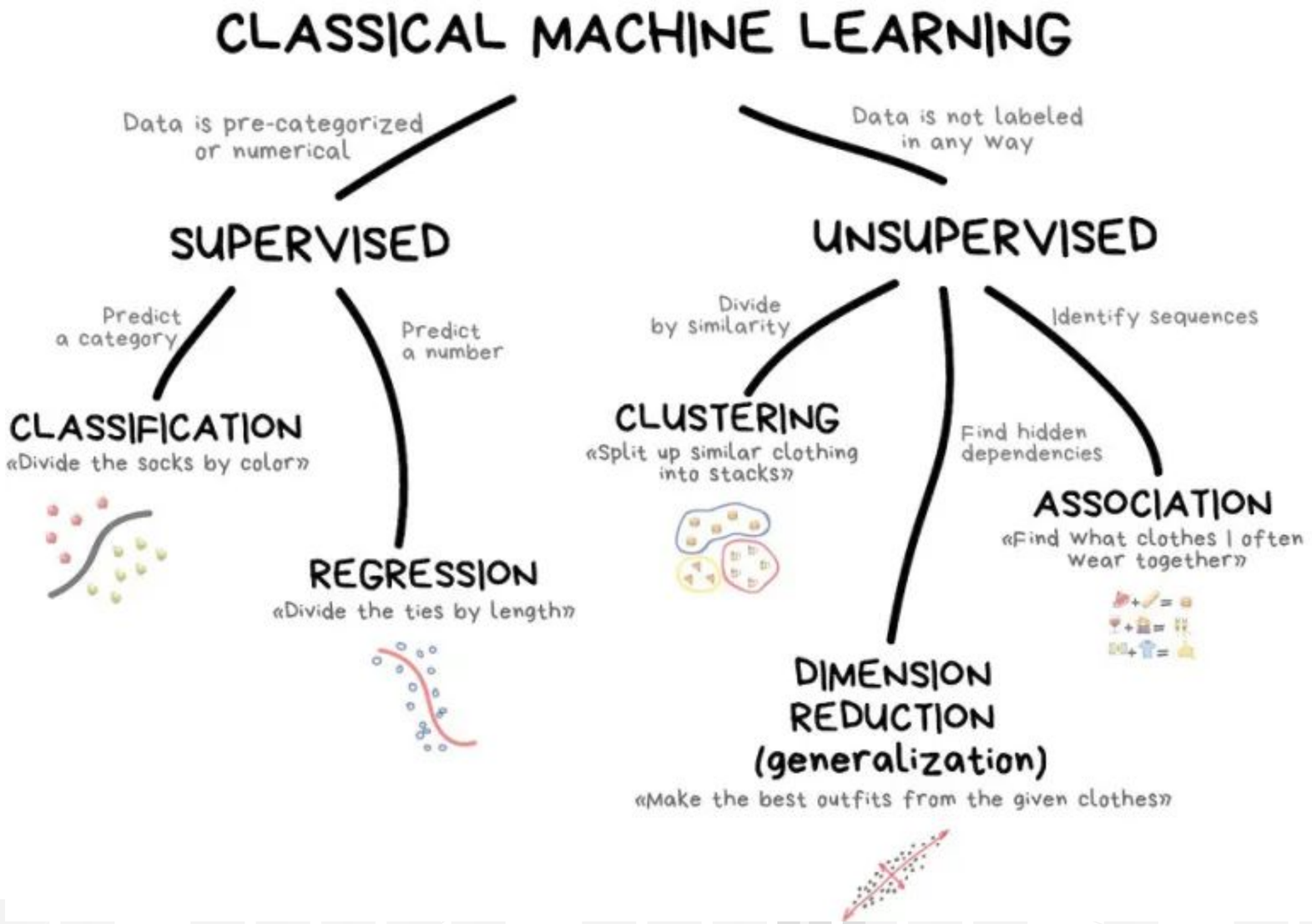
TradingCast

Nosso podcast

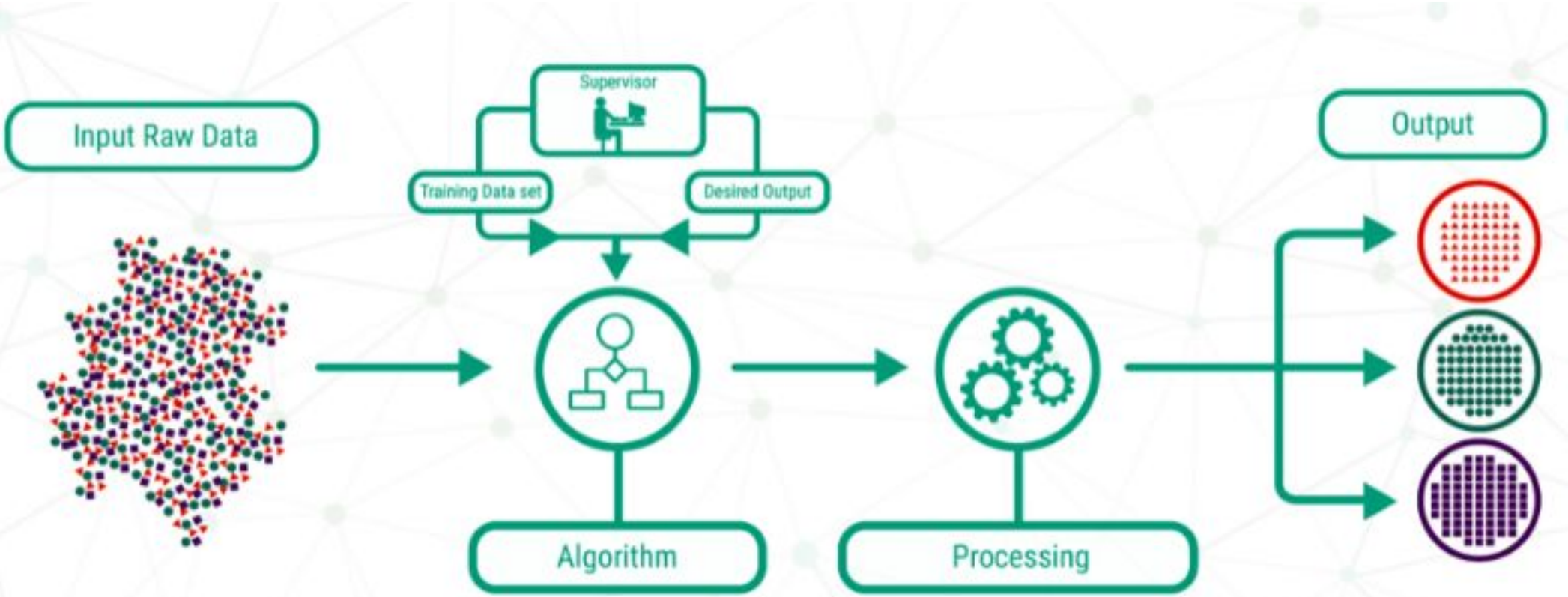
O que é Machine Learning?



Classe de algoritmos



Modelos supervisionados - classificação



Algoritmos de classificação

- Regressão Logística
- Árvore
- Random Forest
- Gradient Boosting
- Redes Neurais
- Deep Learning

Modelos supervisionados - classificação

Característica:

- Precisa ter uma variável resposta

Estrutura:

Variáveis independentes / explicativas

Variável dependente / Target

X1	x2	x3	x4	x5	x6	x7	...	xi	Target
23	H	22	11	C	345	78,9	...	3	0 / 1

Modelos supervisionados - regressão

Característica:

- Precisa ter uma variável resposta
- A variável queremos prever é contínua (Ex.: R$27mil) - Reg Linear

Estrutura:

Variáveis independentes / explicativas

Variável dependente / Target

X1	x2	x3	x4	x5	x6	x7	...	xi	Target
23	H	22	11	C	345	78,9	...	3	293

Modelos não supervisionados

Tipos de modelos:

- Agrupamento
- Redução de dimensionalidade
- Associação

Modelos não supervisionados

Agrupamento:

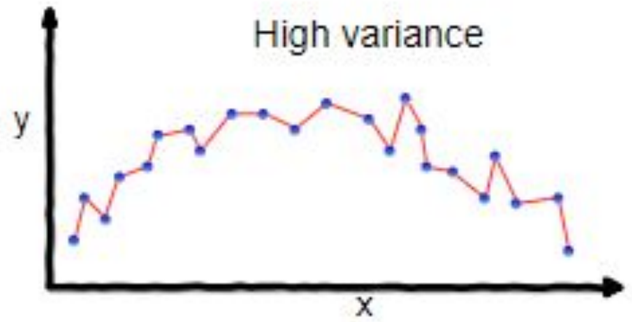
sample

Cluster/group

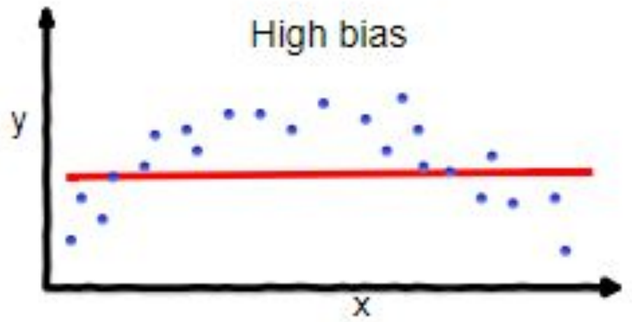
Famílias de algoritmos de agrupamento

Particionamento	Hierárquicos	Density-based
• K-means	• Aglomerativo	• DBSCAN

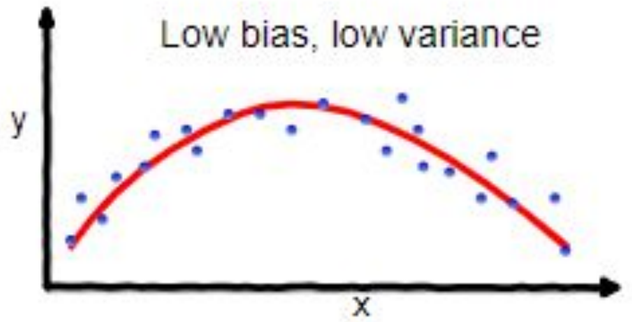
Um modelo busca sempre a melhor generalização



overfitting

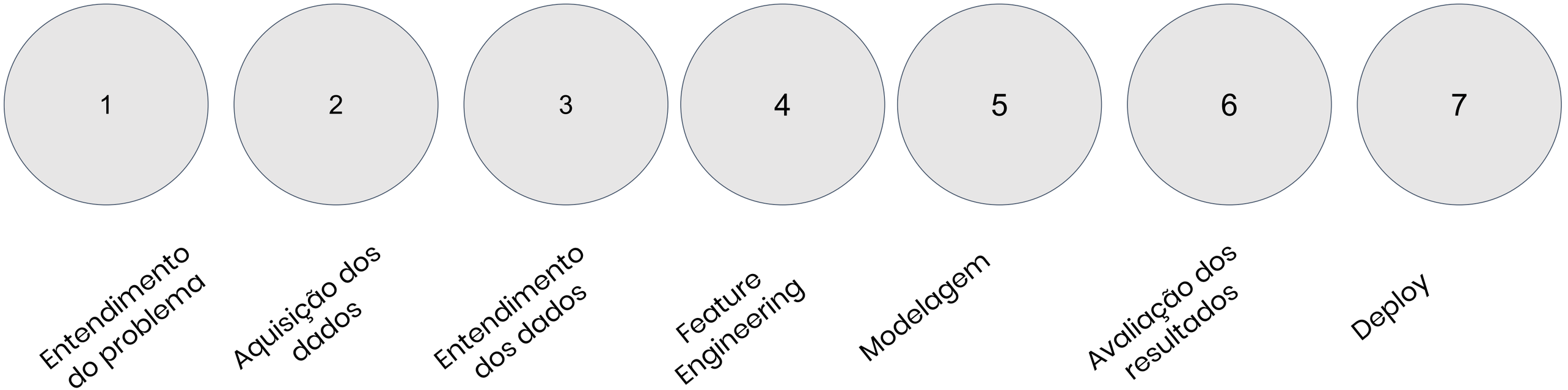


underfitting



Good balance

Framework de Modelagem de dados



EDA - Exploratory Data Analysis

A fase de EDA é sempre feita duas vezes no ciclo de modelagem:

1. Fase de entendimento dos dados em n bases;
2. Fase de entendimento dos dados na ABT;

Primeiros passos na análise descritiva estão ligados a entender o tipo variável que estamos trabalhando.

Tipos de variáveis

Os dados podem ser:

Qualitativos:

Qualitativo **nominal**

1. defeito no ar condicionado;

2. defeito no vidro;

3. defeito no freio de mão;

Tipos de variáveis

Qualitativo **ordinal**:

E - excelente;

MB - muito bom;

B - bom;

R - ruim;

Tipos de variáveis

Temos também as variáveis quantitativas:

Quantitativas **discretas**:

X - número de pessoas;

Y - número de compras;

Z - número de casos de coronavírus;

Tipos de variáveis

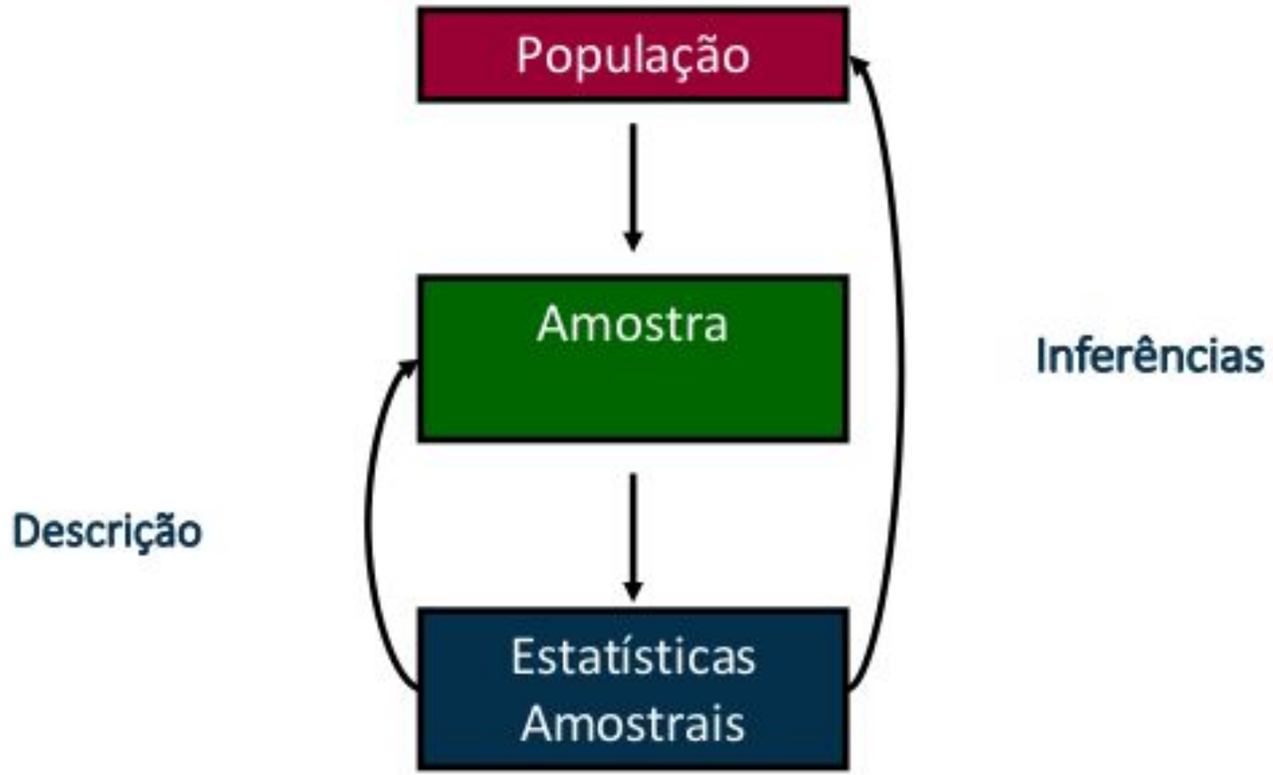
Quantitativas **contínuas**:

X - valor da compra;

Y - preço do imóvel;

Z - valor monetário no banco;

EDA - Exploratory Data Analysis



EDA - Exploratory Data Analysis

Terminologia usada:

Parâmetro - medida usada para descrever uma característica da população;

Estatística - característica da amostra.

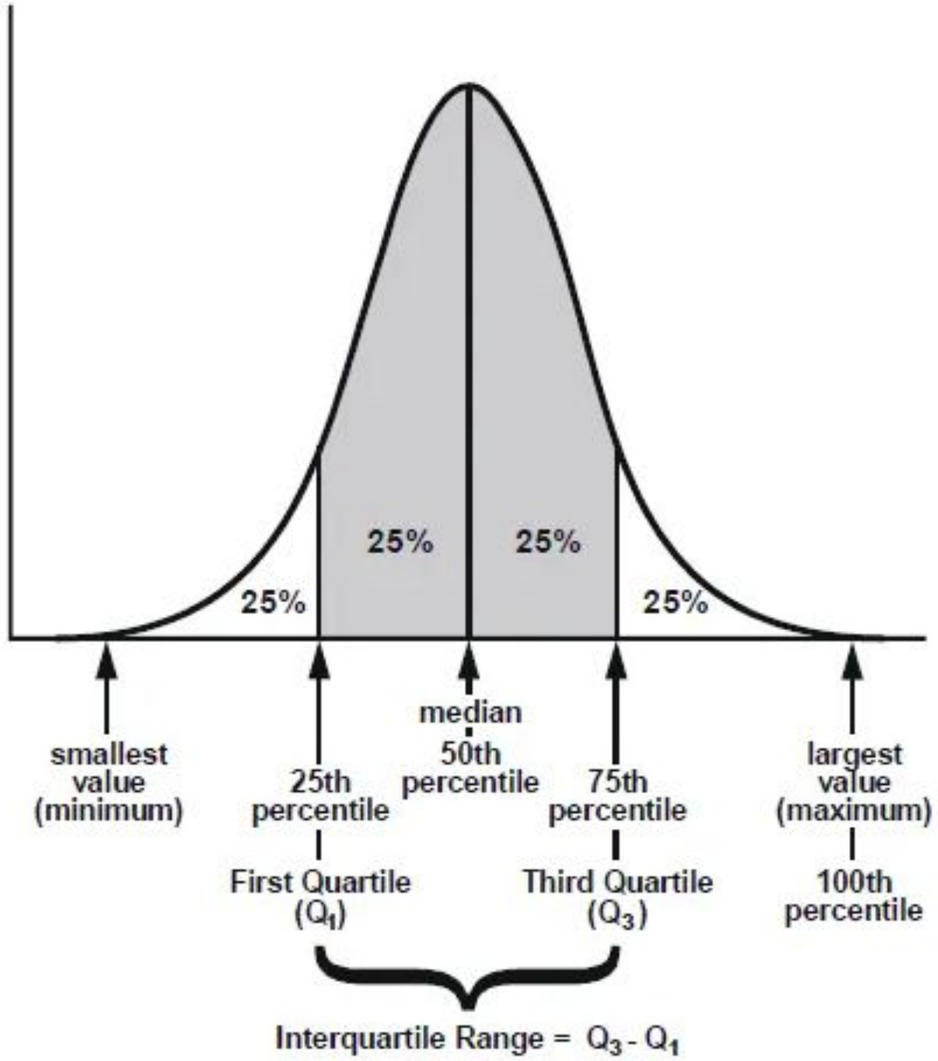
	Parâmetros Populacionais	Estatísticas Amostrais
Media	μ	$\bar{x}$
Variância	σ^2	s^2
Desvio Padrão	σ	s

EDA - Exploratory Data Analysis

Como podemos extrair significado dos dados?

- medidas de posição;
- medidas de dispersão;
- análise gráfica;
- medidas de assimetria;
- medidas de associação;

Interquartile Range - IQR

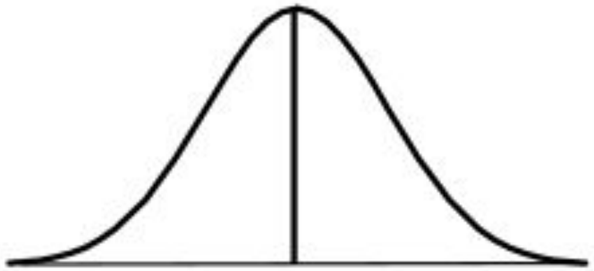


Assimetria dos dados

Mostra o quanto e como a distribuição de frequências se distancia de uma forma simétrica.

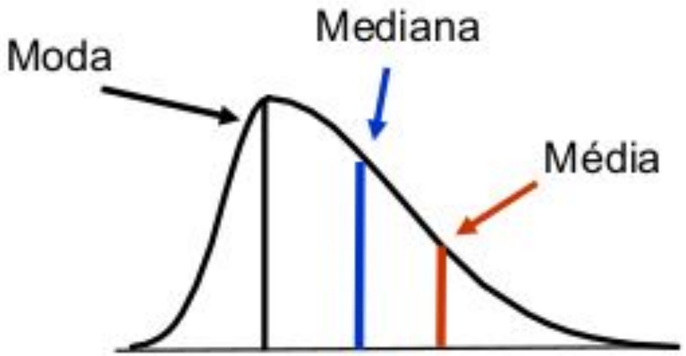
Distribuição Simétrica

Média = Mediana = Moda



Assimetria = 0

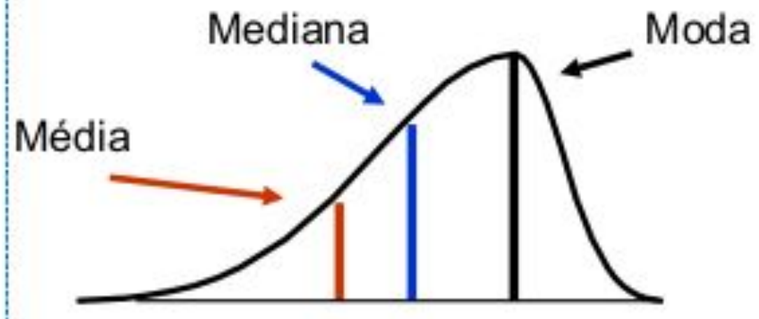
Assimetria à direita ou positiva



Assimetria >0

Cauda à direita mais pesada
(valores acima da média)

Assimetria à esquerda ou negativa



Assimetria < 0

Cauda à esquerda mais pesada
(valores abaixo da média)

Feature engineering

Passos:

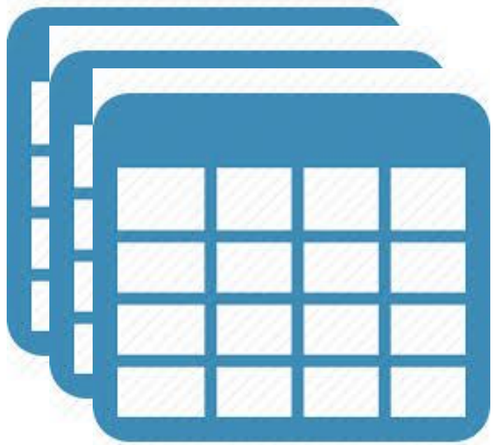
- Juntar os dados das diversas bases em uma tabela única;
- Criar a target, se aplicável;
- Fazer o tratamento de missing;
- Binning de variáveis;
- Seleção de atributos;

Feature engineering

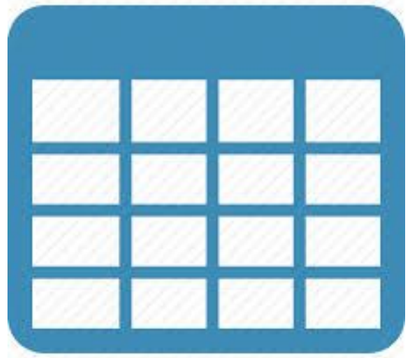
Criar um pipeline de tratamento dos dados

Raw data

stage



book



ABT

Tratamento de variáveis

Identificar os tipos de variáveis:

Tipo da Variável

```
data.dtypes
```

```
ID                      int64
Target                  int64
GrupoEconomico          int64
Sexo                   object
Idade                   int64
GrupoRisco              int64
ValorCompraAnual      float64
GastoMax              float64
GastoMedio            float64
UF                     object
CidadeResidencia       object
RegiaodoPais           object
NumeroComprasOnline   float64
dtype: object
```

UF
BA
BA
RO
PI
MA

```
data.tail(5)
```

	ID	Target	GrupoEconomico	Sexo	Idade	GrupoRisco
886	887	0	2	homem	27	0
887	888	1	1	mulher	19	0
888	889	0	3	mulher	35	1
889	890	1	1	homem	26	0
890	891	0	3	homem	32	0

Tratamento de missings

Para categórica:

```
data['Sexo'] = data['Sexo'].fillna('MISS')
```

- Criar uma nova classe “MISS” que sinalize o missing.

Para numérica:

```
data['ValorCompraAnual'].fillna(data['ValorCompraAnual'].mean(), inplace=True)
```

Imputar a média, mediana, moda ou rodar um modelo de regressão.

Criando as variáveis dummy

Depois de tratado o missing, precisamos agora transformar as categóricas em variáveis dummy.

Observar cardinalidade da variável e realizar o get_dummies ou label_enconder *

Para isso usaremos a função do pandas: pd.get_dummies() e a função do sklearn: LabelEncoder()

* Cuidado no uso do label enconder, var pode tornar-se ordinal

Criando as variáveis dummy

Analisando a cardinalidade:

```
data.nunique()
```

```
ID                    891
Target                  2
GrupoEconomico          3
Sexo                    2
Idade                  65
GrupoRisco              7
ValorCompraAnual      261
GastoMax              261
GastoMedio            261
UF                     26
CidadeResidencia      851
RegiaodoPais            5
NumeroComprasOnline   704
dtype: int64
```

Para cardinalidade <=10:

pd.get_dummies

Cardinalidade >10:

LabelEncoder

get_dummies / LabelEncoder

Cardinalidade <=10:

```
#gerando variáveis dummies
dum = pd.get_dummies(data,
                     columns=['GrupoEconomico','Sexo','GrupoRisco', 'RegiaodoPais'],
                     drop_first=True,
                     prefix = ['GrupoEconomico','Sexo','GrupoRisco', 'RegiaodoPais'],
                     prefix_sep='_')
dum.head()
```

Cardinalidade >10:

```
from sklearn import preprocessing

le = preprocessing.LabelEncoder()


le_tkt = le.fit_transform(dum['UF'])
le_tkt_df1 = pd.DataFrame(le_tkt, columns=['LE_UF'])
```

Normalizar variável?

Depende do contexto do problema;
Não é algo necessário sempre;

Prós:

- Melhora o custo computacional;

Contras:

- Perde a sensibilidade do dado;

Tipos de normalização

Standardization:

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

with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$

Min-Max scaling:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

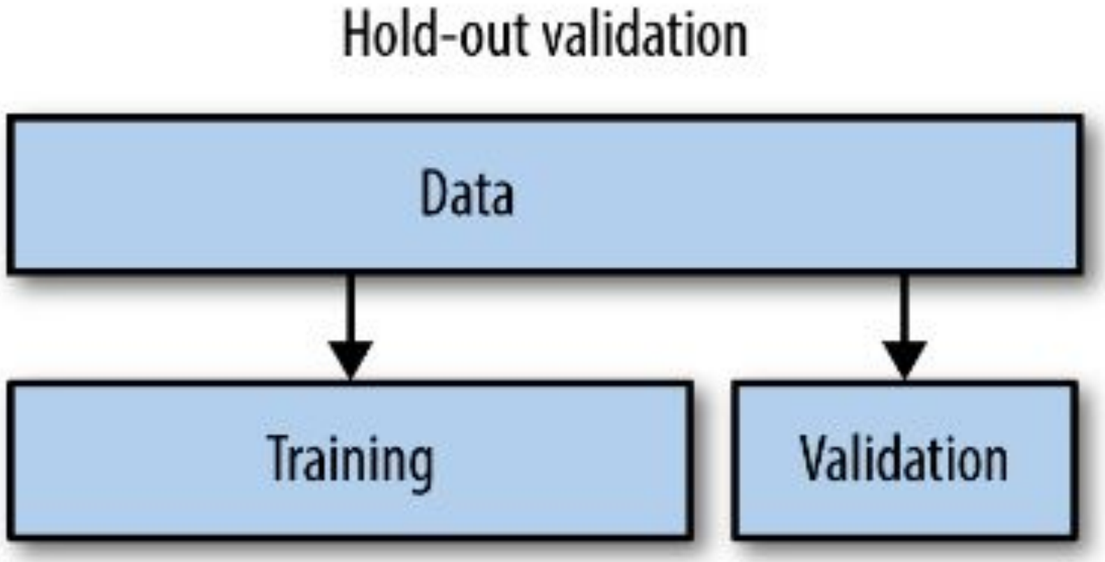
Cross validation

As técnicas existentes para cross validation são:

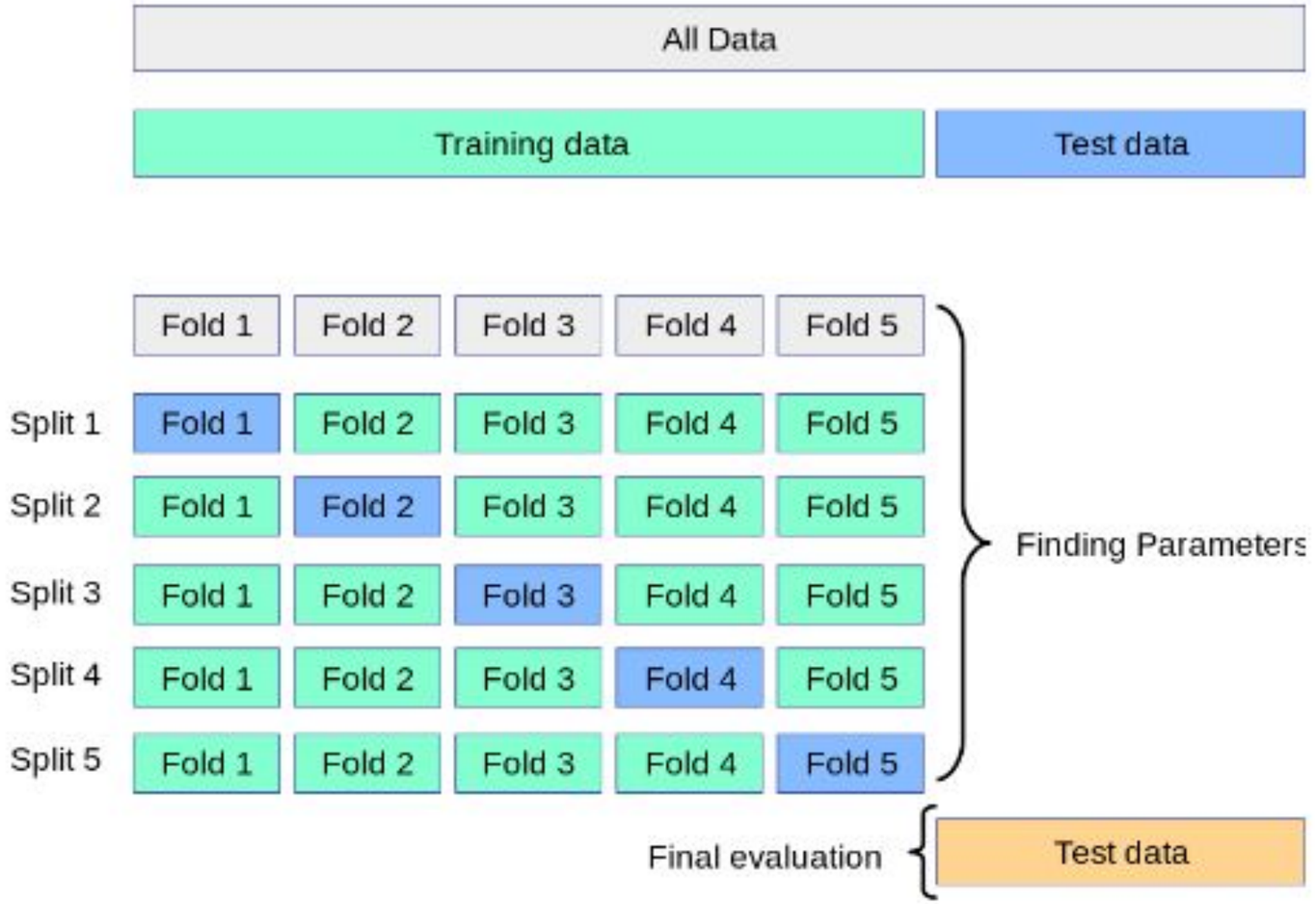
- Hold out;
- k-folds;
- Leave one out;

Cross validation

- Hold out



K-fold


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

Cross validation

Leave one out

```python
from sklearn.model_selection import LeaveOneOut
X = [1, 2, 3, 4]
loo = LeaveOneOut()
for train, test in loo.split(X):
    print("%s %s" % (train, test))

```

```
[1 2 3] [0]
[0 2 3] [1]
[0 1 3] [2]
[0 1 2] [3]
```

Quanto o modelo está acertando?

Métricas de qualidade e ajuste

```
>>> from sklearn.metrics import accuracy_score
>>> y_pred = [0, 2, 1, 3]
>>> y_true = [0, 1, 2, 3]
>>> accuracy_score(y_true, y_pred)
0.5
>>> accuracy_score(y_true, y_pred, normalize=False)
2
```

Obrigado!