Aplicações de Aprendizado de Máquina & PLN

Juvenal J. Duarte

Ementa

- Conteúdo:
  - 19/06: Introdução + Regressão + Reg. Linear + Arv. Regressão
  - 26/06: Classificação + Arv. Decisão + KNN
  - 10/07: Análise de agrupamentos + K-Means
  - 24/07: Recomendação / Regras de associação + Apriori

Avaliação

- Serão quatro aulas, cada uma com uma atividade valendo $^1/_4$ da nota.

- O critério de aprovação é obtenção de conceito igual ou superior a 7 na media entre as atividades.

- Exercícios com implementação em Python (Pandas + Numpy + etc), entrega via Jupyter Notebook.

Pré Requisitos

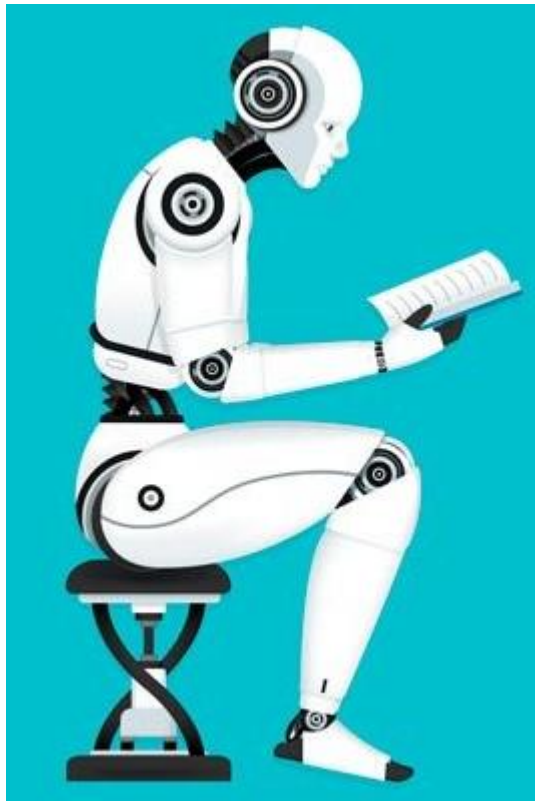
- Background em programação Python + Jupyter Notebook.
- Conhecimento básico em algebra, cálculo e prob. & estatística.
- Inglês técnico.

Introdução:
Aprendizado de Máquina

Juvenal J. Duarte

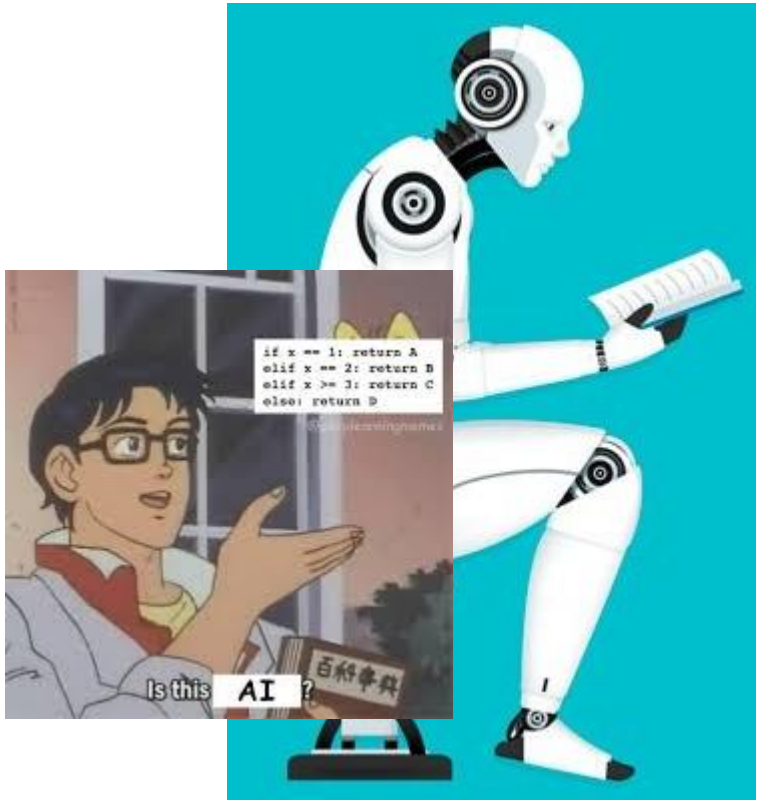
Aprendizado de Máquina

- Como ocorre a aprendizagem?



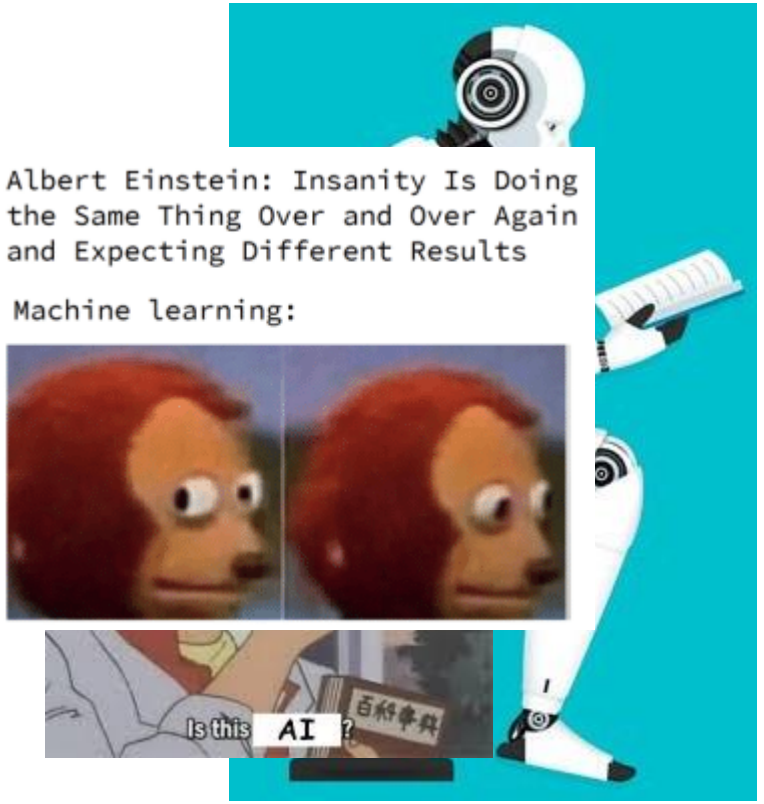
Aprendizado de Máquina

- Como ocorre a aprendizagem?
- Afinal, o que diferencia ML de programação comum?



Aprendizado de Máquina

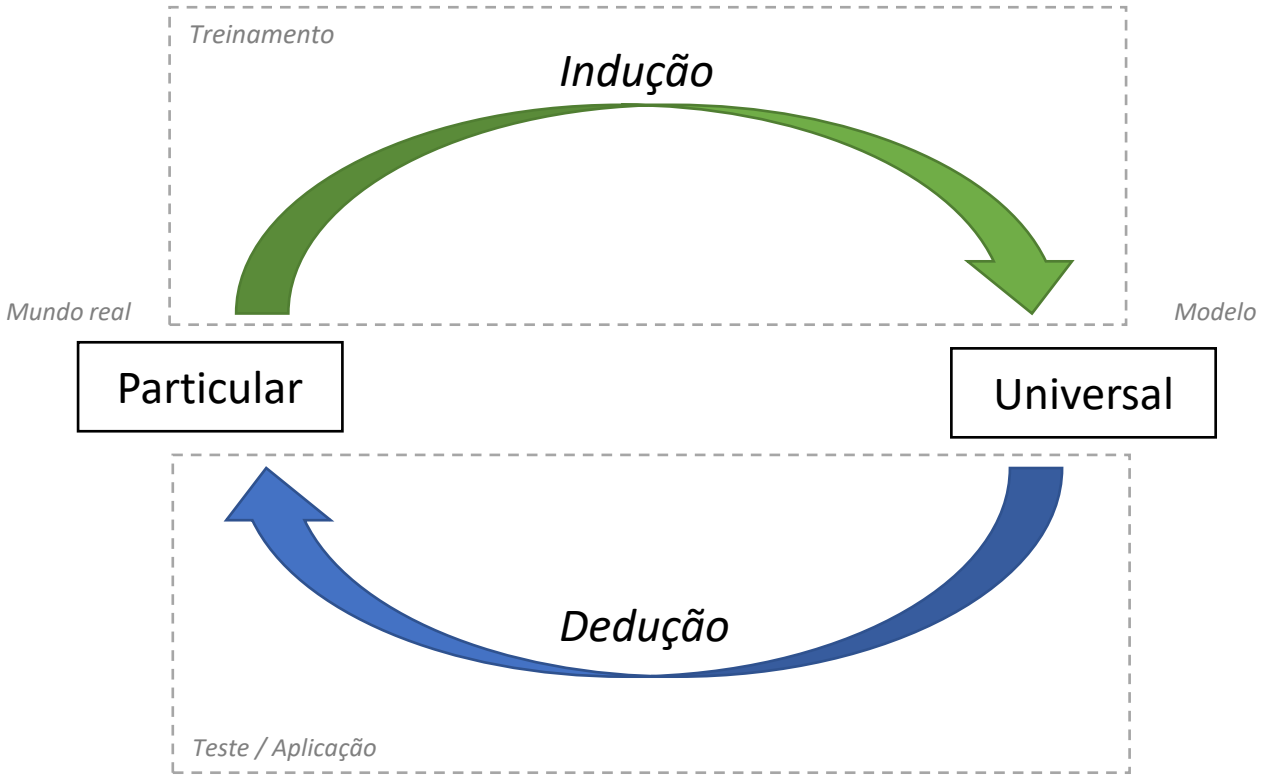
- Como ocorre a aprendizagem?
- Afinal, o que diferencia ML de programação comum?
- Quais as limitações?



Aprendizado de Máquina

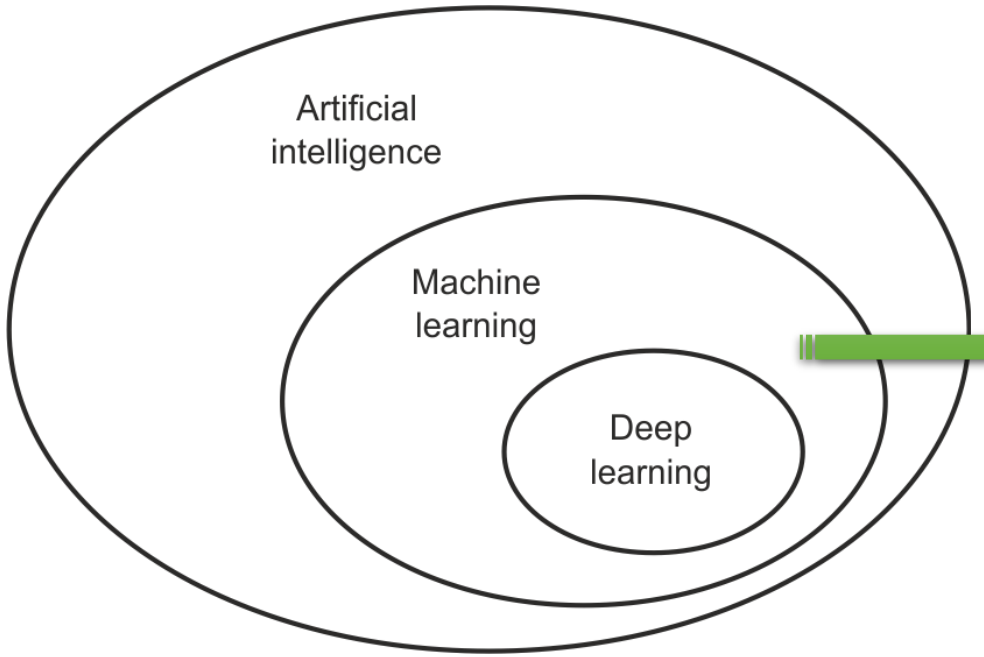
- Como ocorre a aprendizagem?
- Afinal, o que diferencia ML de programação comum?
- Quais as limitações?

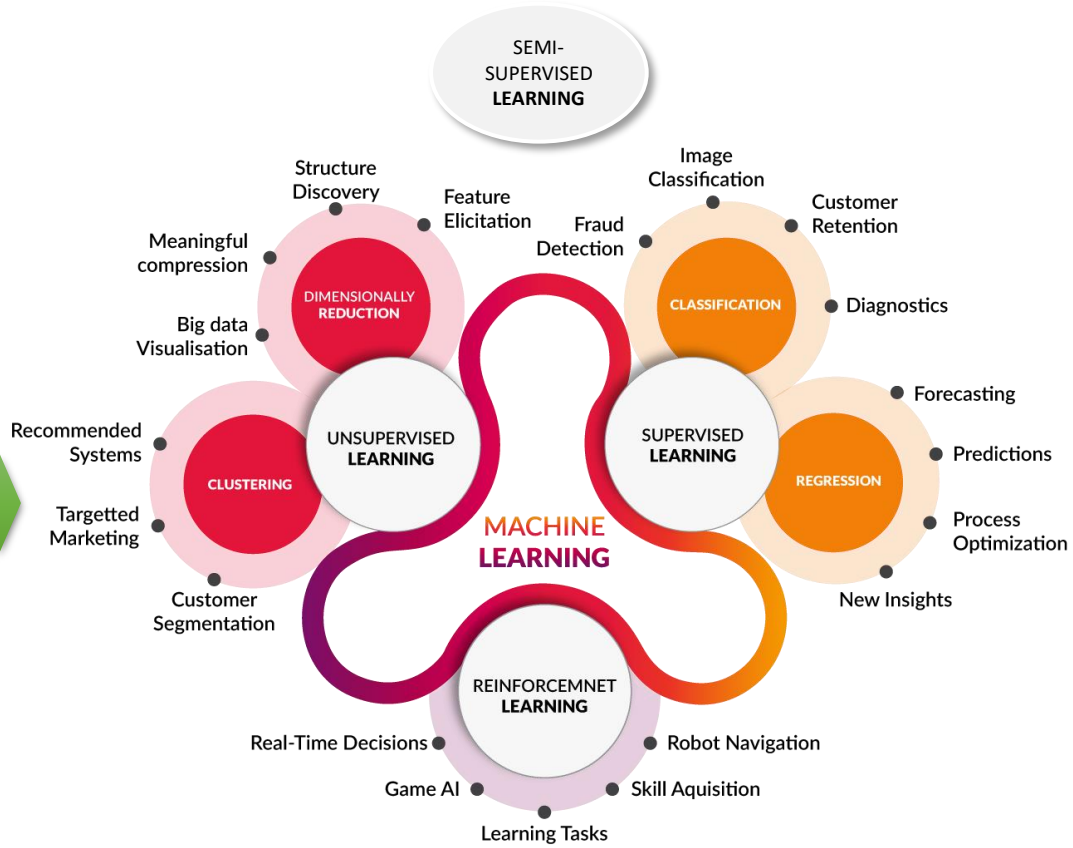
Aprendizado por indução



Definição de termos e escopo

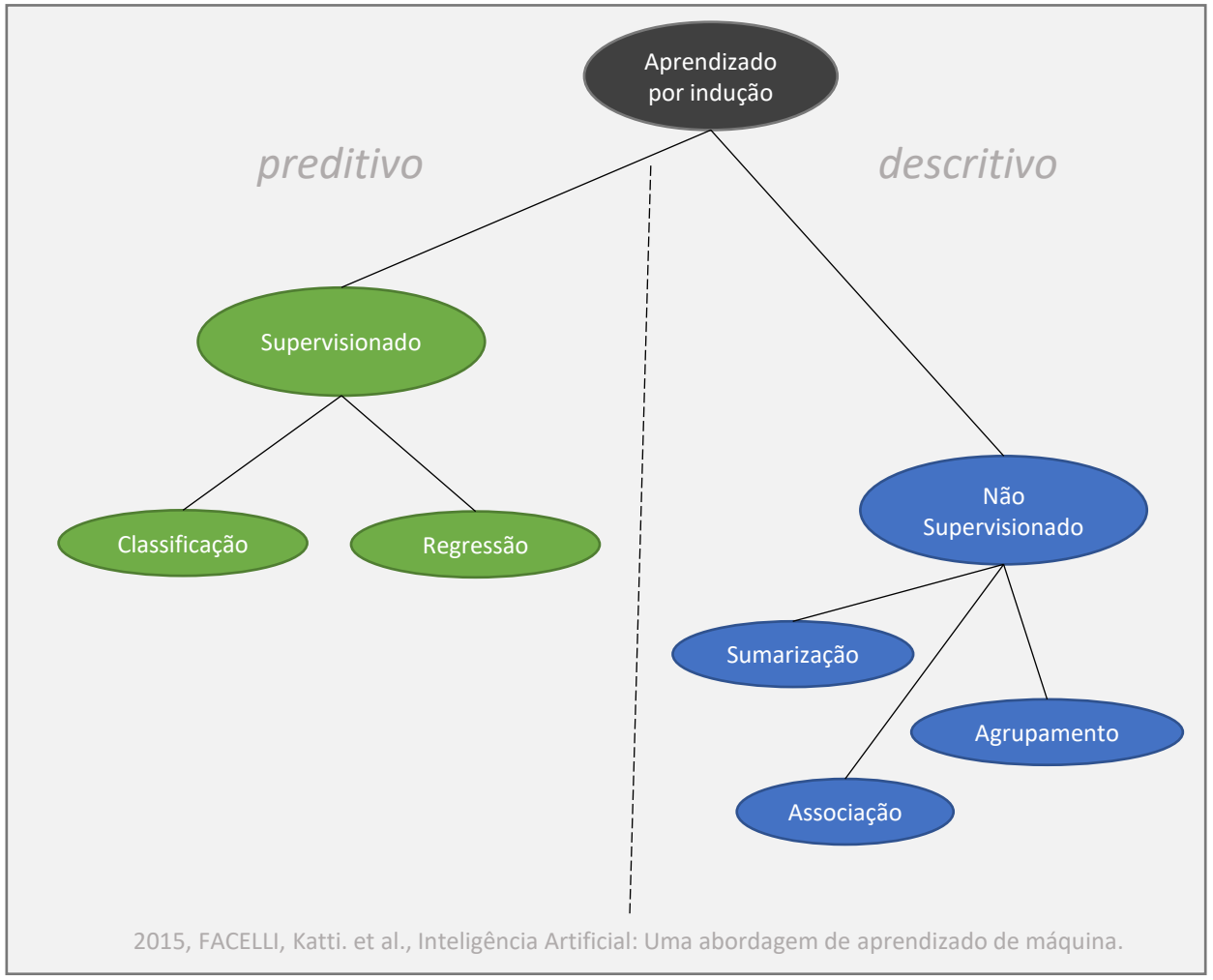
Inteligência Artificial > Aprendizado de Máquina > Deep Learning





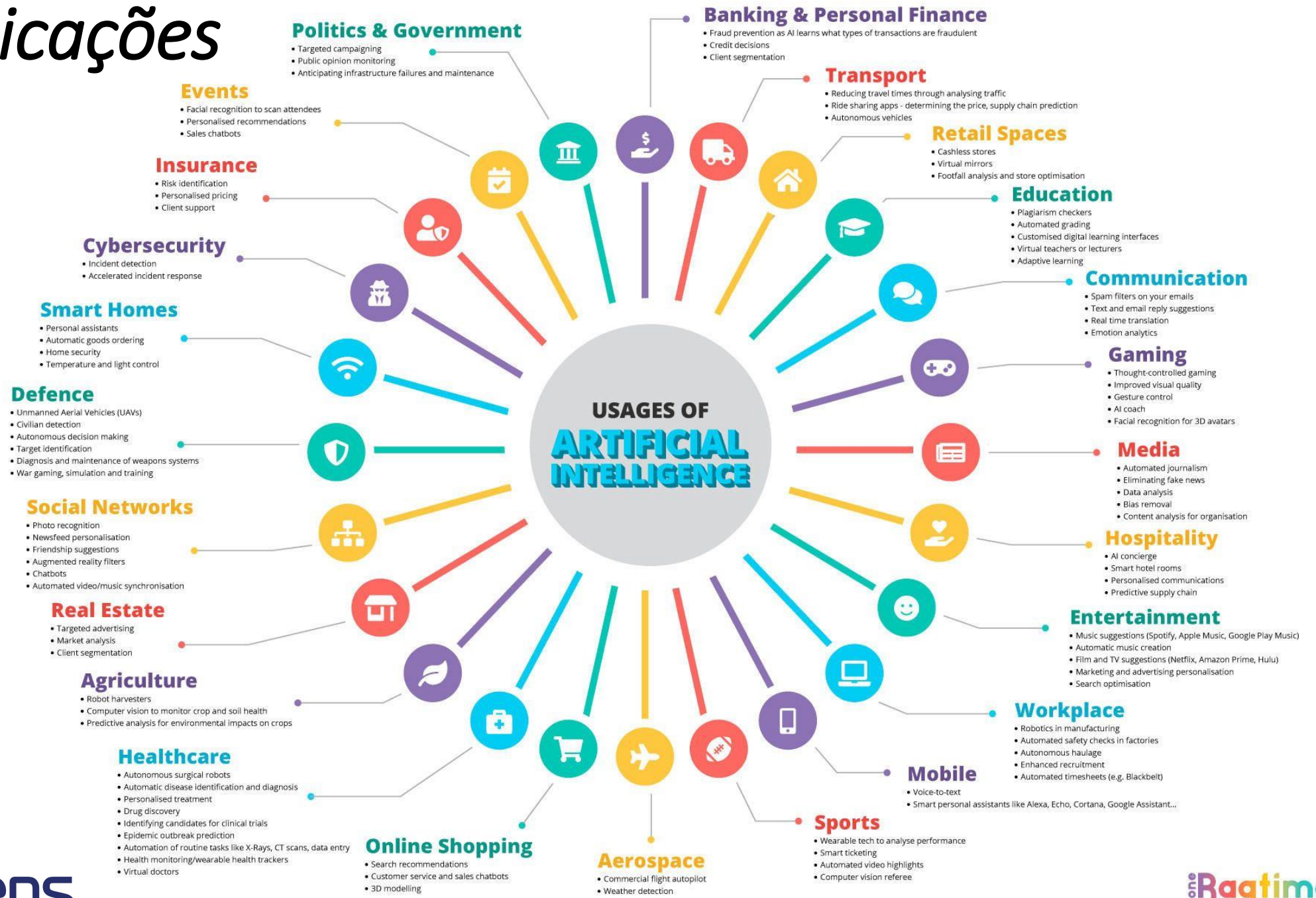
Aprendizado por indução

- Principais tarefas são:
  - Predizer (Apr. Supervisionado):
    - Diagnóstico de doenças.
    - Reconhecimento facial.
    - Predição de fraudes.
    - Predição de desistência (*churn*).
    - SAC: Qual será o volume de ligações?
    - Qual o preço justo de um imóvel?
    - Algum outro caso interessante?
  - Descrever (Apr. não Supervisionado):
    - Sistemas de recomendação.
    - Segmentação de clientes.
    - Agrupamento de documentos similares.
    - Algum outro exemplo?



2015, FACELLI, Katti. et al., Inteligência Artificial: Uma abordagem de aprendizado de máquina.

Aplicações



Facens

Ragtime

Aprendizado por indução

Formalização

- Aprendizado supervisionado:
  - São fornecidos o conjunto de atributos independentes ($\vec{X}$) e o atributo alvo (y). A partir dos exemplos o algoritmo gera um modelo, por *indução*, capaz predizer novas amostras por *dedução*.

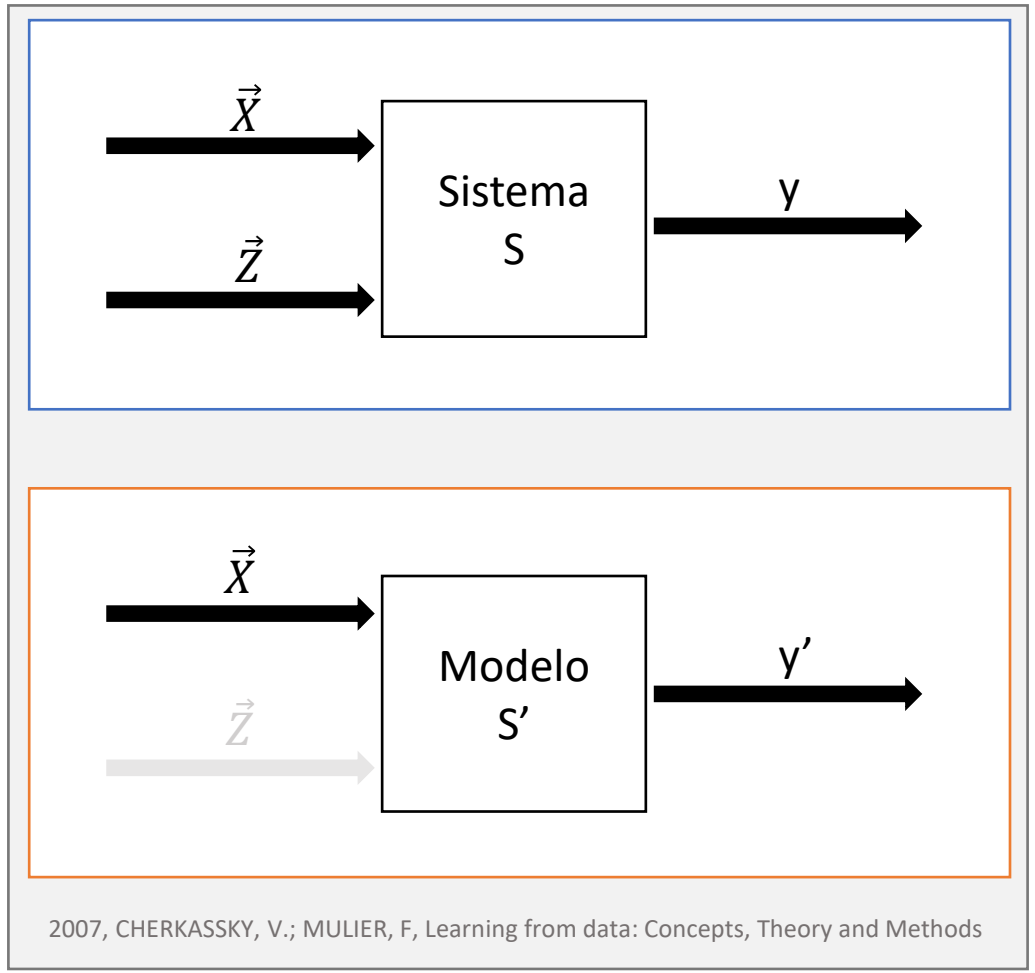
- Aprendizado não supervisionado:
  - É fornecido apenas o conjunto de atributos ($\vec{X}$). Os algoritmos buscam padrões descritivos nos dados através da análise das características dos exemplos e suas relações.

Aprendizado supervisionado

Formalização

- Existe um sistema S, com entradas observadas $\vec{X}$ e entradas não observadas $\vec{Z}$, que produz os resultados y baseado nos inputs.

- Existe um modelo S', cuja função busca imitar o comportamento do Sistema S, mapeando o conjunto de entradas $\vec{X}$ a uma saída y'.



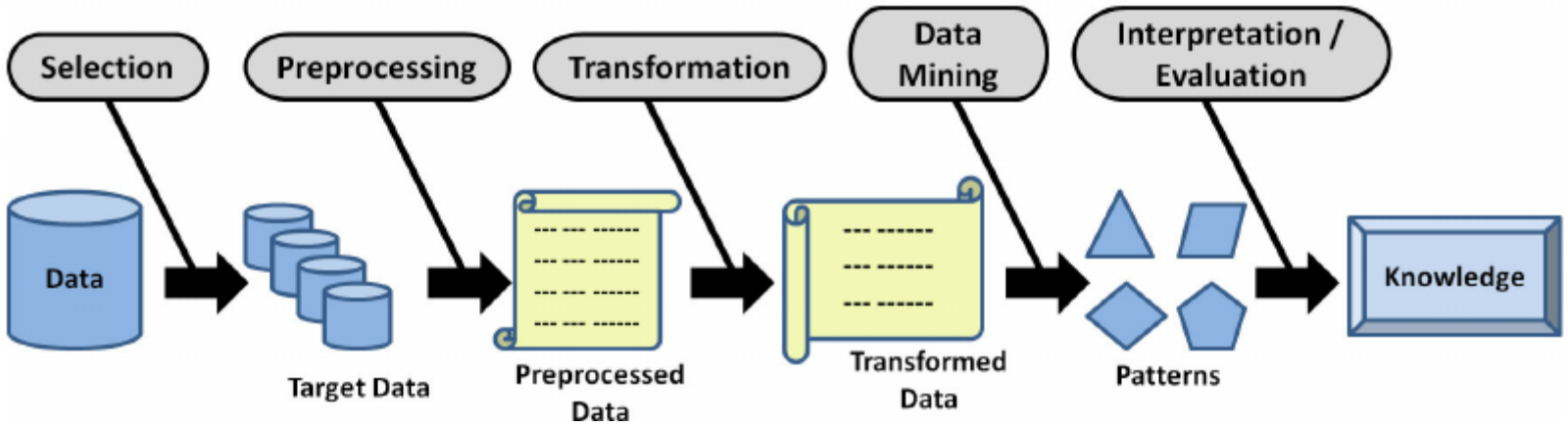
2007, CHERKASSKY, V.; MULIER, F, Learning from data: Concepts, Theory and Methods

Preparação:
Pré-Processamento

Juvenal J. Duarte

Extração de Conhecimento

Knowledge Discovery in Databases (KDD)



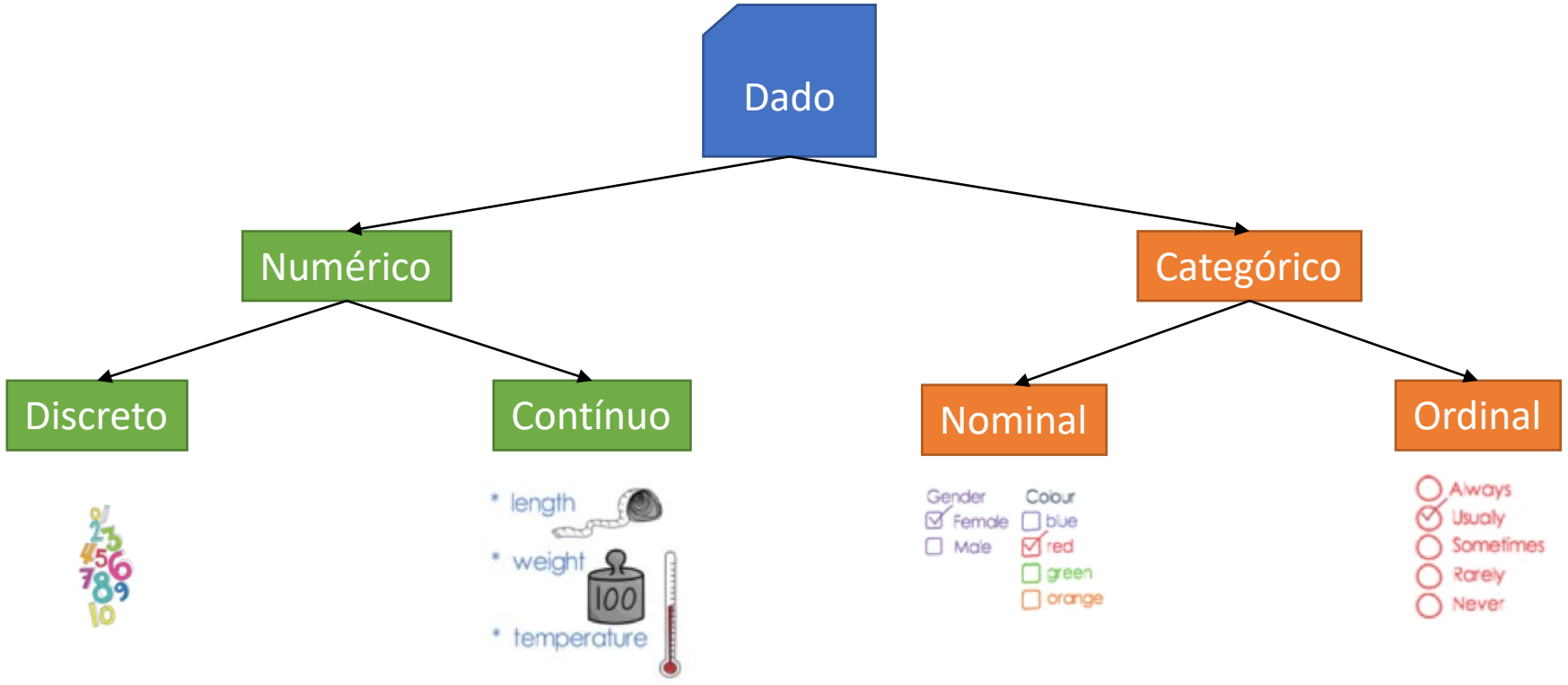
1996, U. Fayyad, From Data Mining to Knowledge Discovery in Databases

Extração de Conhecimento

Por onde começar?

Análise Exploratória

Tipos de dados



Análise Exploratória: Numérico

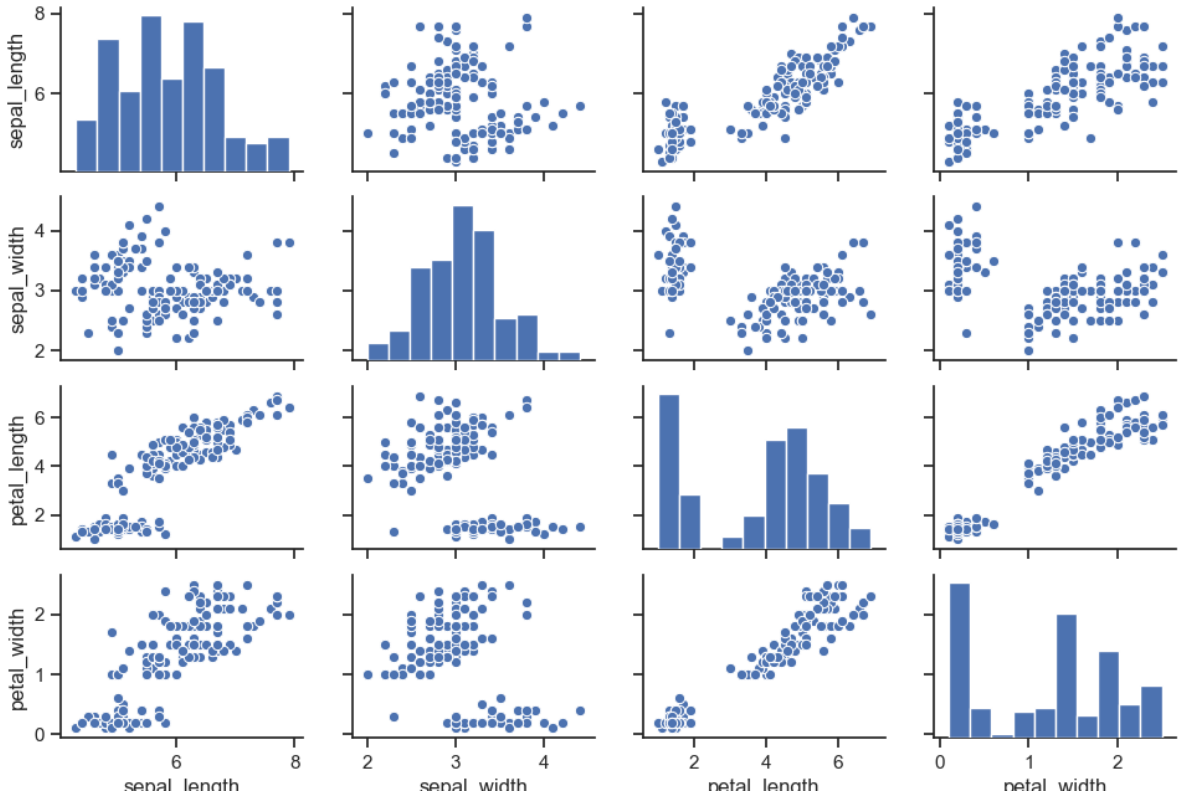
EDA: Entendendo a relação entre atributos com Seaborn

```
#!/usr/bin/python3

import seaborn as sns

sns.set(style="ticks", color_codes=True)
iris = sns.load_dataset("iris")
g = sns.pairplot(iris)

import matplotlib.pyplot as plt
plt.show()
```



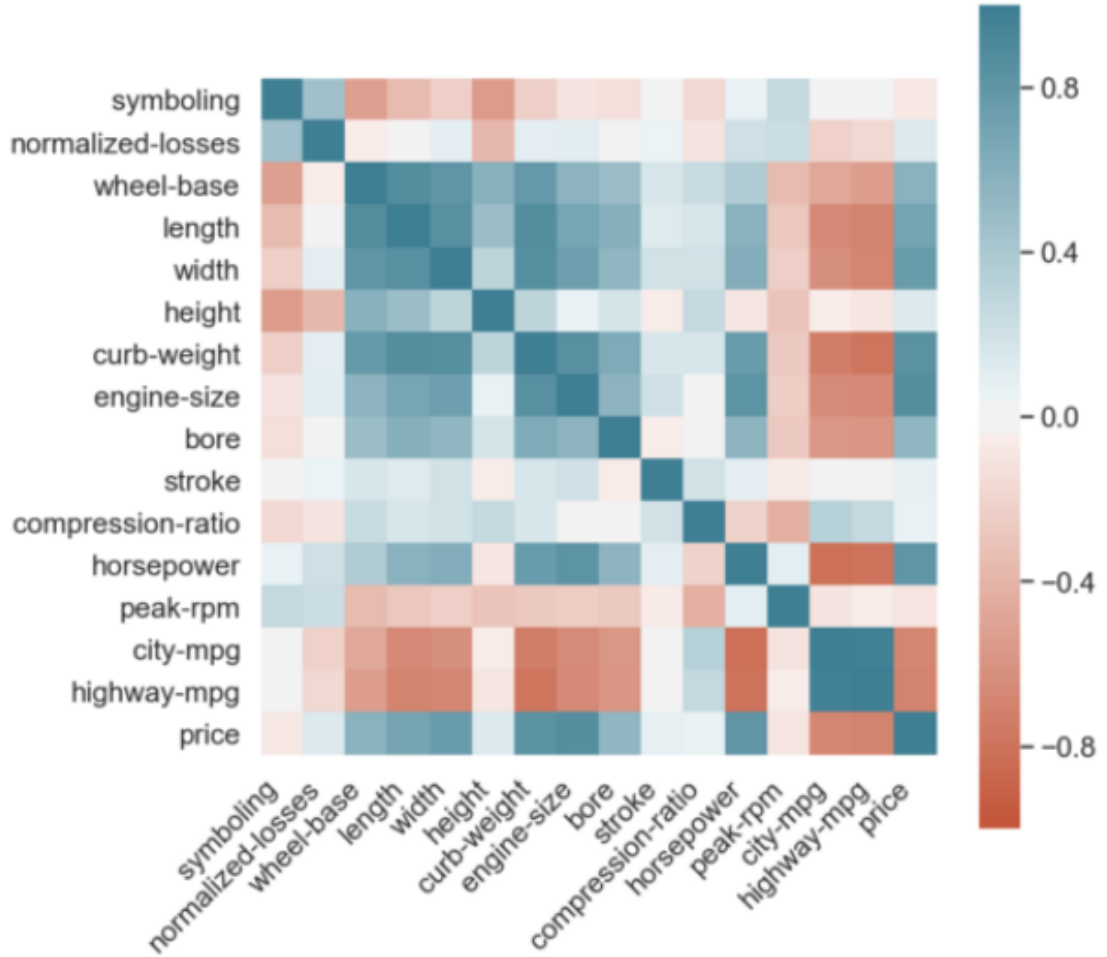
Análise Exploratória: Numérico

EDA: Correlação entre atributos com Seaborn

```
#!/usr/bin/python3

import seaborn as sns

ax = sns.heatmap(
    df.corr(),
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True
)
```



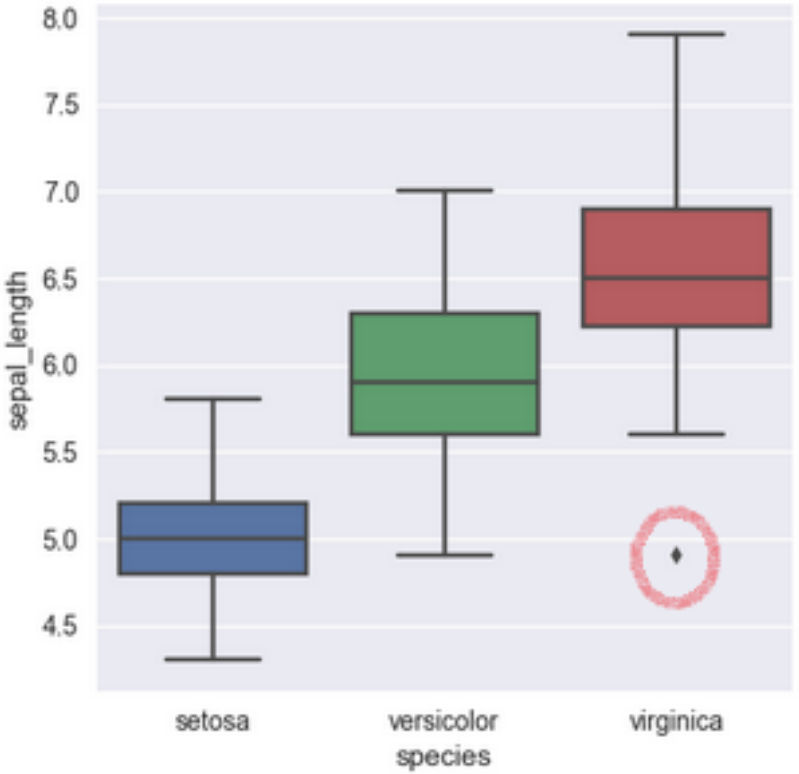
Análise Exploratória: Numérico

EDA: Visualizando outliers com Seaborn

```
#!/usr/bin/python3

import seaborn as sns

df = sns.load_dataset('iris')
sns.boxplot(data=df.ix[:,0:2])
sns.plt.show()
```



Análise Exploratória: Categórico

EDA: Cardinalidade de atributos catégóricos


```
for c in df.columns:
    c_domains = df[c].unique()

    if len(c_domains) > 30:
        print("%s: %d distinct" %(c, len(c_domains)))
    else:
        print("%s: %s" %(c, c_domains))
```

```
branch_id: [0]
customer_code: 838 distinct
group_code: [0 2 1 3]
item_code: 2981 distinct
item_total_price: 55336 distinct
order_id: 24618 distinct
quantity: 290 distinct
register_date: 3121 distinct
sales_channel: 106 distinct
segment_code: [0 2 4 5 1 3 6 7]
seller_code: 290 distinct
total_price: 22713 distinct
unit_price: 19828 distinct
```

Extração de Conhecimento

Vamos botar a mão na massa?

Pre-Processamento

Pandas: manipulação básica

1. Verificar colunas e tipos de dados:

```
df.dtypes
```

```
date_col          datetime64[ns]
company                   object
rank                       int64
day                       object
month                     object
year                      object
revenues                   int64
```

2. Verificar estatísticas:

```
In [28]: #describe neutral quick statistics
         X_neutral_data.describe()
```

Out[28]:

	Theta	Alpha	LowBeta	HighBeta	Gamma	AllAddedUp	y
count	1786.000000	1786.000000	1786.000000	1786.000000	1786.000000	1786.000000	1786.0
mean	9.351044	4.411534	3.703502	5.730875	10.823143	34.020099	0.0
std	8.934447	2.267715	1.905130	6.372256	16.540386	25.378729	0.0
min	0.808450	0.767048	0.544357	0.645059	0.638017	5.206910	0.0
25%	3.563850	2.571570	2.074190	2.514865	2.289160	16.642800	0.0
50%	5.859970	4.231920	3.620685	3.680320	3.251600	23.705000	0.0
75%	11.915900	5.754865	4.988218	5.570217	9.292220	43.873750	0.0
max	50.100100	15.010100	10.340500	45.044600	68.949900	110.906000	0.0

Pre-Processamento

Pandas: manipulação básica

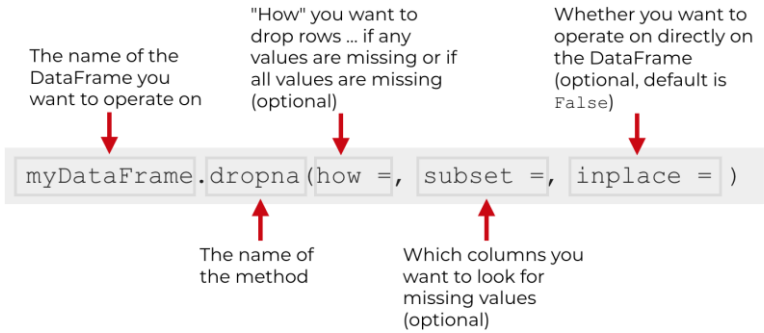
3. Identificar valores nulos/ não nulos em uma coluna:

```
In [8]: df[~df['val'].isna()]
```

Out[8]:

	key	val
1	a	456.0
3	c	32.0

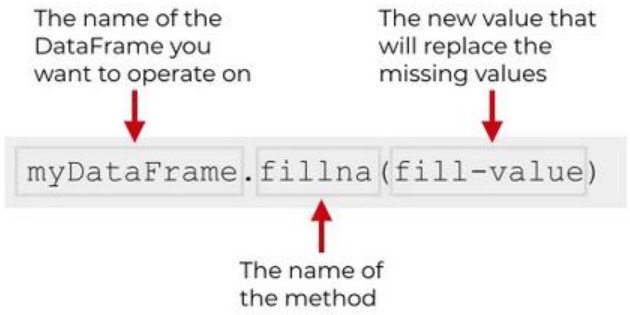
4. Excluir linhas/ colunas com valores nulos:



Pre-Processamento

Pandas: manipulação básica

5. Substituir nulos:



6. Remover linhas duplicadas:

```
data.drop_duplicates(keep='first', inplace=True)
```

```
data.head()
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Children
1	2	Moses(1990)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
4	4	MoTech(2019)	Comedy|Drama|Romance
6	5	Father of the Bride Part II (1995)	Comedy

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Seleção de Modelo

Juvenal J. Duarte

Seleção de modelo

Parâmetro $\neq$ Hiperparâmetros :

- Parâmetros são usados diretamente na função de decisão/ regressão. Ex.: β e α da Regressão Linear
- Hiperparâmetros são variáveis usadas na calibração do modelo. Ex.: learning rate, epochs.

Como saber se um modelo é "bom"?

- Os parâmetros do modelo levam a predições similares ao sistema real ($y \cong \hat{y}$)

E como saber se os parâmetros obtidos são "bons"?

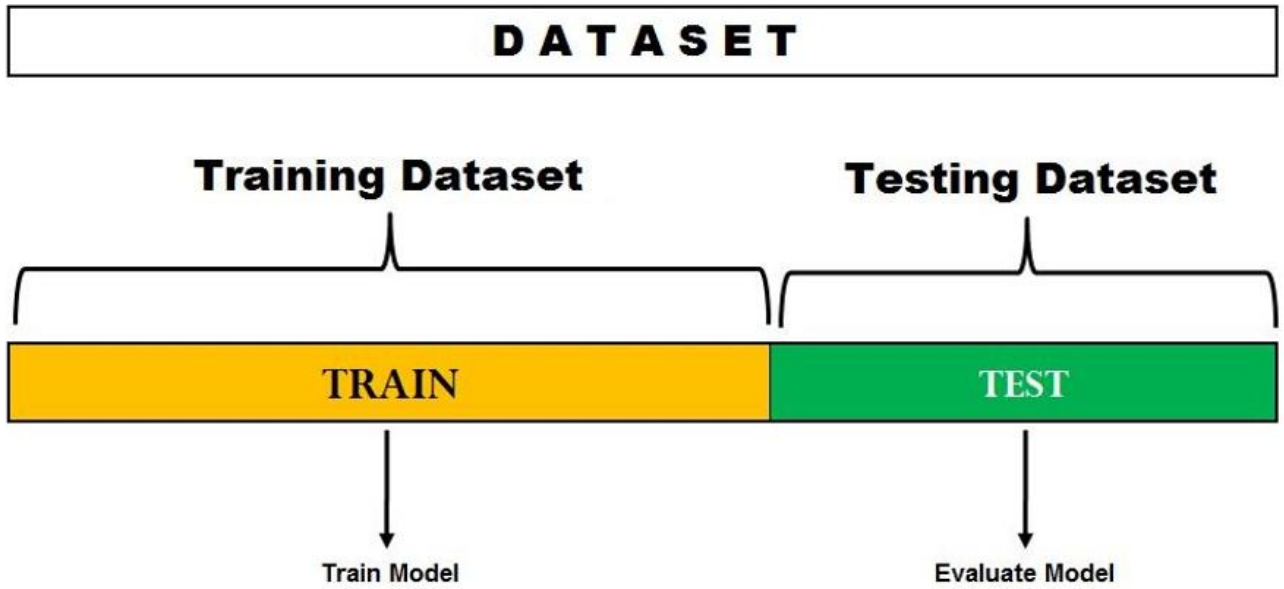
- Separe parte dos seus dados para teste!

Caso o modelo não mostre resultados satisfatório no teste, ajuste os hiperparâmetros, os dados, a metodolgia etc.

Separação de Dados

Hold Out

- Divisão treino/teste: método Hold-Out
  - Separa uma porção dos dados para calibrar o modelo, outra para testar sua precisão.
  - Normalmente é definida uma porcentagem do dataset para testes.
  - Porcentagens comuns são 70/30 e 80/20, mas pode variar muito dependendo da quantidade de registros!



Separação de Dados

Hold Out

Variações importantes:

- Amostragem aleatória (shuffle): Evita que registros “mais difíceis” se concentrem somente no grupo de treino ou de teste.

- Amostragem estratificada (stratified): Para problemas de classificação, garante que a distribuição de exemplos de cada classe seja igual ou parecida entre as porções de treino e teste.

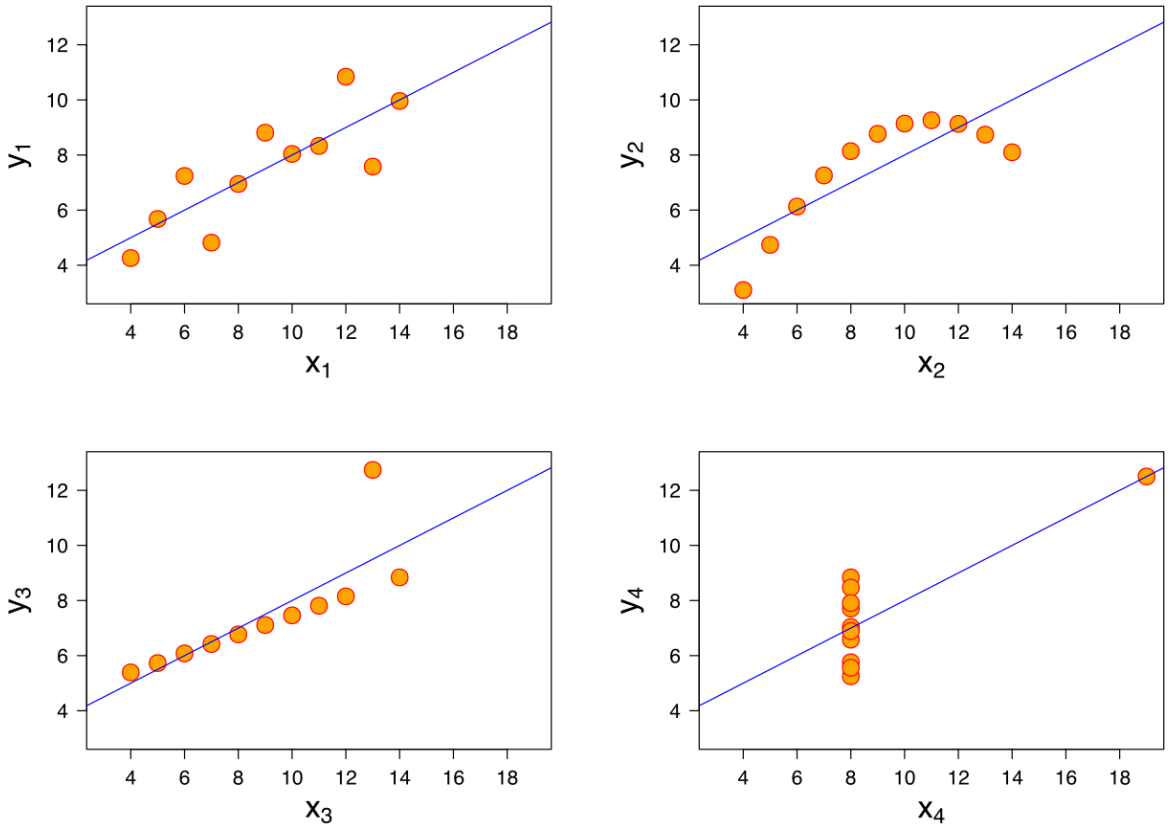
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Mineração de padrões:
Regressão Linear

Juvenal J. Duarte

Regressão linear

Investiga relações lineares entre variáveis. Qual das relações abaixo apresenta relação mais linear?



Wikipedia:
https://en.wikipedia.org/wiki/Linear_regression

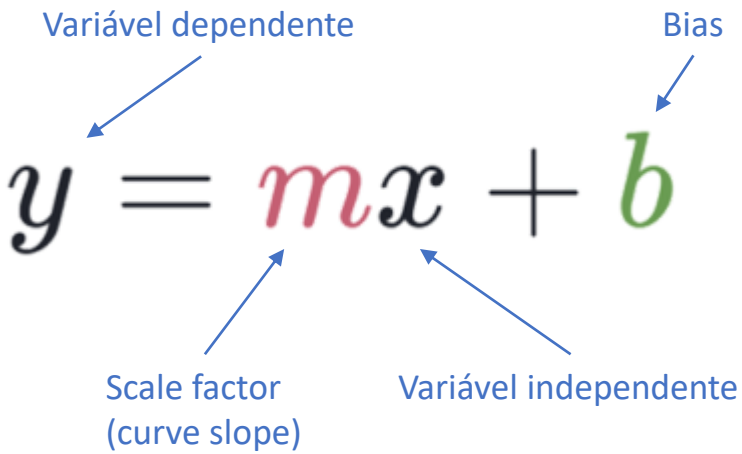
Regressão linear univariada

O que é uma relação de dependência linear?

$$y = mx + b$$

Regressão linear univariada

O que é uma relação de dependência linear?



Regressão linear multivariada

Como generalizer a formula para o caso de n atributos?

$$y = m_1 x_1 + m_2 x_2 + m_3 x_3 + \cdots + m_n x_n + b$$

$$y = b + \sum_{i=1}^{n} x_i m_i$$

Regressão linear multivariada

Exemplos

$$Colheita = m_1 Temperatura + m_2 Chuva + m_3 Área + m_4 Fertilizantes + \ldots + b$$

$$Preço\ Dinâmico = m_1 Localização + m_2 Custo + m_3 Demanda + m_4 Logística + \cdots + b$$

$$Previsão\ de\ Demanda = m_1 DemandasAnteriores + m_2 Usuários + m_3 Sazonalidade + \cdots + b$$

$$Credit\ Score = m_1 Renda + m_2 Gastos + m_3 Escolaridade + \cdots + b$$

Regressão linear multivariada

Formalização de parâmetros

Quantidade de linhas / registros / amostras: m

Quantidade de colunas / atributos / characteristicas: n

Conjunto de entradas: $\vec{X}_{m,n} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}$

Conjunto de pesos: $\vec{B}_{n,1} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_n \end{bmatrix}$

Bias: α

(1) Predição (formula vetorial):

$$\hat{y}_{m,1} = \vec{X}_{m,n}\,\vec{B}_{n,1} + \alpha$$

Regressão linear: função de custo

Como estimar os hiper parâmetros e saber se o modelo é bom o suficiente?

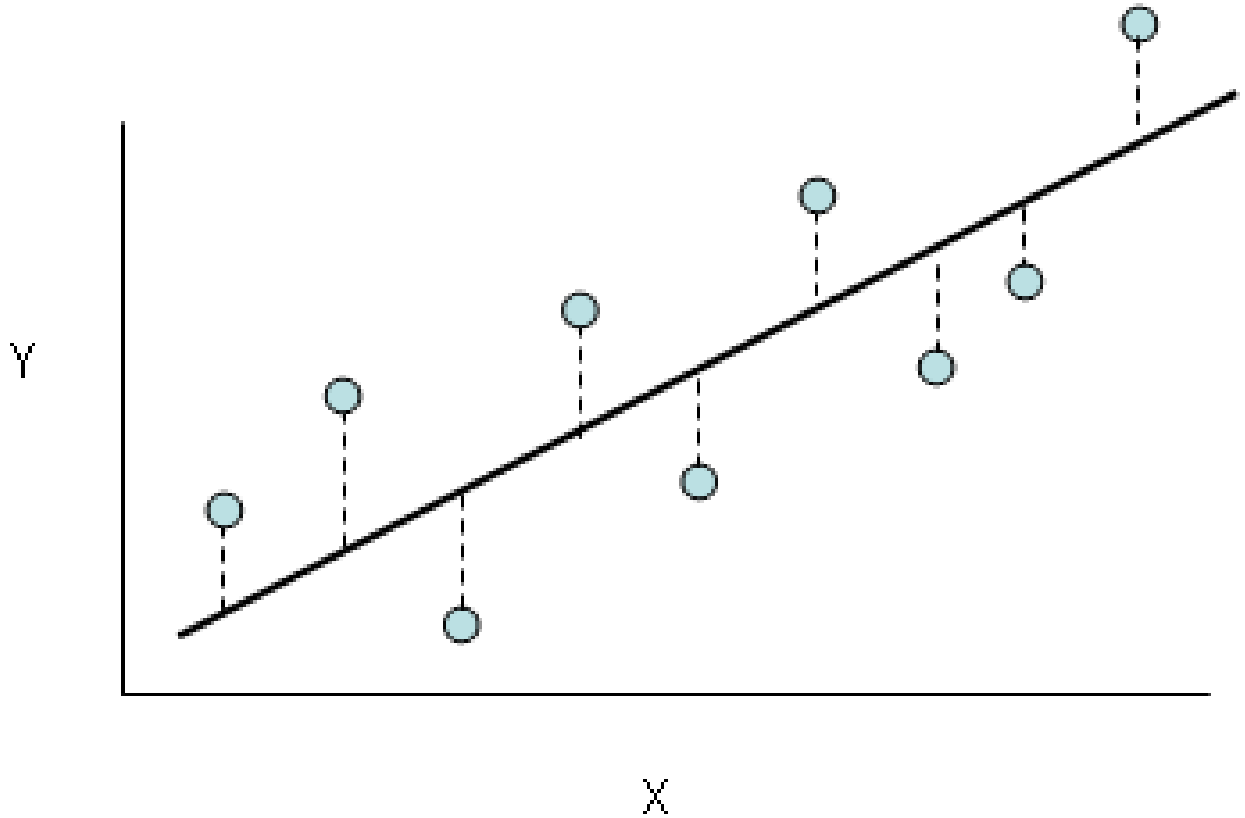
(2)

$$loss(i) = y_i - \hat{y}_i$$

Erro Quadrático Médio (MSE):

(3)

$$cost = J = \frac{1}{m}\sum_{i=}^{m} loss(y_i, \hat{y}_i)^2$$



Regressão linear: otimização

Como estimar os melhores hiper-parâmetros? Métodos de otimização:

- Gradiente Descendente:
    - Funciona bem para qualquer número de atributos.
    - Iterativo.
    - Roda tão rápido quanto os parâmetros escolhidos (learning rate, epochs).
    - É tão preciso quanto os parâmetros escolhidos.

- Resolução de equações lineares:
    - Método analítico, não precisa de iterações ou taxa de aprendizado.
    - O tempo de execução é $O(n^3)$. Torna-se inviável para datasets “wide”.

Regularização
L1 & L2

- A regularização atua na função de custo para impedir que o modelo se super-ajuste aos dados.

- Os nomes L1 e L2 tem origem na nomenclatura de norma de vetores:

  - L1 (Lasso) = $+ \boldsymbol{\lambda}|\boldsymbol{\beta}|$
  - L2 (Ridge) = $+ \boldsymbol{\lambda}\boldsymbol{\beta}^{\boldsymbol{2}}$

- A Regressão Linear com ambas regulari-Zações L1 e L2 é chamada **Elastic Net**.

Erro Quadrático Médio (MSE):

L1

$$cost = J = \frac{1}{m}\sum_{i=}^{m} loss(y_i, \hat{y}_i)^2 + \boldsymbol{\lambda}|\boldsymbol{\beta}|$$

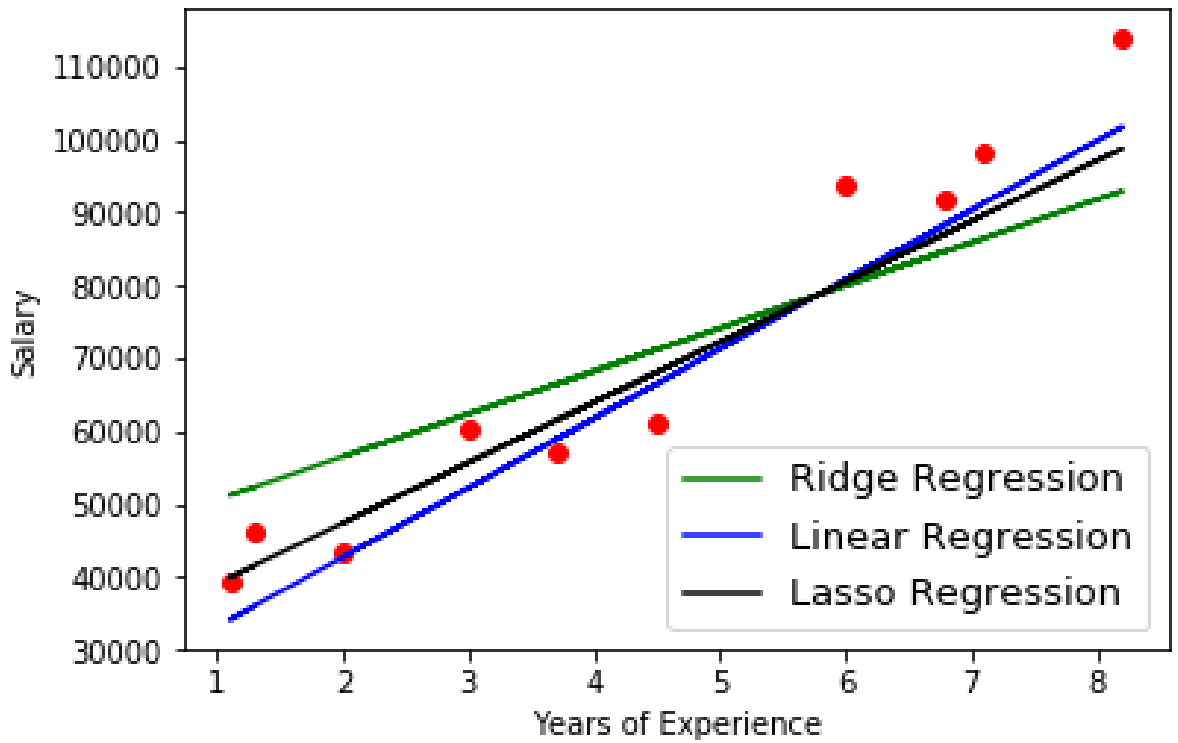
L2

$$cost = J = \frac{1}{m}\sum_{i=}^{m} loss(y_i, \hat{y}_i)^2 + \boldsymbol{\lambda}\boldsymbol{\beta}^{\boldsymbol{2}}$$

Ridge & Lasso

Comparativo

Como estimar os parâmetros e saber se o modelo é bom o suficiente?



Regularização
L1 & L2: Comparativo

- L1 é mais adequada no tratamento de dados esparços.

- A regularização L1 possui a característica de também funcionar como seleção de atributos, principalmente em casos de multicolinearidade.

- Por costumar anular o peso de atributos, L1 muitas vezes leva a pioras no viés.

- L2 é a melhor escolha para maioria dos casos, com evidências teóricas e práticas.

Mineração de padrões:
Avaliação de modelos

Juvenal J. Duarte

Regressão

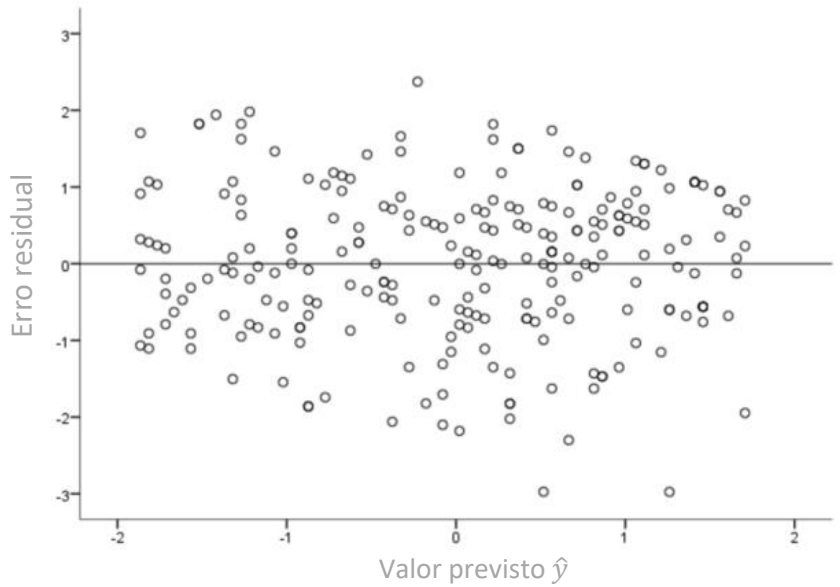
Métricas de avaliação

- **Mean Squared Error (MSE):** Trata bem erros pequenos, mas é mais sensível a outliers.
  - $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
- **Root Mean Squared Error (RMSE):** MSE pode assumir valores muito grandes, RMSE facilita a análise e comparação
  - $RMSE = \sqrt[2]{MSE}$
- **Mean Absolute Error (MAE):** Menos sensível a outliers, mas é mais ameno na penalização de erros.
  - $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

Regressão

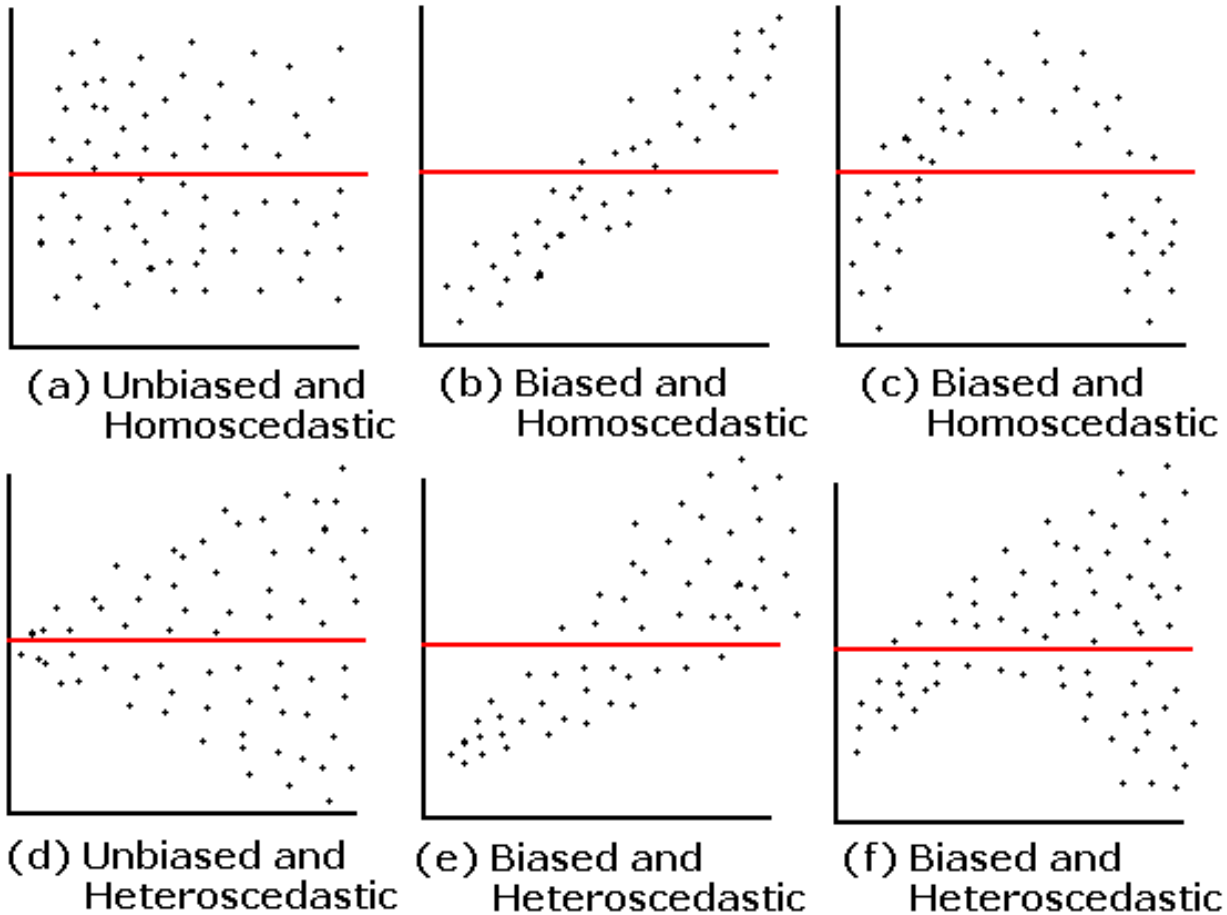
Métricas de avaliação

- Métricas de avaliação fornencem indícios do desempenho do modelo como um todo.
- Na prática, seu modelo pode estar indo muito bem para alguns exemplos e muito mal para outros.
  - Análise dos erros individuais para cada registro.
  - Uso de gráficos residuais:



Regressão

Plots Residuais: Exemplos



Mineração de padrões:
Árvore de Regressão

Juvenal J. Duarte

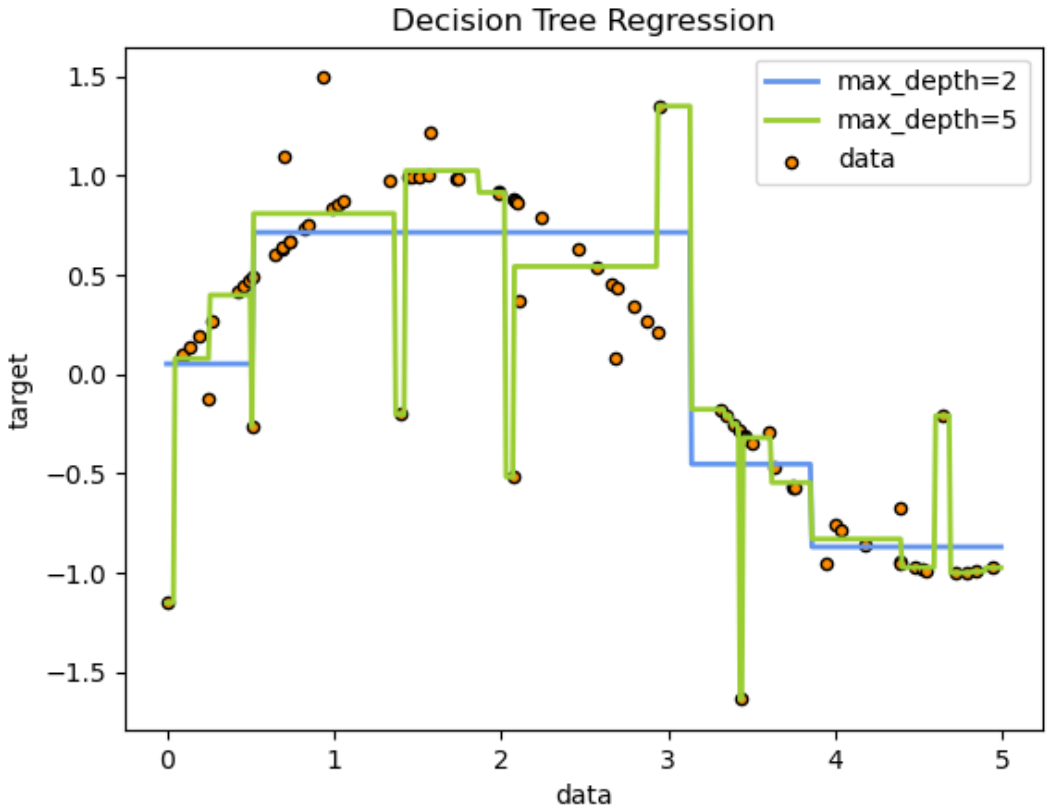
Regressão

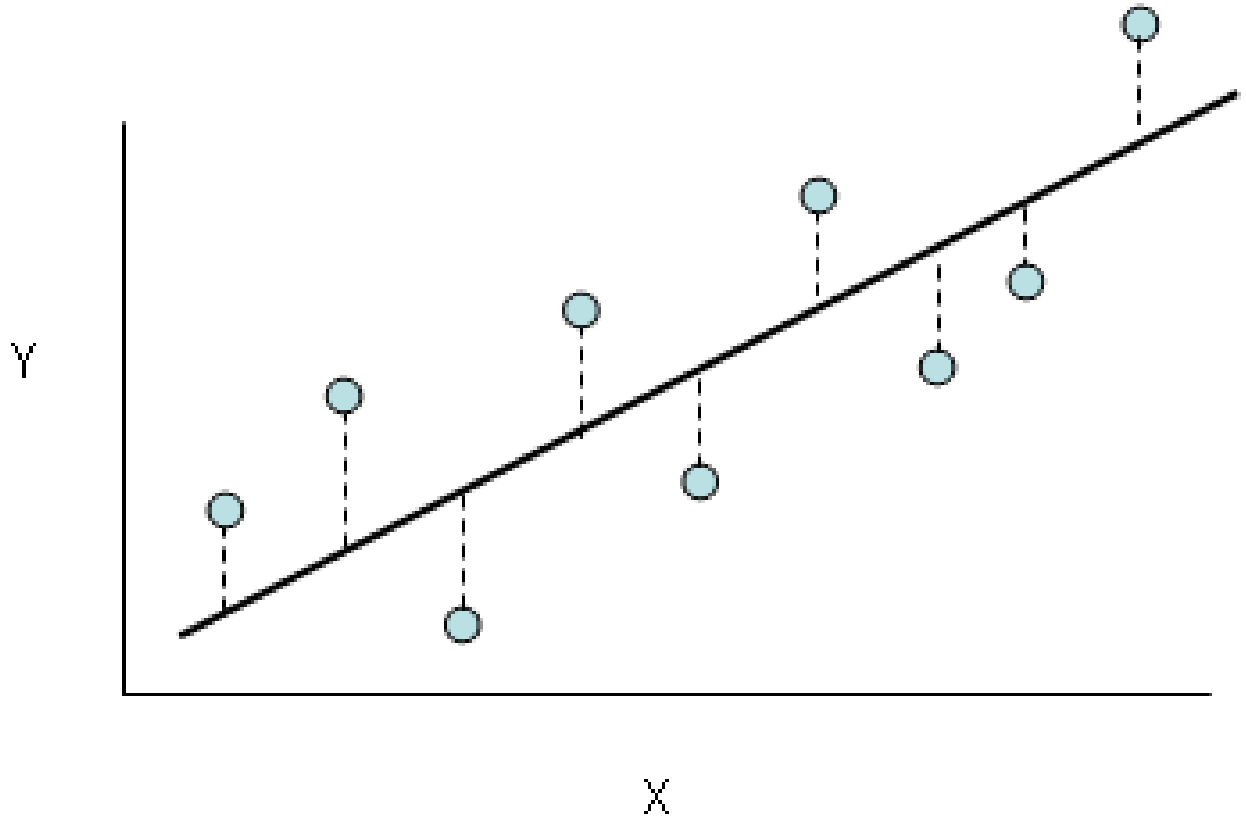
Árvores de regressão

- Método alternativo para o tratamento de relações não lineares.
- Modelo construido através do encadeamento de regras. As saídas são “menos contínuas” que na regressão linear.

Regressão: Comparativo

Regressão Linear & Árvores de Regressão





Regressão:
Exercício prático

Juvenal J. Duarte

Exercício de Regressão:

Desempenho de alunos

Descrição do dataset em:

https://archive.ics.uci.edu/ml/datasets/student+performance

Como ler os dados:

```
df_m = pd.read_csv("https://raw.githubusercontent.com/brvnl/MultivariateLinearRegression/master/student/student-mat.csv",
                   sep=";")

df_p = pd.read_csv("https://raw.githubusercontent.com/brvnl/MultivariateLinearRegression/master/student/student-por.csv",
                   sep=";")
```

Exercício de Regressão:

Desempenho de alunos

Sugestão de bibliotecas:

Regressão Linear:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Elastic Net:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Árvores de Regressão:

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

Separação de dados:

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Exercício de Regressão:

Desempenho de alunos

Modelos avançados:

Florestas Aleatórias:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Multilayer Perceptron (Rede Neural):

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html