



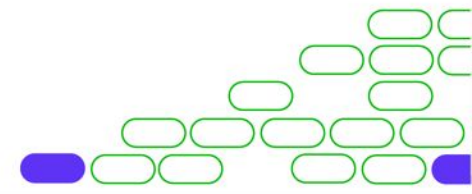
Faculdade



Introdução à Análise de Dados

**CAPÍTULO 1. INTRODUÇÃO À ANÁLISE
DE DADOS**

PROF. MATHEUS MENDONÇA





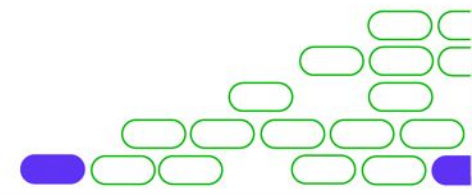
Faculdade



Introdução à Análise de Dados

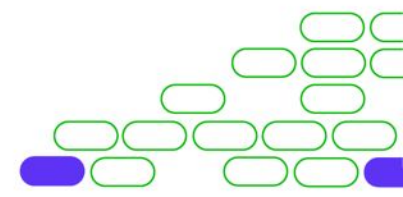
**AULA 1.1. APRESENTAÇÃO DA
DISCIPLINA**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Motivação.
- ❑ Tópicos abordados.





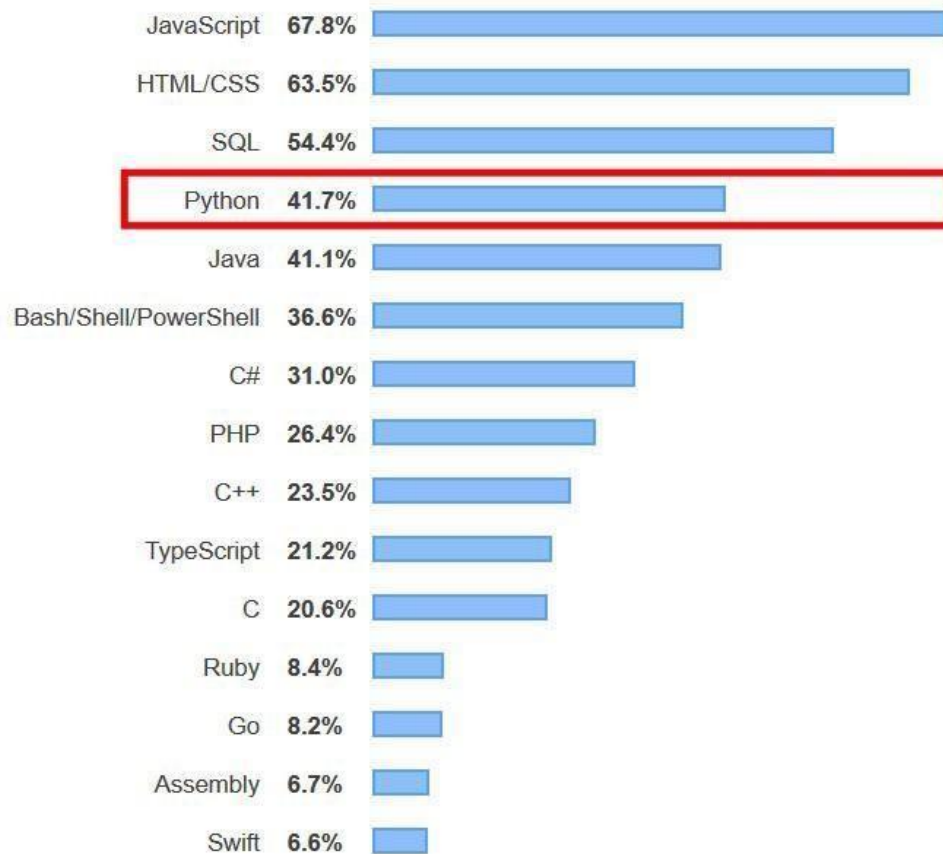
Most Popular Technologies



Programming, Scripting, and Markup Languages

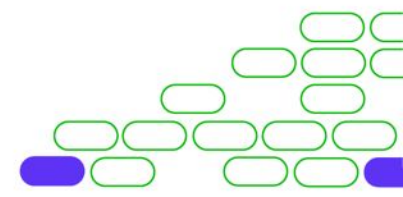
All Respondents

Professional Developers



Fonte:

<https://insights.stackoverflow.com/survey/2019#most-popular-technologies>





O sucesso do Python

- Open source.



O sucesso do Python

- Open source.
- Comunidade ativa.



O sucesso do Python

- Open source.
- Comunidade ativa.
- Diversas bibliotecas (também open source)
para análise/ciência de dados:

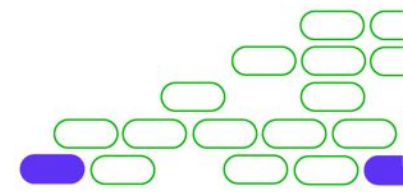


Neste módulo

NumPy

```
# cria um array de 2 dimensões: matrix 3x3  
a = np.array([[1, 2, 3], [2, 3, 4], [3, 4, 5]])  
print("Array criado:\n", a)  
print("shape:", a.shape)
```

```
Array criado:  
[[1 2 3]  
 [2 3 4]  
 [3 4 5]]  
shape: (3, 3)
```

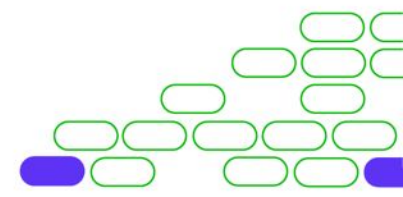


Neste módulo



```
# leitura dos dados  
df = pd.read_csv("https://pycourse.s3.amazonaws.com/temperature.csv")  
  
# visualizando as primeiras 3 linhas  
df.head(3)
```

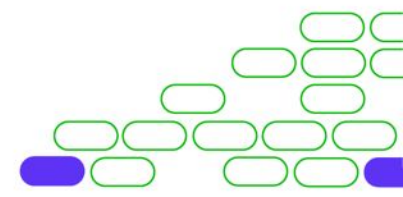
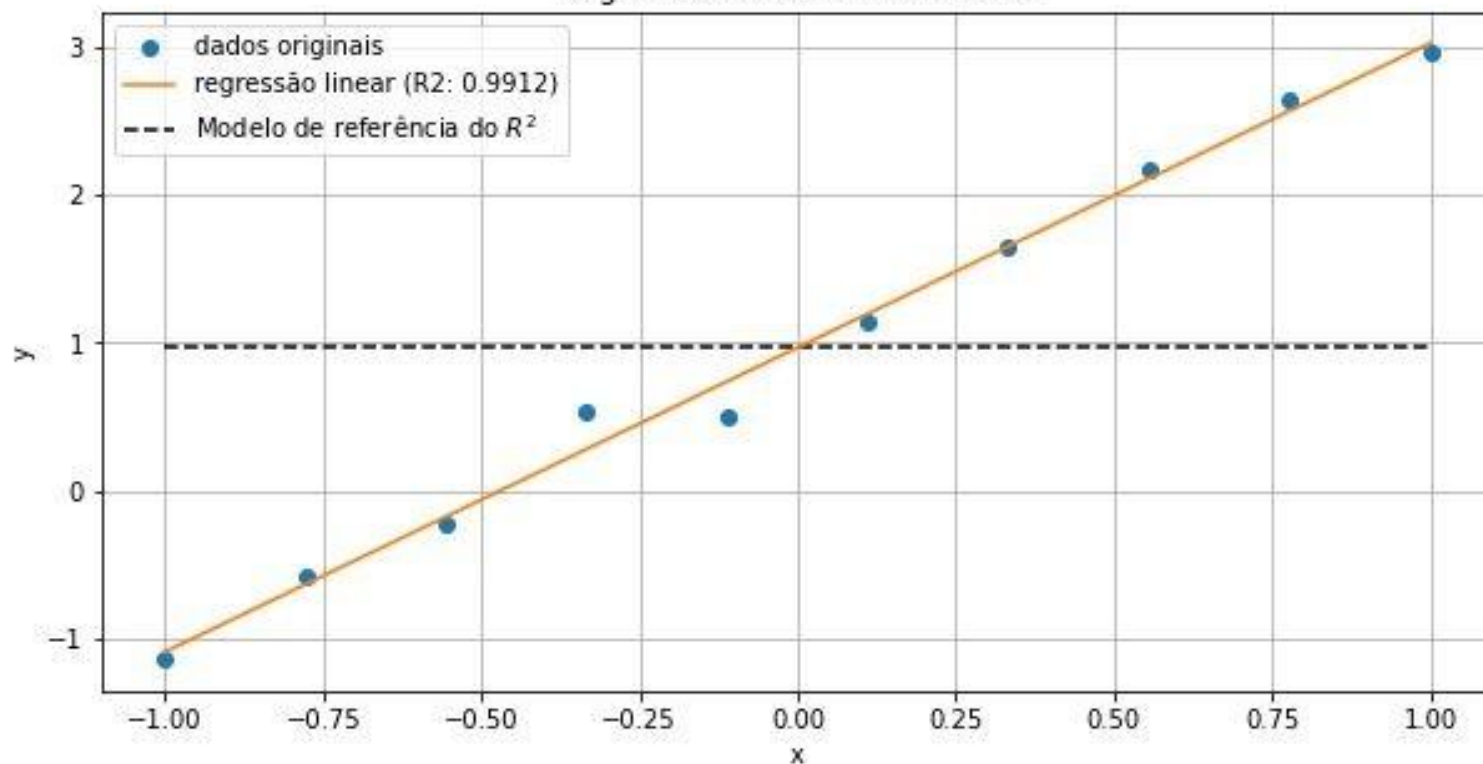
	date	temperatura	classification
0	2020-01-01	29.1	quente
1	2020-02-01	31.2	muito quente
2	2020-03-01	28.5	quente



Neste módulo



Regressão linear no scikit-learn

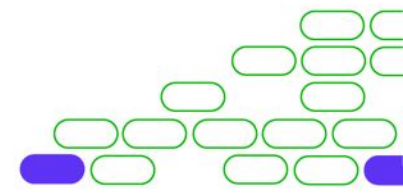


Conclusão

All you need is Python. Python is all you need.

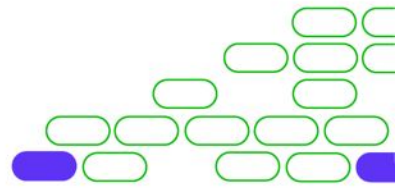


Fonte: <https://towardsdatascience.com/top-9-languages-for-data-science-in-2020-824239f930c>



Na próxima aula

- ❑ Introdução à análise de dados.





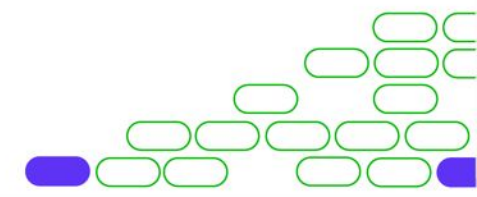
Faculdade



Introdução à Análise de Dados

**AULA 1.2. INTRODUÇÃO À ANÁLISE
DE DADOS**

PROF. MATHEUS MENDONÇA

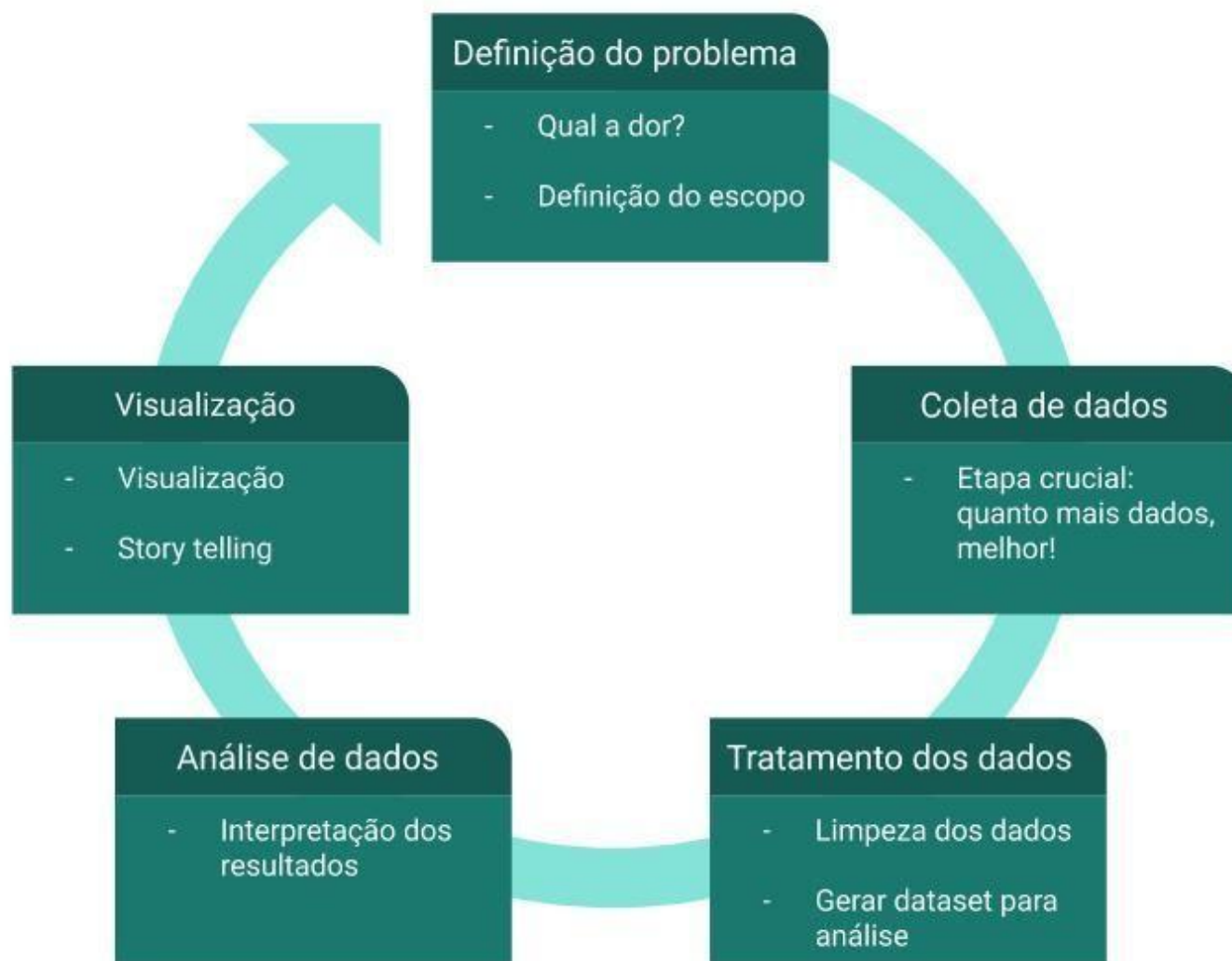


Nesta aula

- ❑ O que é a análise de dados?
- ❑ Análise de dados vs Ciência de dados.



A análise de dados



Análise de dados

- Cargos de um analista de dados:
 - Business analyst.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.
 - Tableau.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.
 - Tableau.
 - Power BI.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.
 - Tableau.
 - Power BI.
- Ferramentas recentes:
 - **Python.**

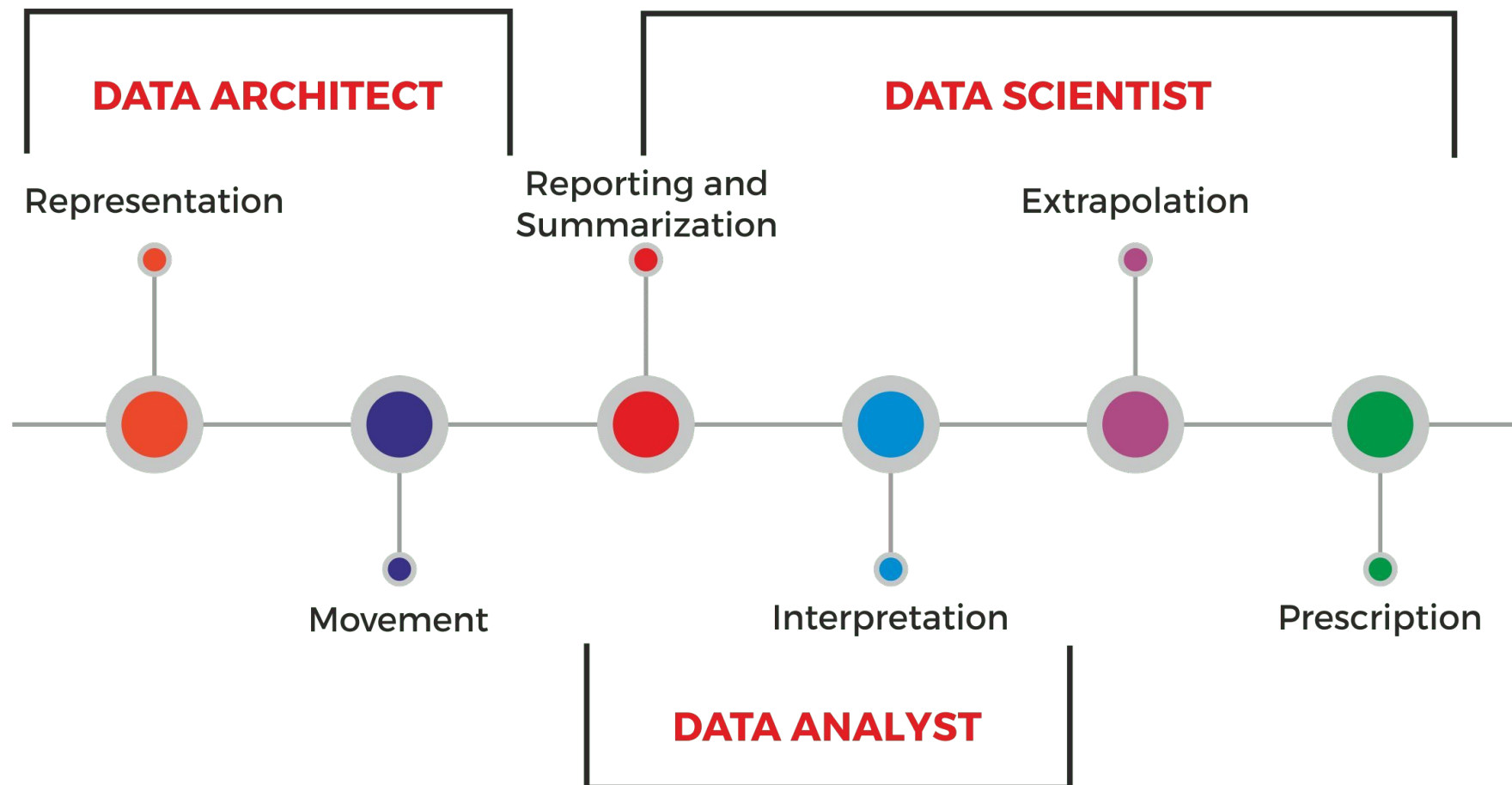
Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.
 - Tableau.
 - Power BI.
- Ferramentas recentes:
 - Python.
 - Pandas.

Análise de dados

- Cargos de um analista de dados:
 - Business analyst.
 - Business intelligence analyst (analista de BI).
- Ferramentas tradicionais:
 - SQL.
 - Excel.
 - Tableau.
 - Power BI.
- Ferramentas recentes:
 - Python.
 - Pandas.
 - Computação em nuvem;
 - Etc.

Análise de dados vs Ciência de Dados



Fonte:

<https://www.dasca.org/world-of-big-data/article/analysts-vs-scientists-the-big-data-puzzle>

Conclusão

- ✓ O analista de dados organiza e analisa os dados existentes para agregar conhecimento à tomada de decisão.
- ✓ O cientista de dados automatiza o processo de análise e cria modelos matemáticos capazes de extrapolar.



Próxima aula

- ❑ Numpy para análise de dados.





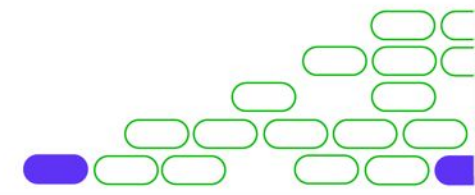
Faculdade



Introdução à Análise de Dados

**CAPÍTULO 2. NUMPY PARA A ANÁLISE
DE DADOS**

PROF. MATHEUS MENDONÇA





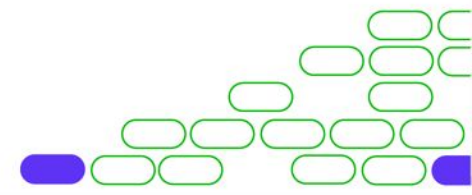
Faculdade



Introdução à Análise de Dados

**AULA 2.1. INTRODUÇÃO AOS
ARRAYS**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Introdução ao numpy.
- ❑ O que são numpy arrays?



Introdução ao numpy

- O [numpy](#) é uma das principais bibliotecas para computação científica em Python.
- Disponibiliza um objeto de array multidimensional de alta performance e diversas ferramentas para se trabalhar com esses objetos.
- Instalação:
 - `pip install numpy;`
 - `conda install numpy.`
- Importando no ambiente de desenvolvimento:
 - `import numpy as np.`

O que são numpy arrays?

- Estrutura de dados para manipulação e álgebra matricial:

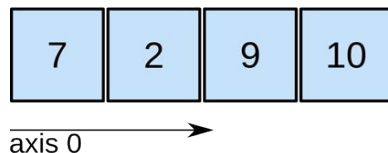
Index:	0	1	2	3	4
Value:	88	19	46	74	94



O que são numpy arrays?

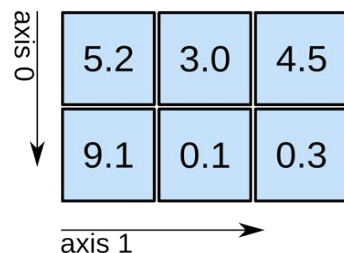
- Estrutura de dados para manipulação e álgebra matricial.
- Possibilita trabalhar com estruturas de dados n-dimensionais.

1D array



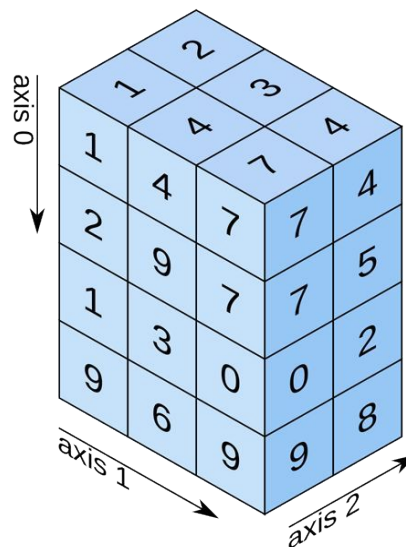
shape: (4,)

2D array



shape: (2, 3)

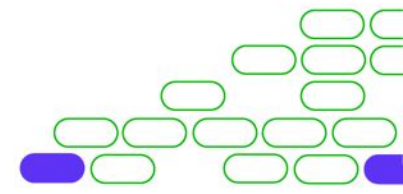
3D array



shape: (4, 3, 2)

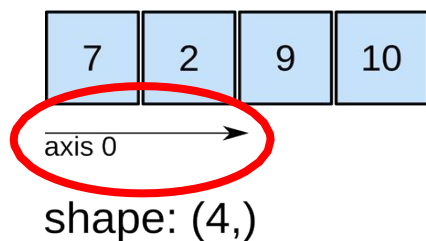
Fonte:

https://fapt.github.io/python_crashkurs_doc/include/numpy.html

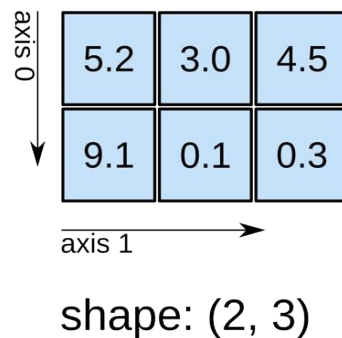


Numpy – axis

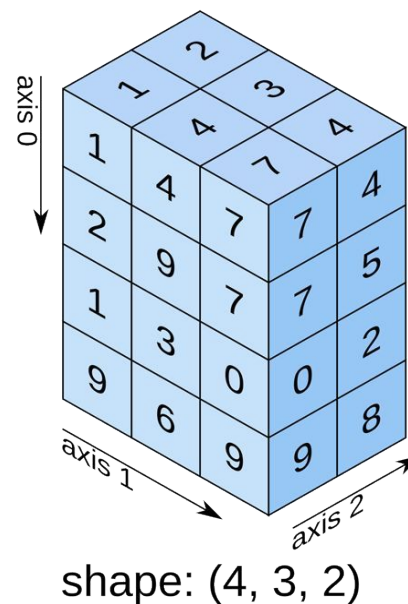
1D array



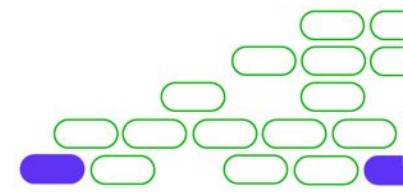
2D array



3D array



Caso 1D: direção ao longo das linhas.



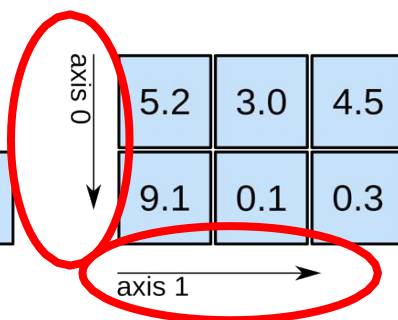
Numpy – axis

1D array



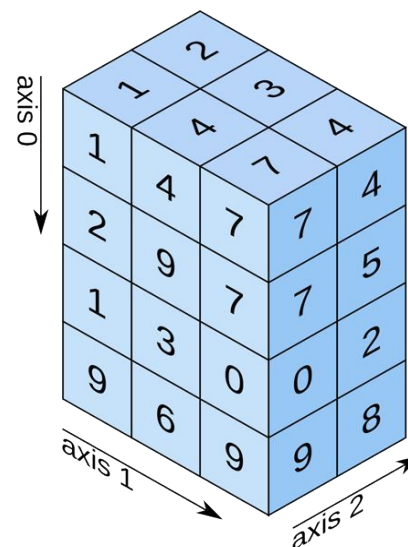
shape: (4,)

2D array



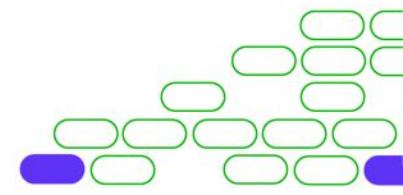
shape: (2, 3)

3D array



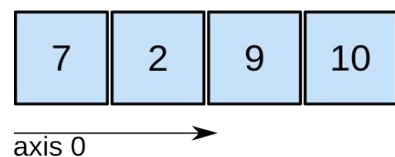
shape: (4, 3, 2)

Caso 2D: direção ao longo das linhas e colunas.



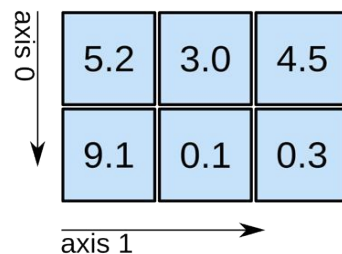
Numpy – axis

1D array



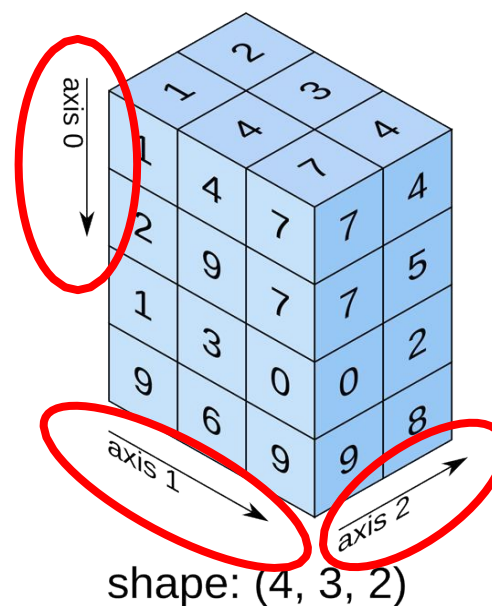
shape: (4,)

2D array

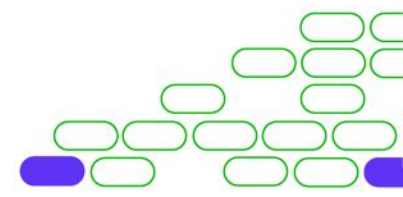


shape: (2, 3)

3D array

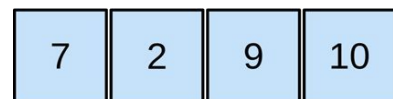


Caso 3D: direção ao longo dos eixos x, y e z. Exemplo: imagens.



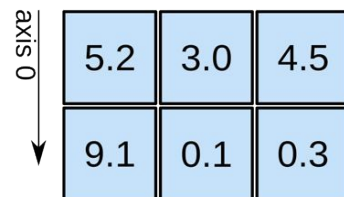
Numpy – shape

1D array



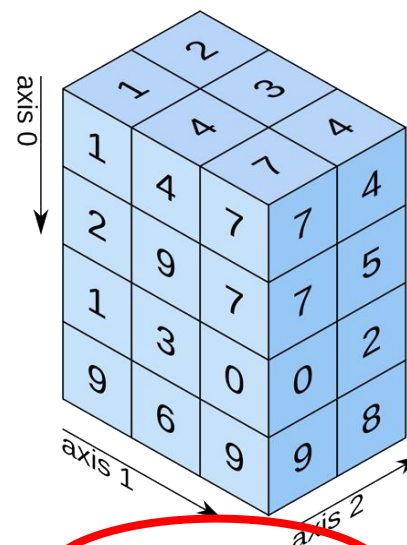
shape: (4,)

2D array



shape: (2, 3)

3D array



shape: (4, 3, 2)

Quantidade de elementos em cada eixo.

Numpy performance

Comparativo do tempo de execução de um algoritmo de machine learning implementado com **Python puro** e implementado com o **Numpy**:

Implementation	Elapsed Time
Pure Python with list comprehensions	18.65s
NumPy	0.32s

Conclusão

- ✓ Arrays são estruturas para manipulação de dados numéricos em forma de vetores e matrizes.
- ✓ Numpy possui alta performance.



Referências

- ❑ **A quick introduction to the numpy array. Disponível em:**
<https://www.sharpsightlabs.com/blog/numpy-array-python/>
- ❑ **Numpy/Scipy – Python documentation. Disponível em:**
https://fgnt.github.io/python_crashkurs_doc/include/numpy.html
- ❑ **Pure Python vs NumPy vs TensorFlow Performance Comparison. Disponível em:**
<https://realpython.com/numpy-tensorflow-performance/>
- ❑ **NumPy. Disponível em:** <https://numpy.org>

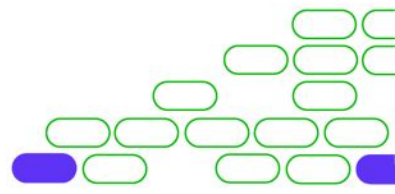


Próxima aula

- ❑ Criação de arrays – Prática.



XPe





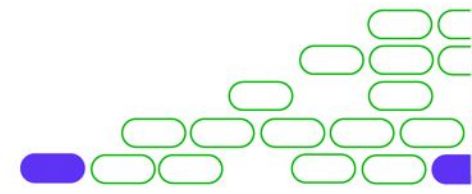
Faculdade



Introdução à Análise de Dados

**AULA 2.2. CRIAÇÃO DE ARRAYS –
PRÁTICA**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Criação de arrays no numpy.





Conclusão

- ✓ Aprendemos a criar arrays no numpy.



Na próxima aula

- ❑ Indexação de arrays.





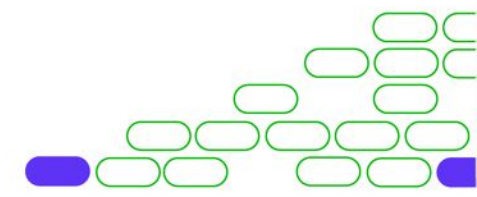
Faculdade



Introdução à Análise de Dados

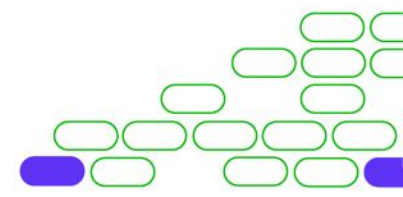
**AULA 2.3. INDEXAÇÃO DE
ARRAYS**

**PROF. MATHEUS
MENDONÇA**



Nesta aula

- ❑ Indexação de arrays:
 - ❑ Acessando elementos.
 - ❑ Slicing.



Numpy array

- Relembrando:
 - Os índices ao longo de uma dimensão variam de 0 a $n-1$, onde n é o número de elementos da dimensão.

A =

Index:	0	1	2	3	4
Value:	88	19	46	74	94



Acessando elementos

- Acessando o valor do elemento no índice 1 (segundo elemento do array):
 - $A[1] \rightarrow 19$

A =

Index:	0	1	2	3	4
Value:	88	19	46	74	94

Acessando elementos

- Acessando o último elemento de A:
 - $A[-1] \rightarrow 94$
 - $A[4] \rightarrow 94$

A =

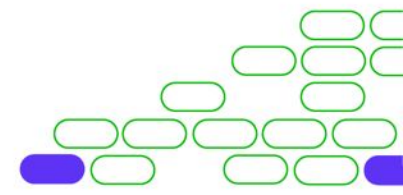
Index:	0	1	2	3	-1
Value:	88	19	46	74	94

Acessando elementos

- Índices negativos significam que o array será acessado de trás para frente:
 - $A[-1] \rightarrow 94$;
 - $A[-2] \rightarrow 74$;
 - etc.

A =

Index:	0	1	2	3	4
Value:	88	19	46	74	94





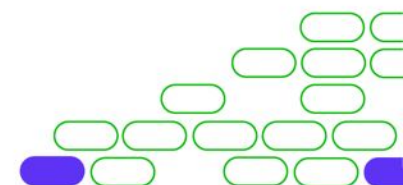
Acessando elementos

- Acessando elementos em um array 2D (matriz B):

○ B[2, 1] -> 45.

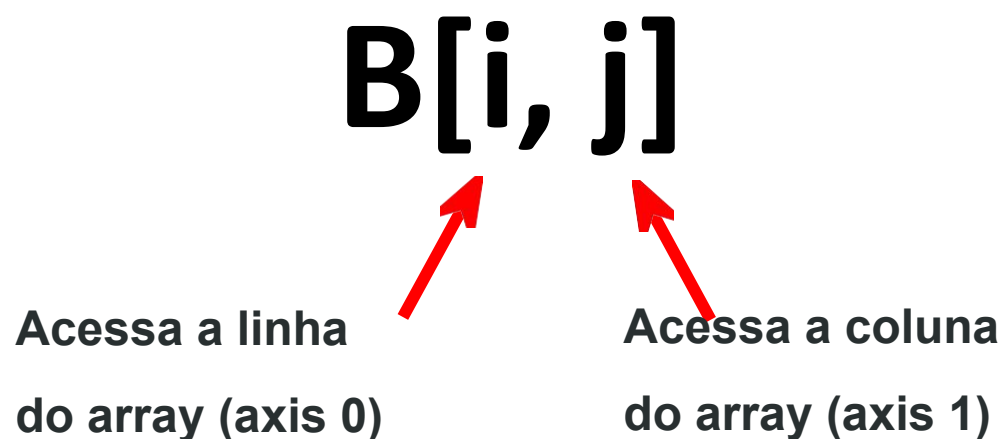
B =

	0	1	2	3	4
0	88	19	46	74	94
1	69	79	26	7	29
2	21	45	12	80	72
3	28	53	65	26	64
4	71	96	34	61	52



Acessando elementos

- Acessando elementos em um array 2D (matriz B):




Slicing

- Acessando mais de um elemento em um array:
 - `A[1:3]` ->

A =

Index:	0	1	2	3	4
Value:	88	19	46	74	94



Slicing

- Acessando elementos em um array 2D (matriz B):
 - `B[1:3, 1:4]` ->

B =

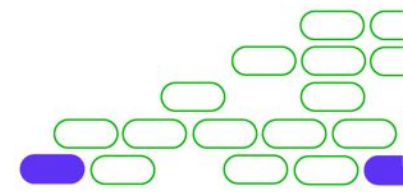
	0	1	2	3	4
0	88	19	46	74	94
1	69	79	26	7	29
2	21	45	12	80	72
3	28	53	65	26	64
4	71	96	34	61	52

Slicing

- Acessando elementos em um array 2D (matriz B):

$B[i:k, j:l]$

ATENÇÃO: os índices **k** e **l** não entram no slicing,
o slicing incluirá até **k-1** e **l-1**, respectivamente.



Conclusão

- ✓ Indexação.
- ✓ Slicing.



Referências

- ❑ A quick introduction to the numpy array. Disponível em:
<https://www.sharpsightlabs.com/blog/numpy-array-python/>



Próxima aula

- ❑ Prática de indexação de arrays.





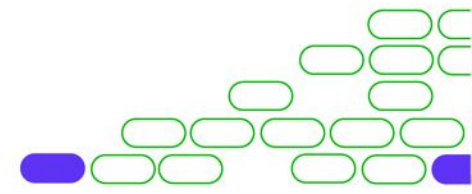
Faculdade



Introdução à Análise de Dados

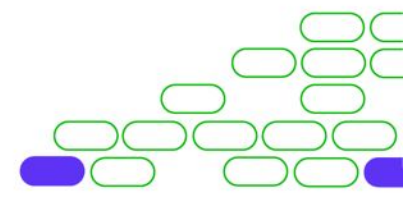
**AULA 2.4. INDEXAÇÃO DE ARRAYS –
PRÁTICA**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Indexação de arrays:
 - ❑ Acessando elementos.
 - ❑ Slicing.



Conclusão

- ✓ Indexação.
- ✓ Slicing.

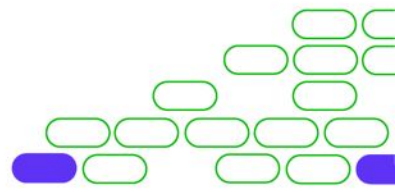


Próxima aula

- ❑ Operações aritméticas.



XPe





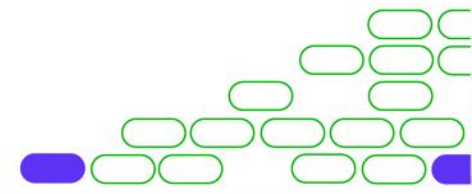
Faculdade



Introdução à Análise de Dados

**AULA 2.5. OPERAÇÕES
ARITMÉTICAS**

PROF. MATHEUS MENDONÇA



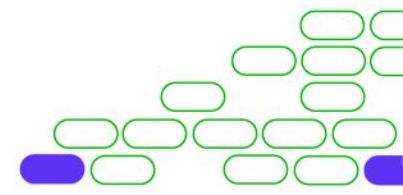
Nesta aula

- ❑ Operações aritméticas:
 - ❑ Operações elemento a elemento.
 - ❑ Broadcasting.
 - ❑ Operações matriciais.

Operações elemento a elemento

Operações aritméticas elemento a elemento:

- Soma:
 - Sobrecarga de operador “+”;
 - np.add.
- Subtração
 - Sobrecarga de operador “-”;
 - np.subtract.
- Divisão
 - Sobrecarga de operador “/”;
 - np.divide
- Multiplicação
 - Sobrecarga de operador “*”;
 - np.multiply

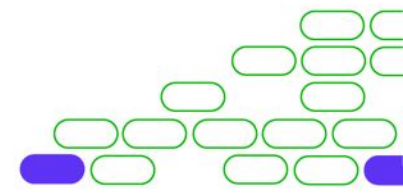


Operações elemento a elemento

$$u = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$u + v = \begin{bmatrix} 1+0 \\ 0+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Fonte: <https://cognitiveclass.ai/blog/nested-lists-multidimensional-numpy-arrays>

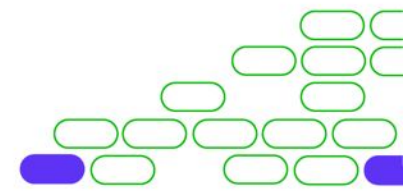


Operações elemento a elemento

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$X \circ Y = \begin{bmatrix} (0)2 & (0)1 \\ (0)1 & (1)2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Fonte: <https://cognitiveclass.ai/blog/nested-lists-multidimensional-numpy-arrays>

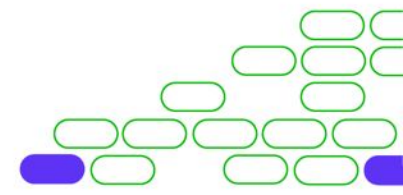


Broadcasting

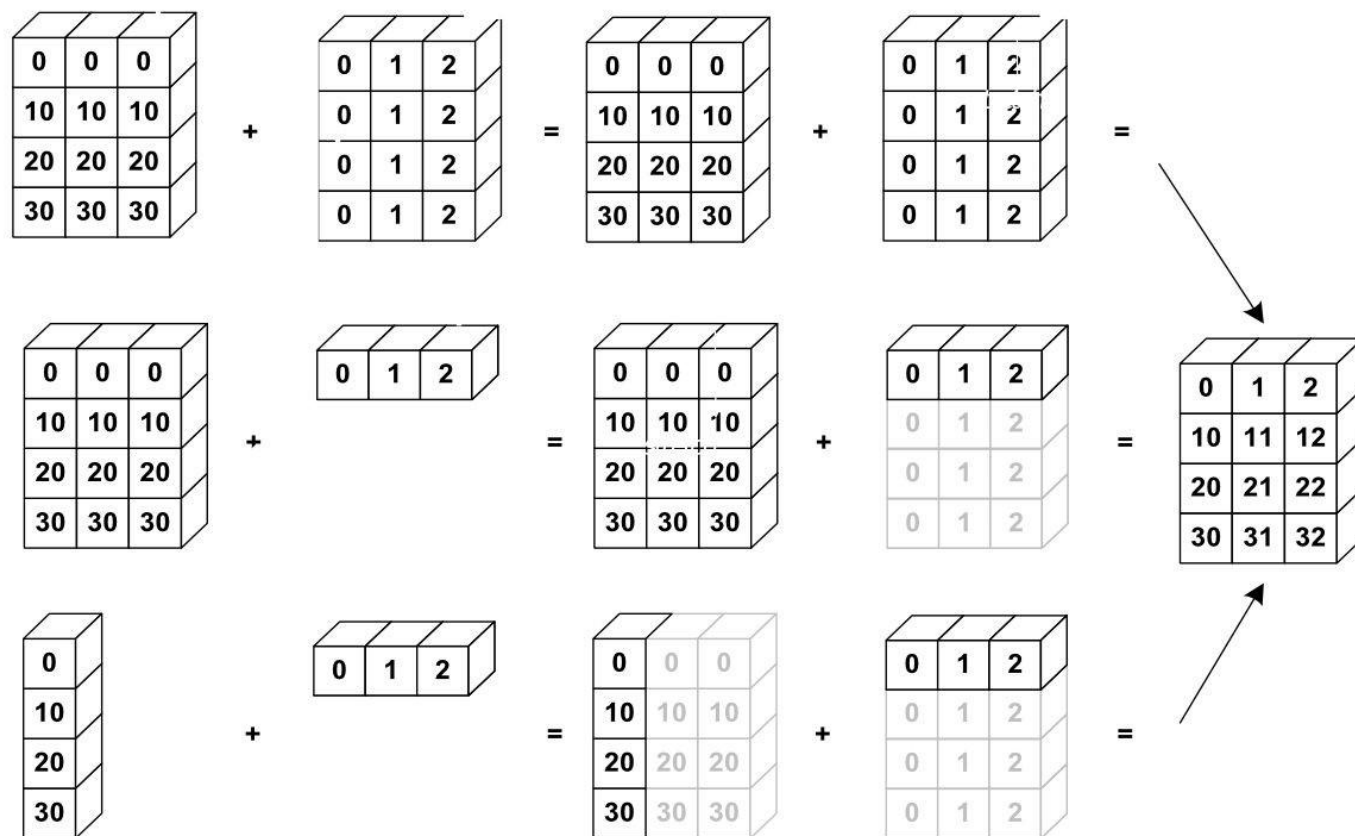
$$y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$2y = \begin{bmatrix} 2(1) \\ 2(2) \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Fonte: <https://cognitiveclass.ai/blog/nested-lists-multidimensional-numpy-arrays>



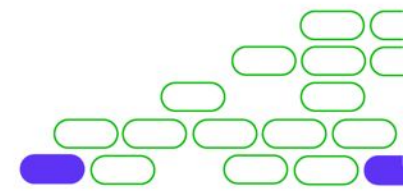
Broadcasting



Operações matriciais

Multiplicação de matrizes:

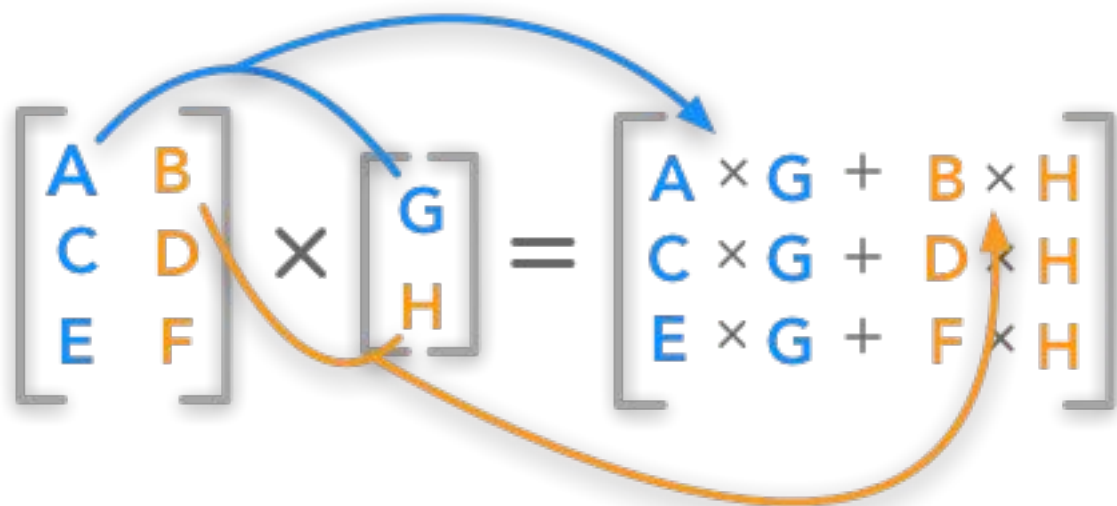
- Python puro: `A @ B`
- Numpy:
 - `np.dot(A, B)`
 - `A.dot(B)`



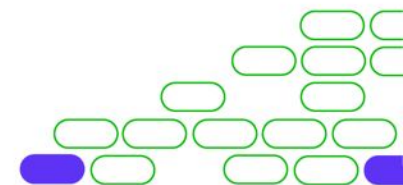
Operações matriciais

Multiplicação de matrizes:

- Python puro: `A @ B`
- Numpy:
 - `np.dot(A, B)`
 - `A.dot(B)`



$$\begin{bmatrix} A & B \\ C & D \\ E & F \end{bmatrix} \times \begin{bmatrix} G \\ H \end{bmatrix} = \begin{bmatrix} A \times G + B \times H \\ C \times G + D \times H \\ E \times G + F \times H \end{bmatrix}$$



Conclusão

- ✓ Operações elemento a elemento.
Broadcasting.
- ✓ Operações matriciais.
- ✓



Referências

- ❑ **From Python Nested Lists to Multidimensional numpy Arrays. Disponível em:**
<https://cognitiveclass.ai/blog/nested-lists-multidimensional-numpy-arrays>
- ❑ **Introduction to Multiplying Matrices and Vectors using Python/Numpy examples and drawings. Disponível em:**
<https://hadrienj.github.io/posts/Deep-Learning-Book-Series-2.2-Multiplying-Matrices-and-Vectors/>

Próxima aula

- ❑ Prática de operações aritméticas com arrays.





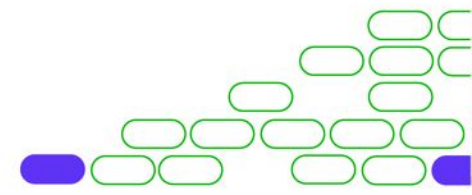
Faculdade



Introdução à Análise de Dados

AULA 2.6. OPERAÇÕES ARITMÉTICAS: OPERAÇÕES ELEMENTO A ELEMENTO (PRÁTICA)

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Operações aritméticas:
 - ❑ Operações elemento a elemento.
 - ❑ Broadcasting.



Conclusão

- ✓ Operações aritméticas elemento a elemento.
- ✓ Broadcasting.



Próxima aula

- ❑ Prática de operações aritméticas com arrays.





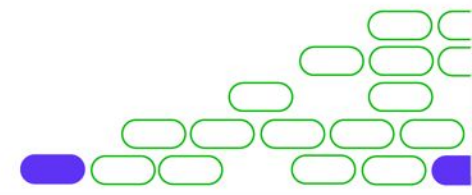
Faculdade



Introdução à Análise de Dados

**AULA 2.7. OPERAÇÕES ARITMÉTICAS: OPERAÇÕES
MATRICIAIS (PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Operações aritméticas:
 - ❑ Operações matriciais.



Conclusão

✓ Operações
matriciais.



Próxima aula

- ❑ Comparações e indexação booleana.





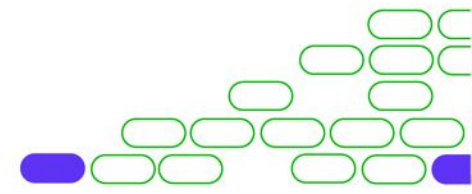
Faculdade



Introdução à Análise de Dados

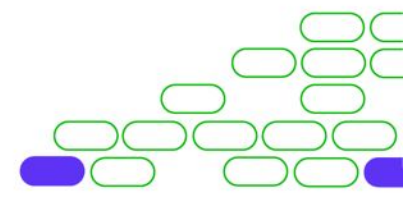
**AULA 2.8. COMPARAÇÕES E INDEXAÇÃO
BOOLEANA**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Comparações.
- ❑ Indexação booleana.



Comparações

- Comparação menor/menor ou igual:

```
# comparações booleanas
A = np.array([1, 2, 3])
B = np.array([2, 0, 2])
s = 3

# menor
print("Comparação menor:")
print(A < B)
print(A < s)

# menor ou igual
print("Comparação menor ou igual:")
print(A <= B)
print(A <= s)
```

```
Comparação menor:
[ True False False]
[ True  True False]
Comparação menor ou igual:
[ True False False]
[ True  True  True]
```

Comparações

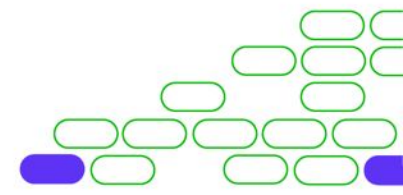
- Comparação maior/maior ou igual:

```
# comparações booleanas
A = np.array([1, 2, 3])
B = np.array([2, 0, 2])
s = 3

# maior
print("Comparação maior:")
print(A > B)
print(A > s)

# maior ou igual
print("Comparação maior ou igual:")
print(A >= B)
print(A >= s)
```

```
Comparação maior:
[False True True]
[False False False]
Comparação maior ou igual:
[False True True]
[False False True]
```



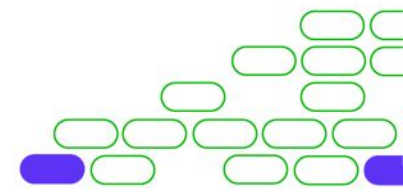
Comparações

- Igualdade:

```
# comparações booleanas  
A = np.array([1, 2, 3])  
B = np.array([2, 0, 2])  
s = 3
```

```
# igual  
print("Comparação de igualdade:")  
print(A == B)  
print(A == s)
```

```
Comparação de igualdade:  
[False False False]  
[False False  True]
```

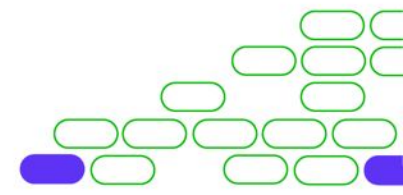


Indexação booleana

- Operação de **filtro**:

```
# indexação booleana: um novo subarray contendo uma  
# cópia dos elementos em que a condição de verificação se aplica  
cond = A <= 2  
D = A[cond]  
print("A:", A)  
print("condição:", cond)  
print("D:", D)
```

```
A: [1 2 3]  
condição: [ True  True False]  
D: [1 2]
```



Conclusão

- ✓ Comparação são operações elemento a elemento.
- ✓ Indexação booleana: filtro.



Próxima aula

- ❑ Prática de comparações e indexação booleana.





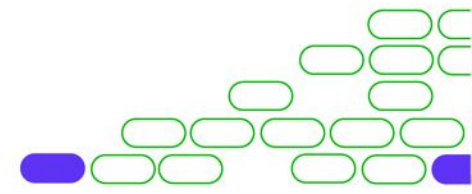
Faculdade



Introdução à Análise de Dados

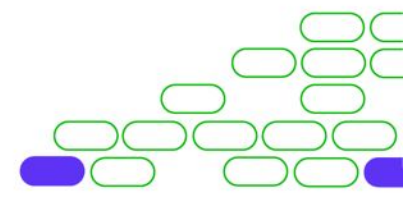
**AULA 2.9. COMPARAÇÕES E INDEXAÇÃO
BOOLEANA (PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Comparações.
- ❑ Indexação booleana.



Conclusão

- ✓ Comparação são operações elemento a elemento.
- ✓ Indexação booleana: filtro.



Próxima aula

- ❑ Dicas de numpy.





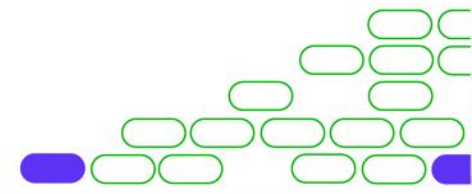
Faculdade



Introdução à Análise de Dados

**AULA 2.10. OPERAÇÕES ÚTEIS NO NUMPY
(PRÁTICA)**

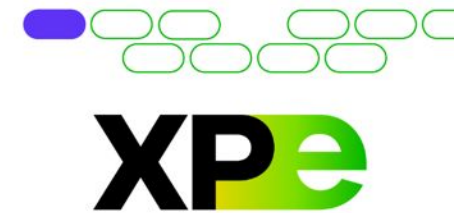
PROF. MATHEUS MENDONÇA



Nesta aula

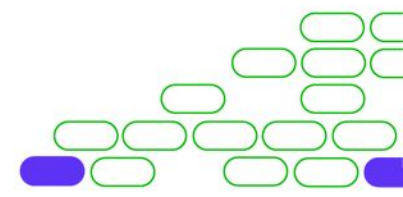
- ❑ Dicas gerais de numpy.





Conclusão

✓ Dicas de
numpy.



Próxima aula

- ❑ Regressão linear.





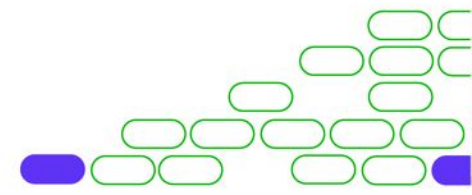
Faculdade



Introdução à Análise de Dados

**AULA 2.11. REGRESSÃO LINEAR NO NUMPY:
CONCEITOS BÁSICOS**

PROF. MATHEUS MENDONÇA



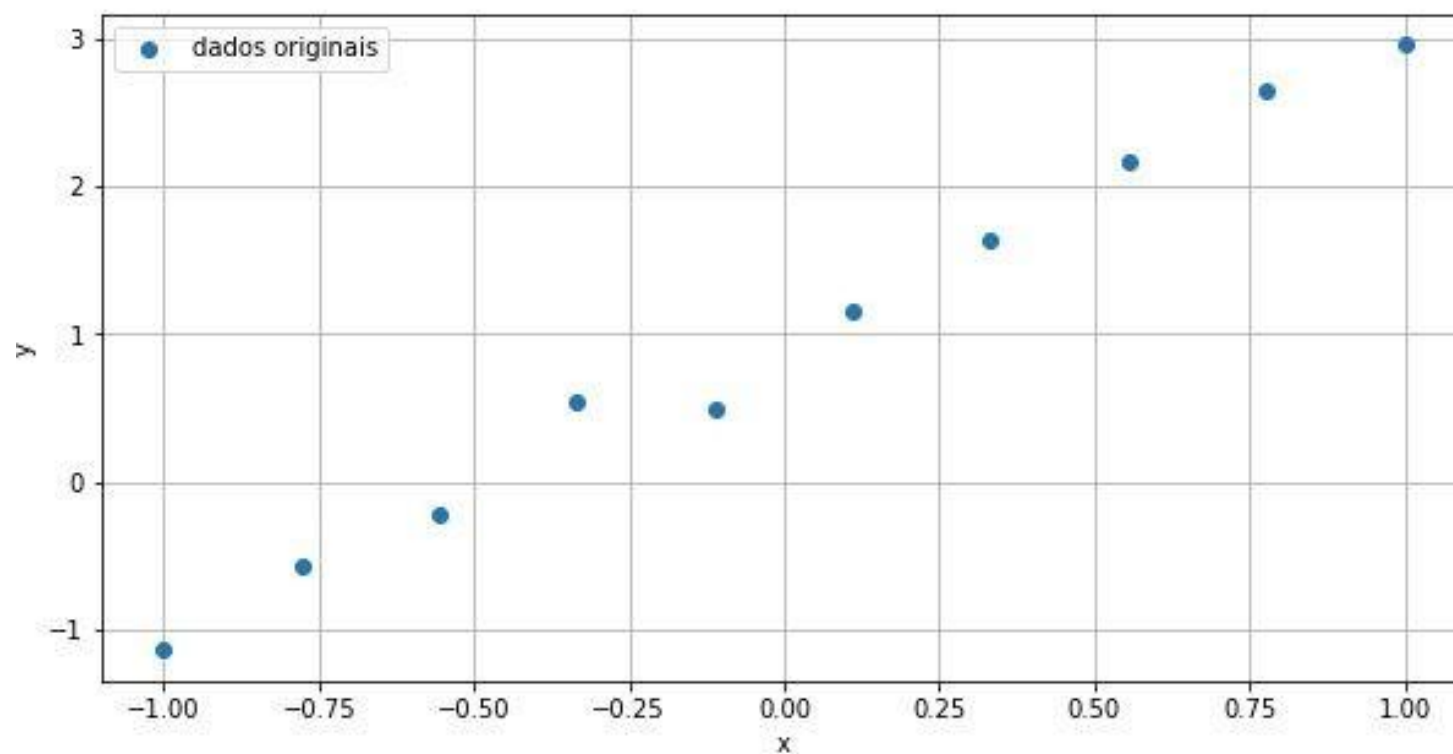
Nesta aula

- ❑ Conceitos básicos de regressão linear.



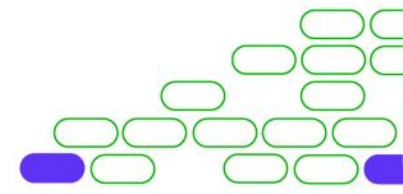
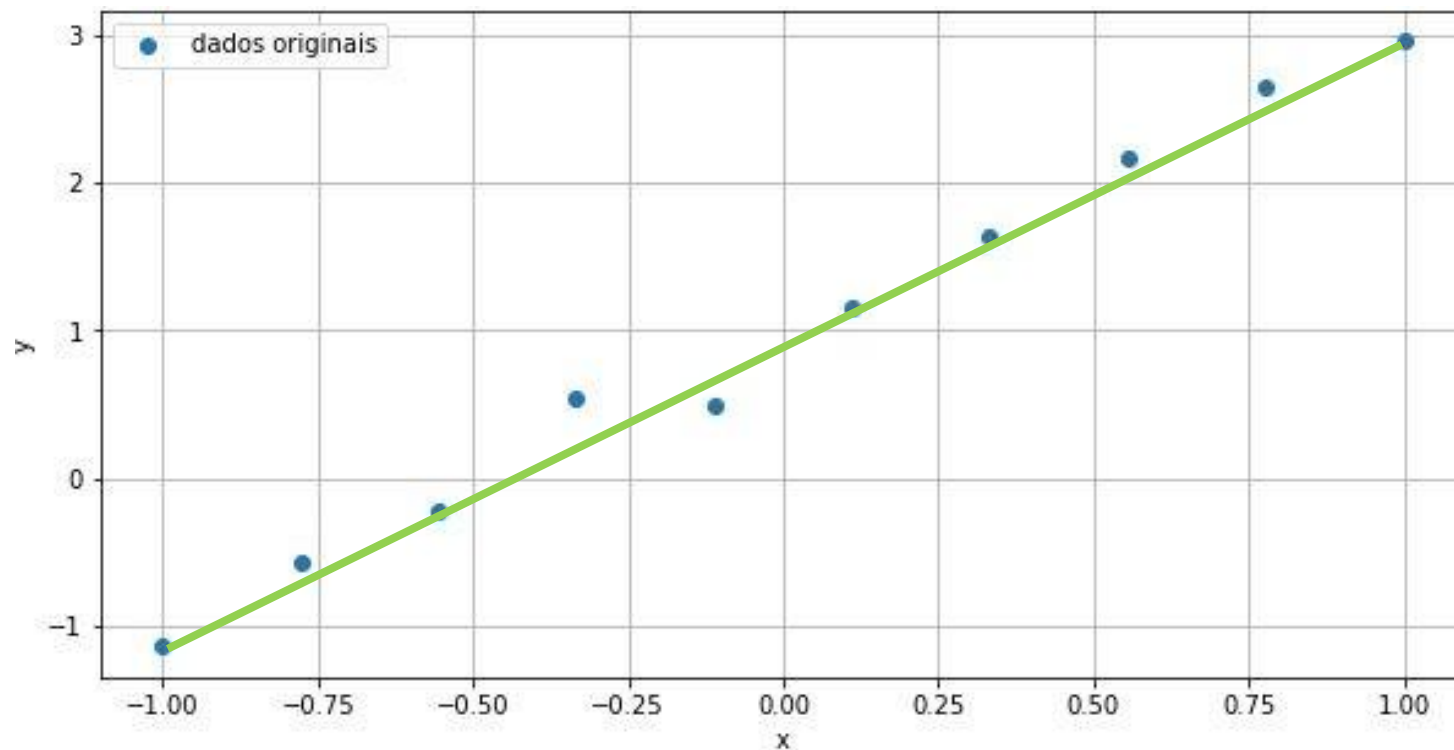
Regressão linear

Problema: dado um conjunto de pontos, queremos achar qual a função que melhor descreve esses pontos.



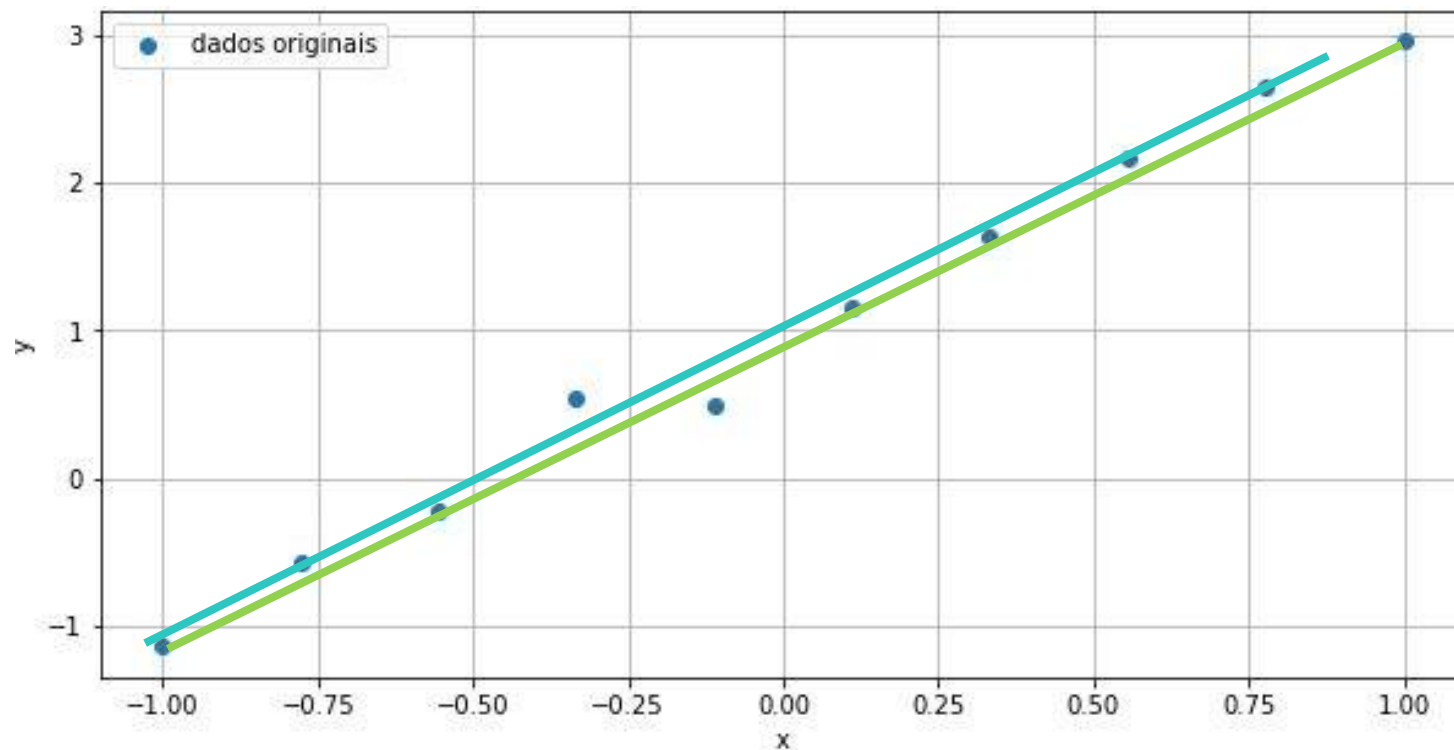
Regressão linear

Diversas possíveis soluções... qual função escolher?



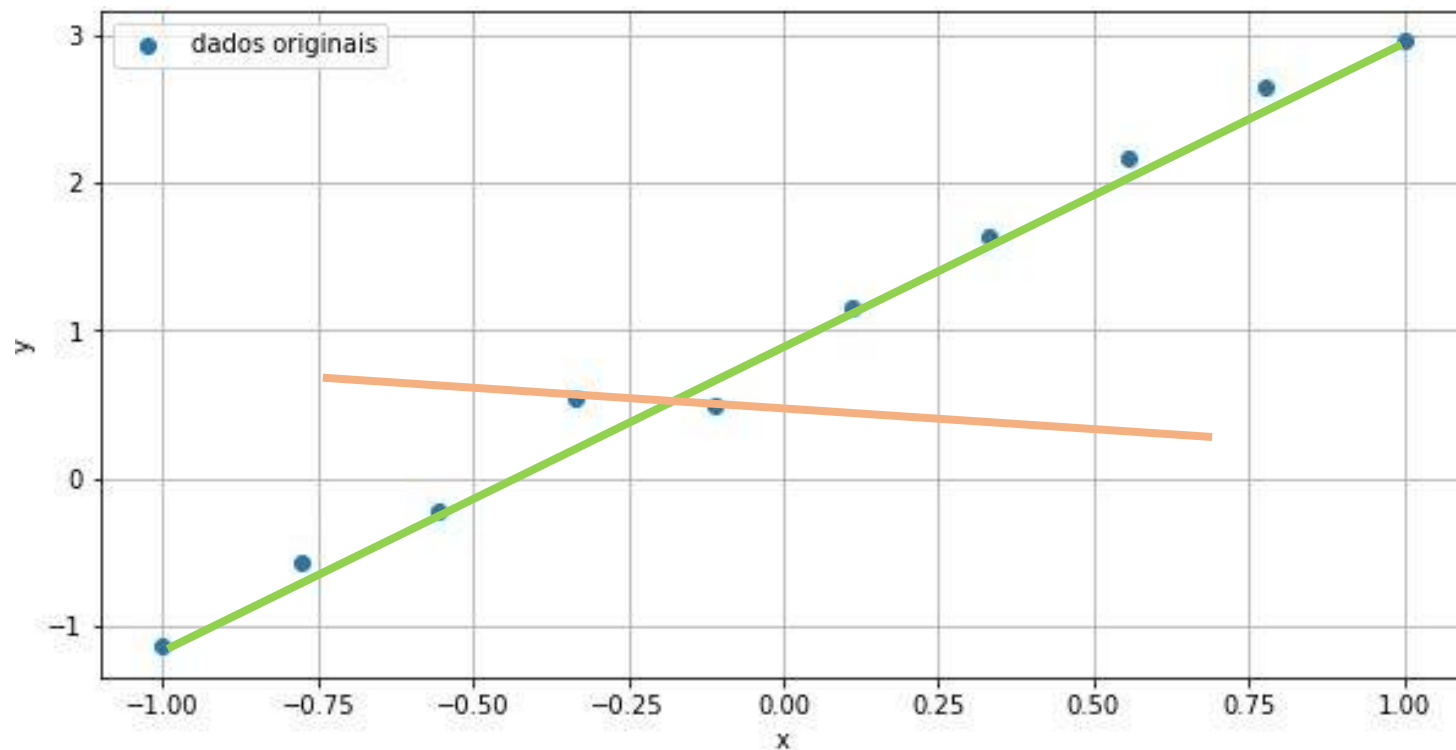
Regressão linear

Diversas possíveis soluções... qual função escolher?



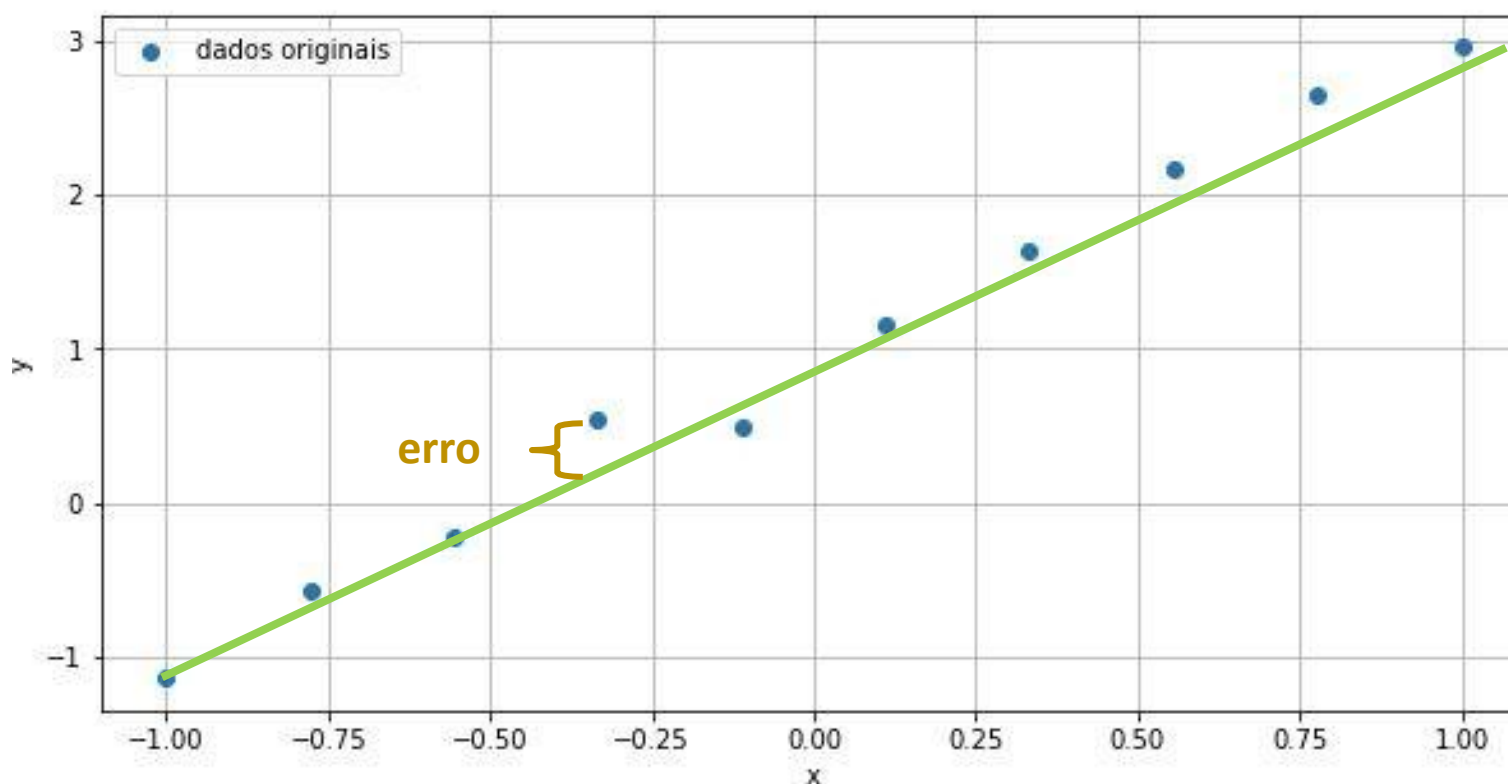
Regressão linear

Diversas possíveis soluções... qual função escolher?



Regressão linear

- A escolha da melhor função deve ser baseada em um critério.
- Um critério comumente utilizada é o **erro quadrático**, que queremos minimizar:

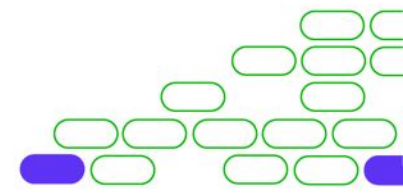


Regressão linear

- A escolha da melhor função deve ser baseada em um critério.
- Um critério comumente utilizada é o erro quadrático, que queremos minimizar.
- Com o critério definido e, sabendo que a função linear possui o seguinte formato:

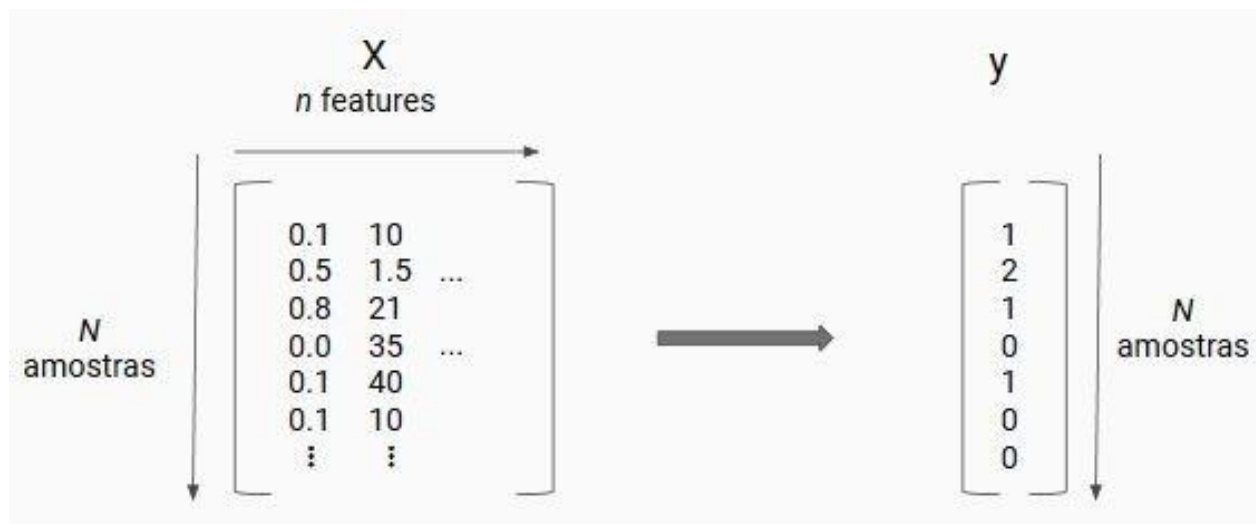
$$y = f(x) = ax + b$$

- O problema de regressão resume-se à determinação dos coeficientes a e b , visto que x e y são dados de entrada.



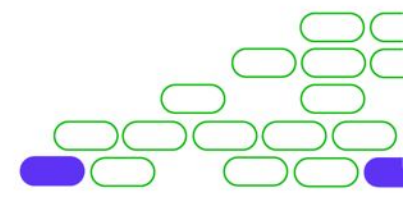
Regressão linear

- Matricialmente:



- A solução fechada deste problema é dada por:

$$(X^T X)^{-1} X^T y$$

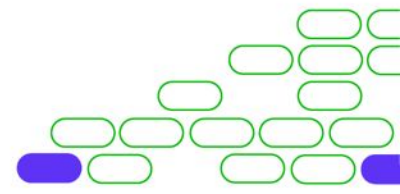
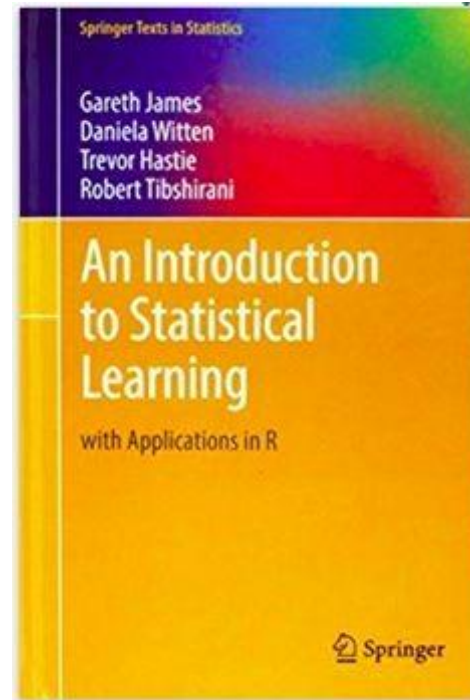
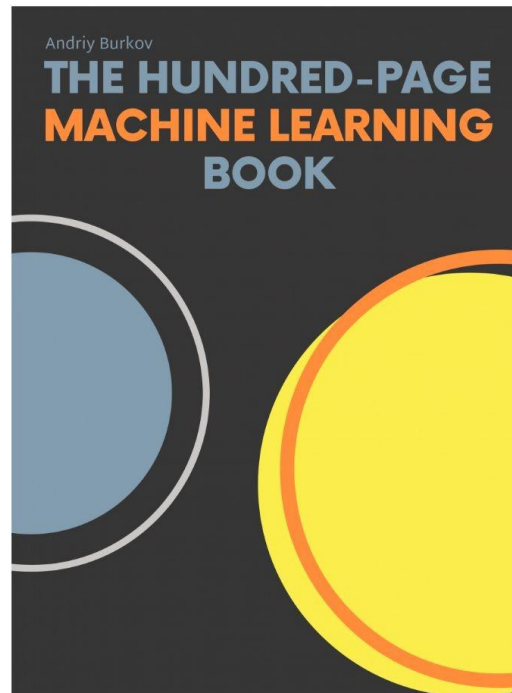


Conclusão

- ✓ Conceitos de regressão linear.



Referências



Próxima aula

- ❑ Regressão linear no numpy (Prática).





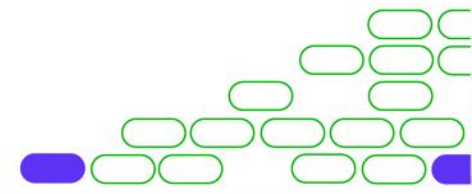
Faculdade



Introdução à Análise de Dados

**AULA 2.12. REGRESSÃO LINEAR NO NUMPY
(PRÁTICA)**

PROF. MATHEUS MENDONÇA



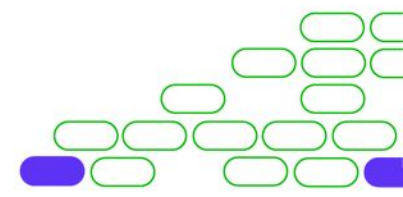
Nesta aula

- ❑ Conceitos básicos de regressão linear.
- ❑ Regressão linear no numpy.



Conclusão

- ✓ Conceitos de regressão linear.
- ✓ Regressão linear no numpy.

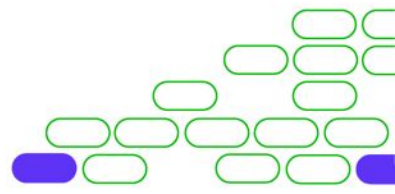


Próxima aula

- ❑ Pandas.



XPe





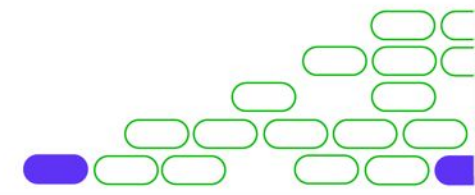
Faculdade



Introdução à Análise de Dados

**CAPÍTULO 3. PANDAS PARA A ANÁLISE
DE DADOS**

PROF. MATHEUS MENDONÇA





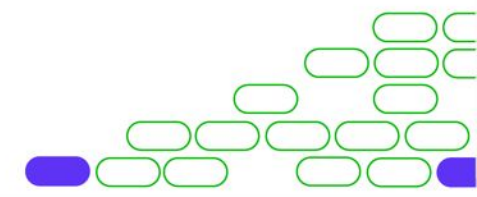
Faculdade



Introdução à Análise de Dados

**AULA 3.1. INTRODUÇÃO AO
PANDAS**

PROF. MATHEUS MENDONÇA



Nesta aula

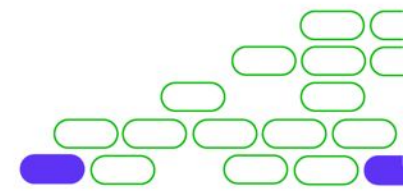
- ❑ Introdução ao Pandas.
- ❑ Dtypes e tipos de objetos.
- ❑ Leitura de dados.



Introdução ao Pandas

[Pandas](#) é um pacote em Python desenvolvido para disponibilizar estruturas de dados rápidas e flexíveis para se trabalhar com dados “relacionais” ou “rotulados”. Ele é adequado para diversos tipos de dados:

- Dados tabulares com colunas de tipos heterogêneos, como por exemplo em tabelas SQL ou planilhas Excel.
- Dados de séries temporais ordenados ou não ordenados.
- Dados matriciais arbitrários, com linhas e colunas.

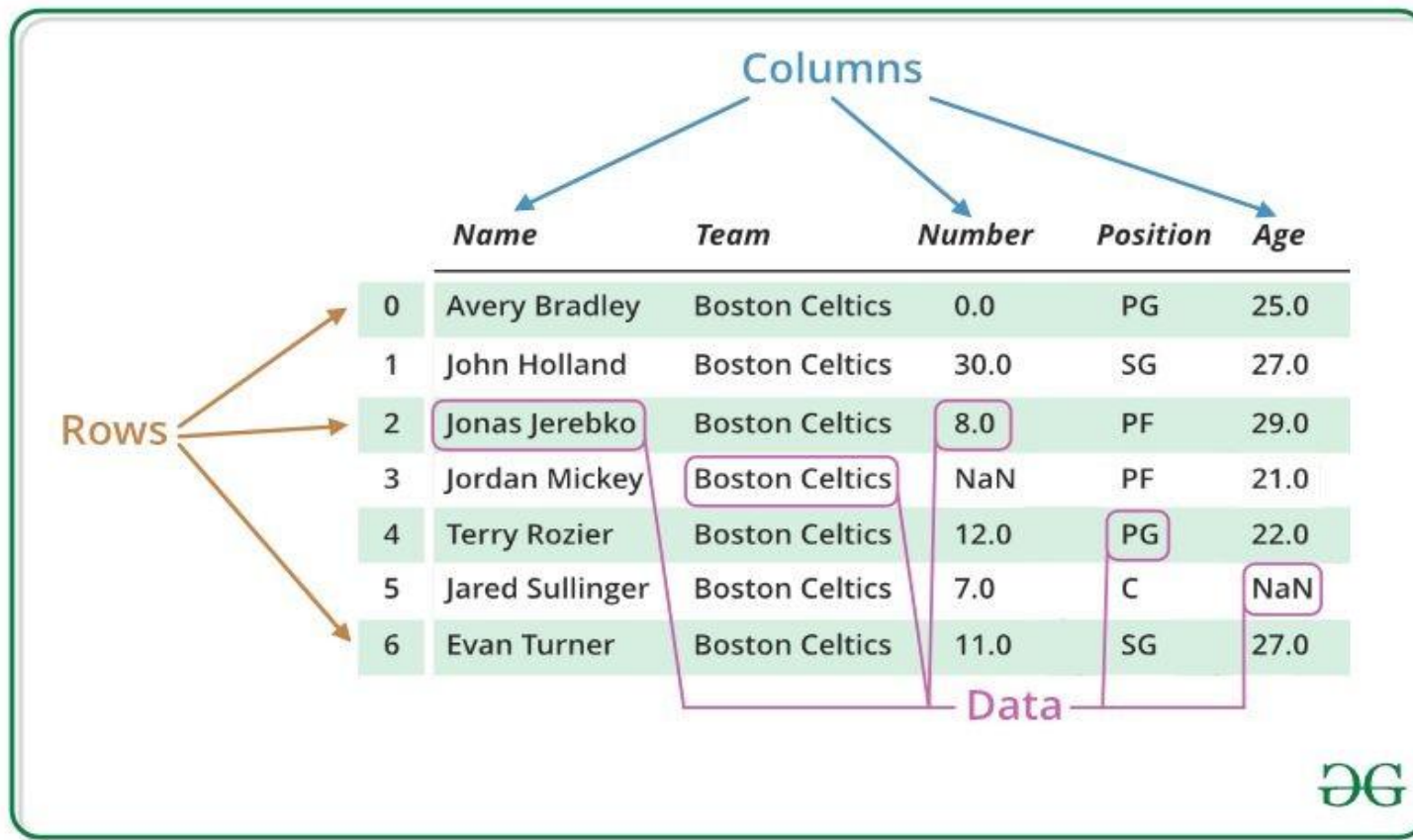


Introdução ao Pandas

- Flexível:
 - Escrito em cima do numpy.
 - Possui métodos do matplotlib.
 - Usado em conjunto com outras bibliotecas de ciência de dados (scipy, scikit-learn etc.).
- Instalação:
 - `pip install pandas`.
 - `conda install pandas`.
- Importando no ambiente de desenvolvimento:
 - `import pandas as pd`.



Introdução ao Pandas



The diagram illustrates a Pandas DataFrame structure. A table is shown with 7 rows and 5 columns. The columns are labeled *Name*, *Team*, *Number*, *Position*, and *Age*. The rows are indexed from 0 to 6. Annotations include:

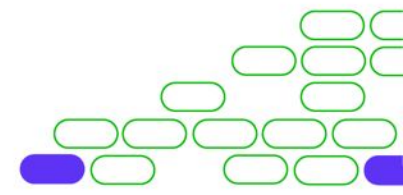
- Columns:** A blue label with arrows pointing to the column headers.
- Rows:** A brown label with arrows pointing to the row indices (0, 1, 2, 3, 4, 5, 6).
- Data:** A purple label with a bracket pointing to the data cells of the table.

 The data is as follows:

	<i>Name</i>	<i>Team</i>	<i>Number</i>	<i>Position</i>	<i>Age</i>
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Fonte:

<https://www.geeksforgeeks.org/python-pandas-dataframe/>



Tipos de dados (dtypes)

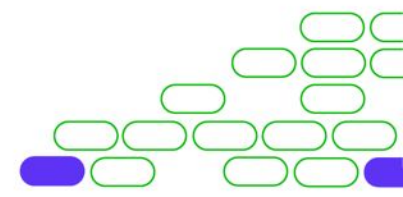
Pandas dtype	Python type	Uso
object	str ou mixed	Texto ou valores mistos numéricos e não-numéricos.
int64	int	Números inteiros.
float64	float	Números ponto flutuantes.
bool	bool	Valores True/False.
datetime64	NA	Valores em formato de data e hora.
timedelta[ns]	NA	Diferença de dois datetimes.
category	NA	Lista finita de texto.

DataFrames e Series

Series 1		Series 2		Series 3		DataFrame
Mango		Apple		Banana		Mango Apple Banana
0 4		0 5		0 2		0 4 5 2
1 5		1 4		1 3		1 5 4 3
2 6		2 3		2 5		2 6 3 5
3 3		3 0		3 2		3 3 0 2
4 1		4 2		4 7		4 1 2 7

Fonte:

<https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/>



Leitura de dados

Para leitura dos dados existem diversas funções, a depender do formato do dado de entrada. Algumas das mais usadas estão listadas abaixo:

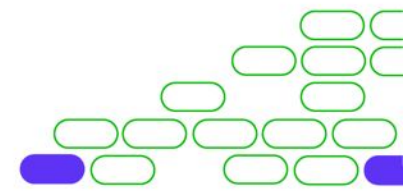
- **read_csv**: leitura de arquivos CSV.
- **read_json**: leitura de arquivos JSON.
- **read_excel**: leitura de arquivos Excel.
- Etc.



Aplicações

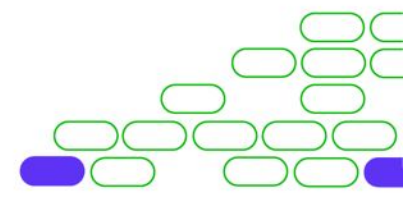
Algumas das tarefas que o Pandas faz com eficiência, são:

- Tratamento de dados faltantes (representados por NaN).
- Tamanhos mutáveis: colunas podem ser inseridas e excluídas de *DataFrames* com facilidade.
- Grupo de funcionalidades poderoso e flexível para agregar e transformar conjuntos de dados.
- Ferramentas de IO robustas para leitura de dados de arquivos, como CSV, Excel e bancos de dados.
- Entre outros.



Conclusão

- ✓ Introdução ao Pandas.
- ✓ Tipos de dados.



Referências

- ❑ **Python Pandas Tutorial: A Complete Introduction for Beginners.** Disponível em:
<https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/>
- ❑ **Pandas.** Disponível em: <https://pandas.pydata.org/>



Próxima aula

- ❑ Introdução ao Pandas – Prática.





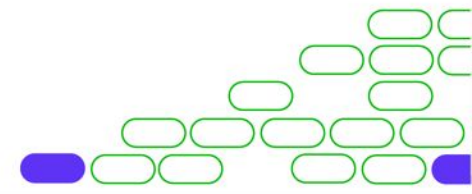
Faculdade



Introdução à Análise de Dados

**AULA 3.2. INTRODUÇÃO AO PANDAS
(PRÁTICA)**

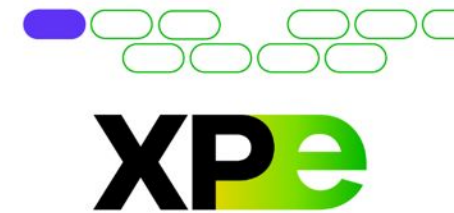
PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Introdução ao Pandas.





Conclusão

✓ Introdução ao
Pandas.

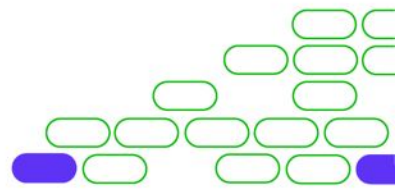


Próxima aula

- ❑ Indexação.



XPe





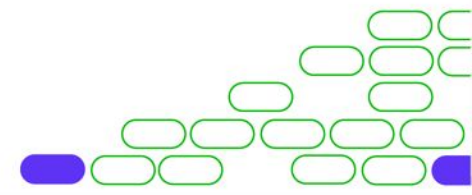
Faculdade



Introdução à Análise de Dados

**AULA 3.3. INDEXAÇÃO NO
PANDAS**

**PROF. MATHEUS
MENDONÇA**



Nesta aula

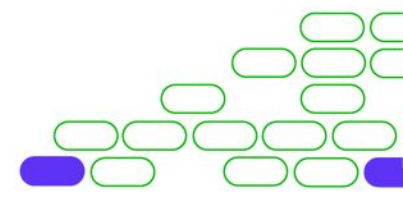
- ❑ Indexação no pandas:
 - ❑ Método `iloc()`.
 - ❑ Método `loc()`.
- ❑ Indexação booleana.

Indexação direta



`data_frame['temperatura']`

	date	temperatura	classification
0	2020-01-01	29.1	quente
1	2020-02-01	31.2	muito quente
2	2020-03-01	28.5	quente



Indexação direta

✓ `data_frame[['temperatura',
'classification']]`

	date	temperatura	classification
0	2020-01-01	29.1	quente
1	2020-02-01	31.2	muito quente
2	2020-03-01	28.5	quente

Método `iloc()`

- ✓ Similar à indexação no numpy:

Nome do DataFrame **Indexação de múltiplas colunas ()**

↑ ↑

`data_frame.iloc[i:k, j:l]`

↓

Indexação de múltiplas linhas

Método loc()

- ✓ Indexação pelo **nome** da linha ou coluna:

Nome do DataFrame

Nome da coluna (str ou list)

`data_frame.loc[nome_linha, nome_coluna]`

Nome da linha (str ou list)

Indexação booleana



```
df[df['classification']=='quente']
```

	date	temperatura	classification
0	2020-01-01	29.1	quente
1	2020-02-01	31.2	muito quente
2	2020-03-01	28.5	quente

Indexação booleana

- `df.loc[df['classification']=='quente', 'temperatura']`

	date	temperatura	classification
0	2020-01-01	29.1	quente
1	2020-02-01	31.2	muito quente
2	2020-03-01	28.5	quente

Conclusão

- ✓ Indexação no Pandas:
 - ✓ Método iloc.
 - ✓ Método loc.
- ✓ Indexação booleana.

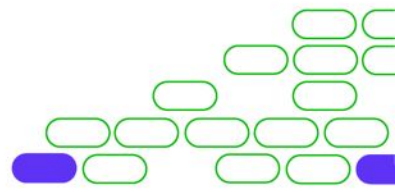


Próxima aula

- Indexação – Prática.



XPe





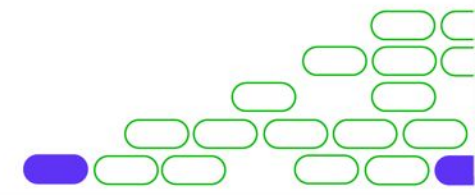
Faculdade



Introdução à Análise de Dados

**AULA 3.4. INDEXAÇÃO NO
PANDAS**

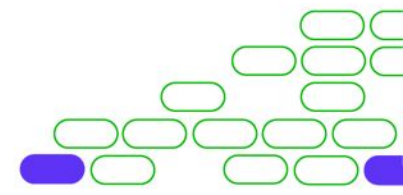
**PROF. MATHEUS
MENDONÇA**



Nesta aula

❑ Indexação no Pandas:

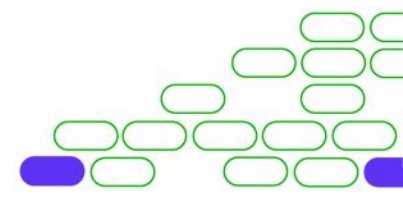
- ❑ Método `iloc()`.
- ❑ Método `loc()`.



Conclusão

✓ Indexação no
Pandas:

- ✓ Método iloc.
- ✓ Método loc.



Próxima aula

- ❑ Indexação booleana (Prática).





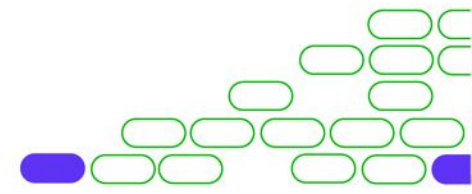
Faculdade



Introdução à Análise de Dados

**AULA 3.5. INDEXAÇÃO BOOLEANA NO
PANDAS**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Indexação booleana.





Conclusão

✓ Indexação
booleana.

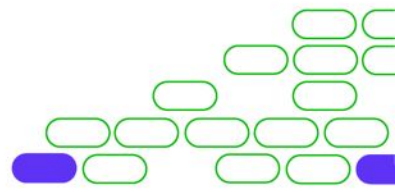


Próxima aula

- ❑ Ordenação.



XPe





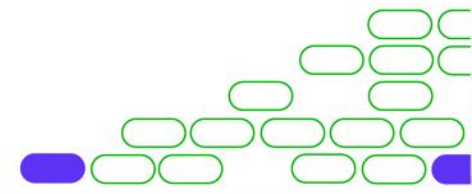
Faculdade



Introdução à Análise de Dados

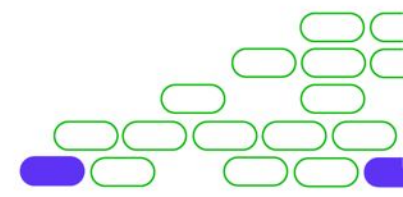
**AULA 3.6. ORDENAÇÃO NO
PANDAS**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Ordenação no Pandas:
 - ❑ Método `sort_values`.



Método sort_values()

- `df.sort_values(by=['col1'])`

	col1	col2	col3			col1	col2	col3
0	A	2	0		0	A	2	0
1	A	1	1		1	A	1	1
2	B	9	9	➡	2	B	9	9
3	NaN	8	4		5	C	4	3
4	D	7	2		4	D	7	2
5	C	4	3		3	NaN	8	4

Método sort_values()

- `df.sort_values(by='col1', ascending=False)`

	col1	col2	col3
0	A	2	0
1	A	1	1
2	B	9	9
3	NaN	8	4
4	D	7	2
5	C	4	3

	col1	col2	col3
4	D	7	2
5	C	4	3
2	B	9	9
0	A	2	0
1	A	1	1
3	NaN	8	4

Método sort_values()

- `df.sort_values(by=['col1', 'col2'])`.

	col1	col2	col3		col1	col2	col3
0	A	2	0		1	A	1
1	A	1	1		0	A	2
2	B	9	9	➔	2	B	9
3	NaN	8	4		5	C	4
4	D	7	2		4	D	7
5	C	4	3		3	NaN	8

Método sort_values()

- `df.sort_values(by='col1', na_position='first')`.

	col1	col2	col3			col1	col2	col3
0	A	2	0		3	NaN	8	4
1	A	1	1		0	A	2	0
2	B	9	9	➡	1	A	1	1
3	NaN	8	4		2	B	9	9
4	D	7	2		5	C	4	3
5	C	4	3		4	D	7	2

Conclusão

- ✓ Ordenação no Pandas.

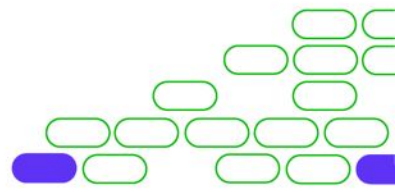


Próxima aula

- ❑ Ordenação – Prática.



XPe





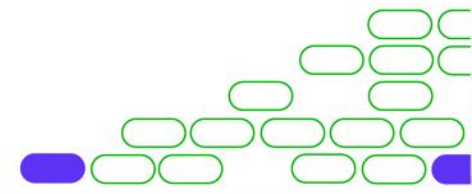
Faculdade



Introdução à Análise de Dados

**AULA 3.7. ORDENAÇÃO NO PANDAS
(PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

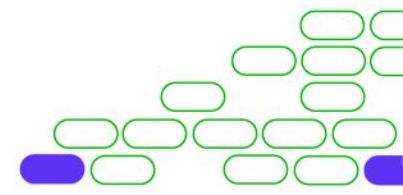
- ❑ Ordenação no Pandas:
 - ❑ Método `sort_values`.
 - ❑ Método `sort_index`.



Conclusão

✓ Ordenação no
Pandas:

- ✓ Método `sort_values`;
- ✓ Método `sort_index`.



Próxima aula

- ❑ Visualização de dados no Pandas (Prática).





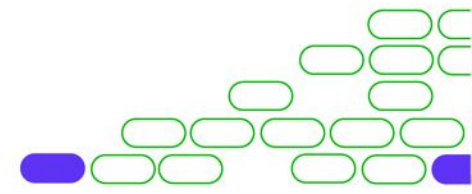
Faculdade



Introdução à Análise de Dados

**AULA 3.8. VISUALIZAÇÃO DE DADOS NO
PANDAS (PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

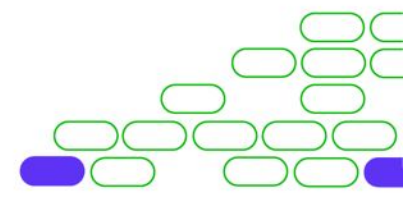
- ❑ Introdução à visualização de dados no Pandas:
 - ❑ Plot de linhas.
 - ❑ Plot de barras.
 - ❑ Plot de “pizza”.





Conclusão

- ✓ Visualização de dados no Pandas.



Próxima aula

- ❑ Dicas gerais no Pandas (Prática).





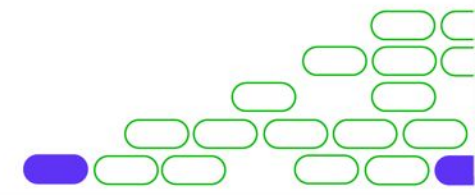
Faculdade



Introdução à Análise de Dados

**AULA 3.9. DICAS GERAIS SOBRE O PANDAS
(PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Dicas gerais sobre o Pandas.

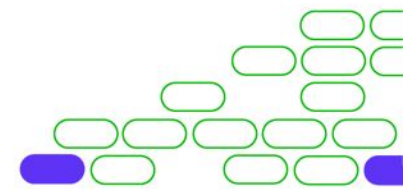


Conclusão



Dicas:

- ✓ Método groupby.
- ✓ Operações inplace.
- ✓ Compartilhamento de memória em cópias.



Próxima aula

- ❑ Introdução ao aprendizado de máquinas.





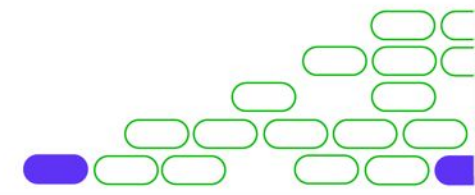
Faculdade



Introdução à Análise de Dados

CAPÍTULO 4. INTRODUÇÃO AO APRENDIZADO DE MÁQUINAS

PROF. MATHEUS MENDONÇA





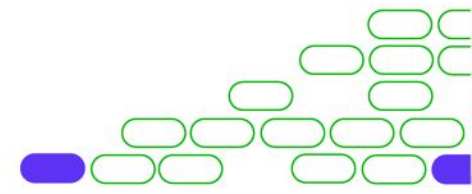
Faculdade



Introdução à Análise de Dados

**AULA 4.1. INTRODUÇÃO AO APRENDIZADO DE
MÁQUINAS**

PROF. MATHEUS MENDONÇA



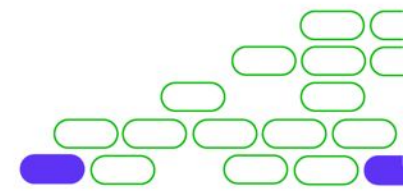
Nesta aula

- ❑ Introdução ao aprendizado de máquinas.



Aprendizado de máquinas

Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.



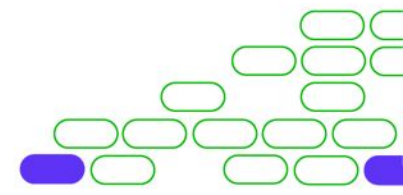
Aprendizado de máquinas

- O aprendizado de máquinas utiliza um conjunto de ferramentas para modelagem e análise de dados denominado Aprendizado Estatístico.
- Abordagem estatística para o problema de Aprendizado de Máquina:
 - Desenvolvimento de modelos capazes de aprender a partir de dados.



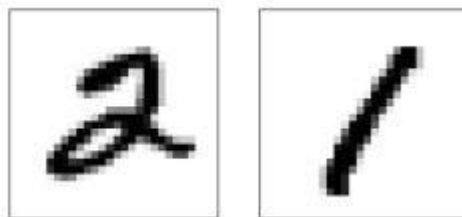
Abordagem simbólica

Abordagem simbólica para a classificação de dígitos:



Abordagem simbólica

Abordagem simbólica para a classificação de dígitos:



Suponha que exista um algoritmo capaz de contar o número de retas e curvas em uma imagem de um dígito:

SE DÍGITO É COMPOSTO POR UMA RETA ENTÃO “UM”
SE DÍGITO É COMPOSTO POR UMA OU MAIS CURVAS ENTÃO “DOIS”

Abordagem simbólica

- Conhecimento do problema é representado por meio de regras (if/else).
- Facilidade de entender o mecanismo de inferência que gerou o resultado.
- Facilidade de alteração do conhecimento do problema.

4

SE DÍGITO É COMPOSTO POR UMA RETA ENTÃO “UM”
SE DÍGITO É COMPOSTO POR UMA OU MAIS CURVAS ENTÃO “DOIS”
SE DÍGITO É COMPOSTO POR TRÊS RETAS ENTÃO “QUATRO”



Abordagem simbólica

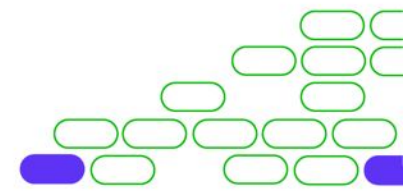
- Dificuldade de modelagem de todo o problema.
- Dificuldade de lidar com incertezas, informação imprecisas etc.



Aprendizado estatístico

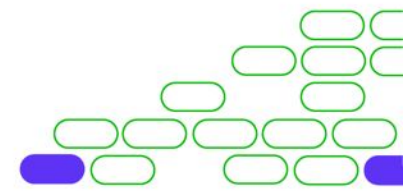
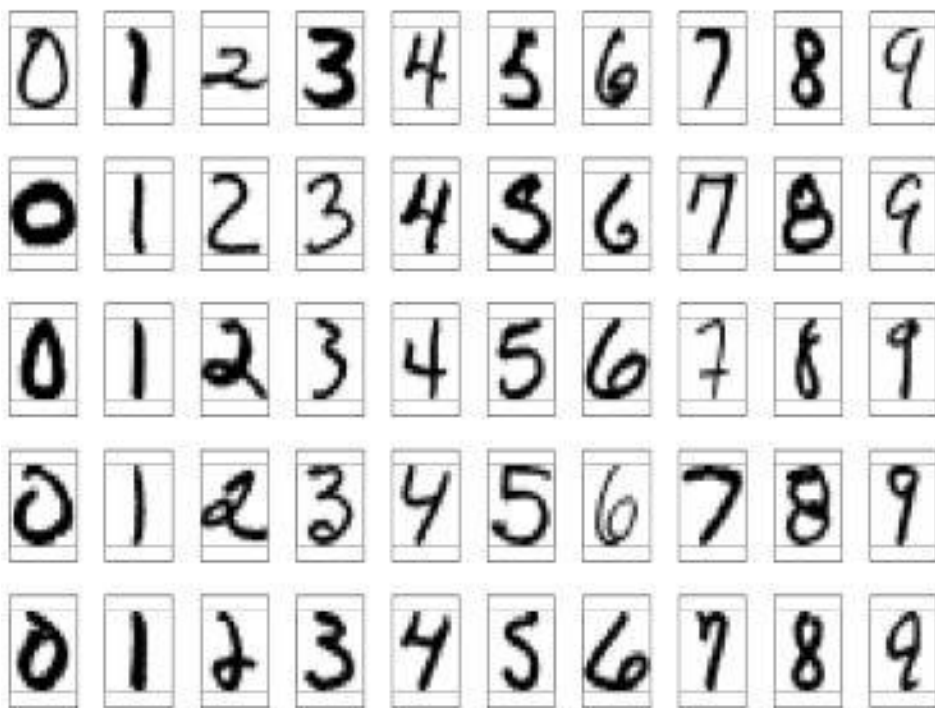
- Aprendizado a partir de dados.
- Inferências a partir de experiências passadas.

“Seu modelo é tão bom quanto forem os dados que o alimentam...”




Exemplos de problemas

- Reconhecimento de dígitos escritos a mão:

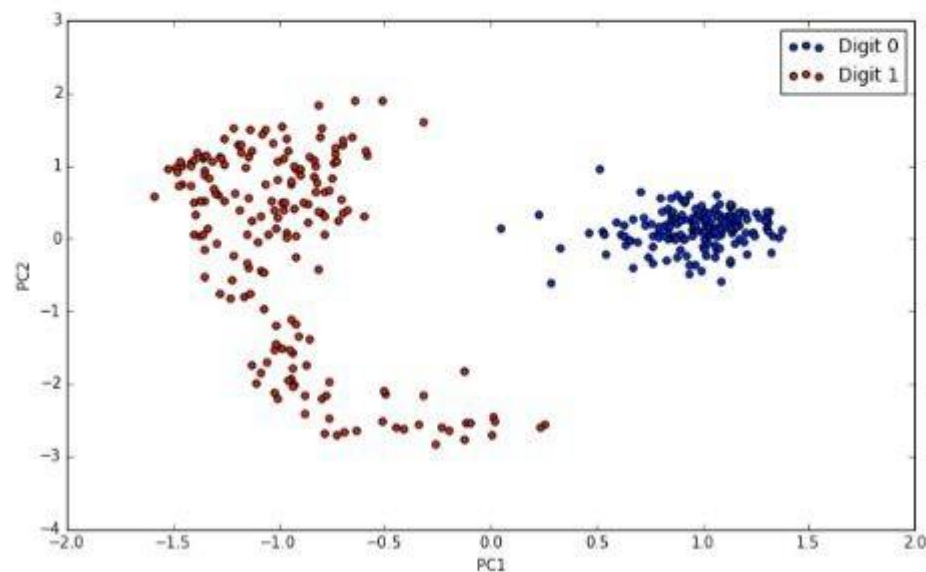


Exemplos de problemas

- Reconhecimento de dígitos escritos a mão:

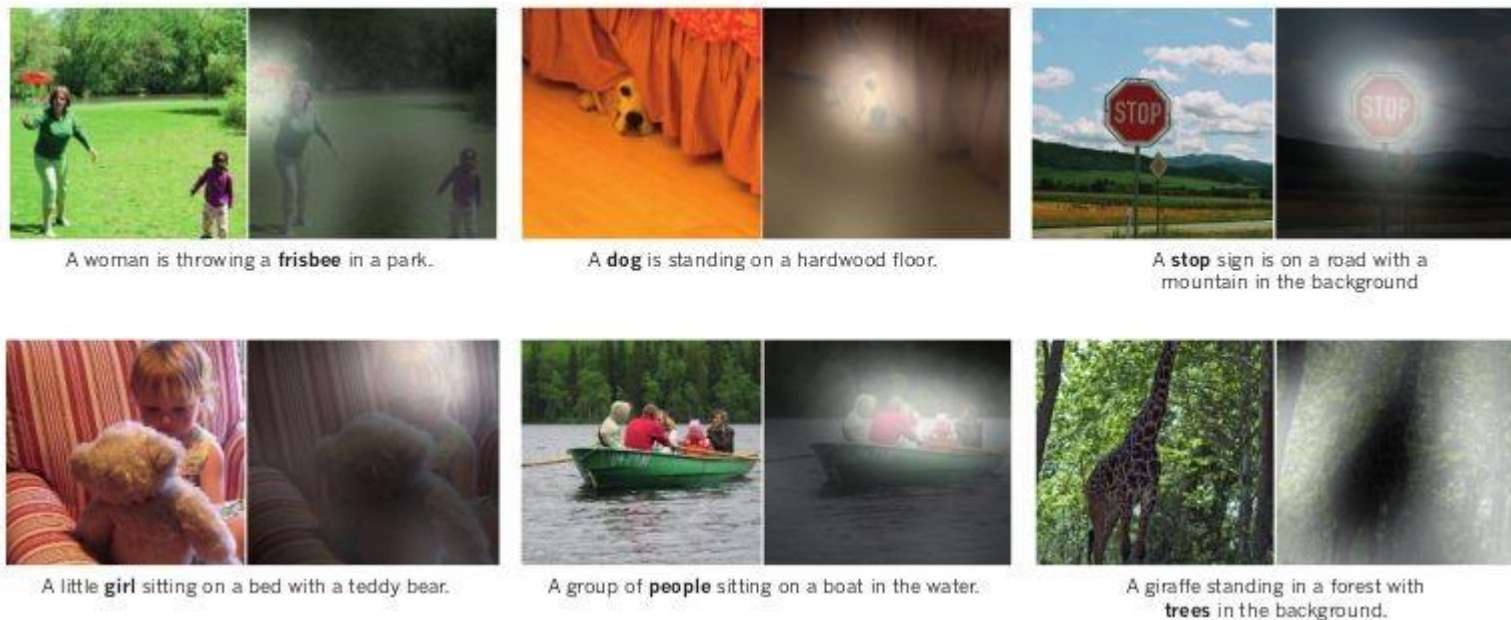
 →

0.	0.	5.	13.	9.	1.	0.	0.	0.	0.	13.	15.	10.	15.	5.
0.	0.	3.	15.	2.	0.	11.	8.	0.	0.	4.	12.	0.	0.	8.
8.	0.	0.	5.	8.	0.	0.	9.	8.	0.	0.	4.	11.	0.	1.
12.	7.	0.	0.	2.	14.	5.	10.	12.	0.	0.	0.	0.	6.	13.
10.	0.	0.	0.											



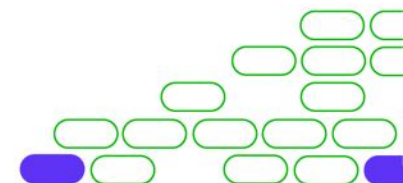
Exemplos de problemas

- Geração automática de legendas de fotos:

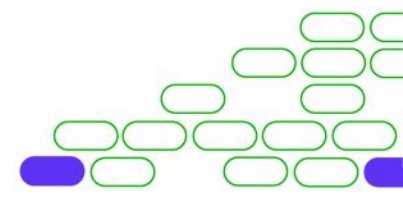
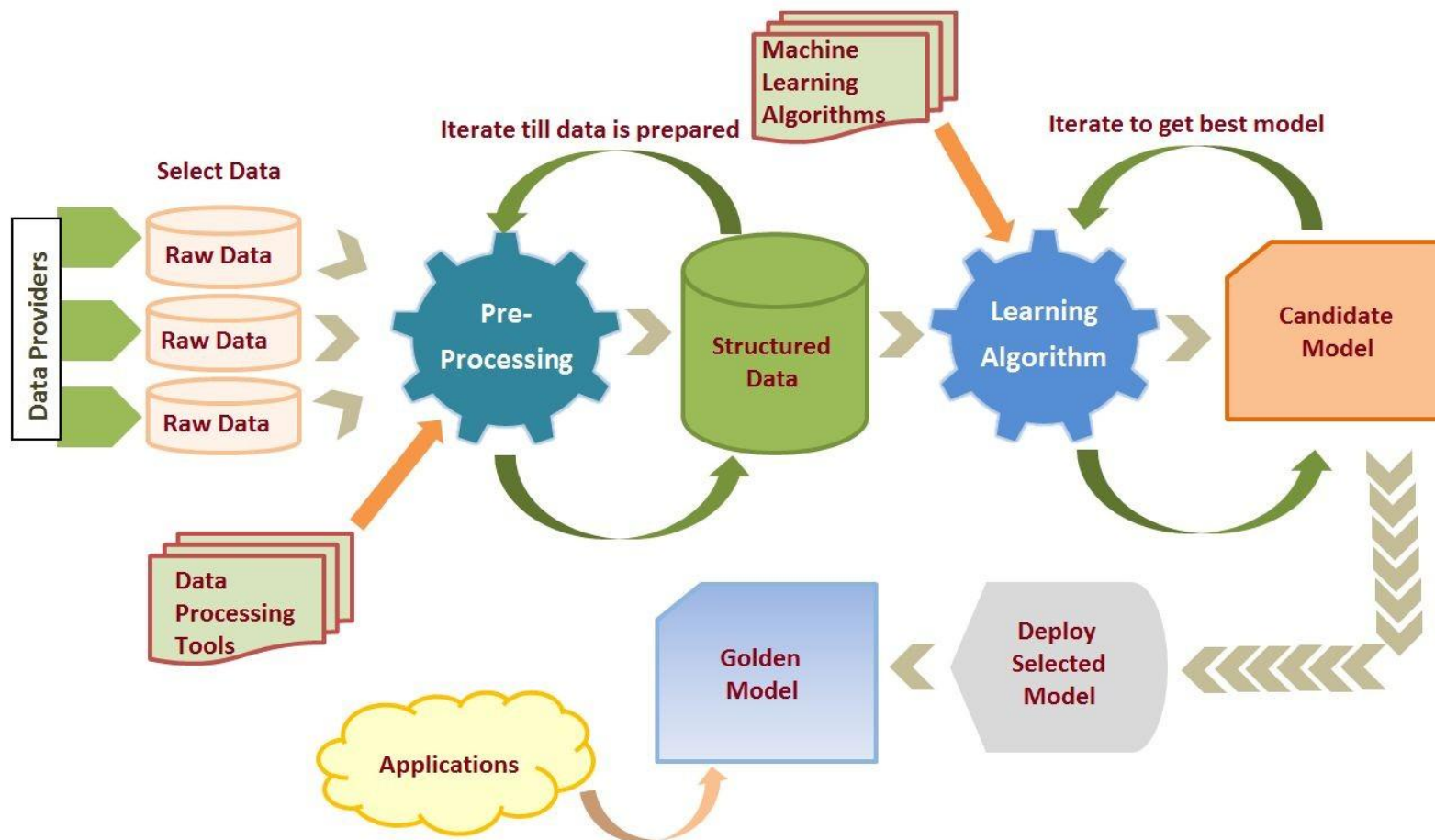


Fonte:

<https://arxiv.org/abs/1502.03044>



Pipeline



Conclusão

- ✓ Aprendizado de máquinas:
 - ✓ Diferença entre abordagem simbólica e o aprendizado estatístico.
 - ✓ Resultados recentes.
 - ✓ Pipeline de um projeto de dados.



Referências

- ❑ **An Introduction to Statistical Learning: With Applications in R: 103,** por Gareth James. Disponível em: <http://www-bcf.usc.edu/~gareth/ISL/>
- ❑ **The Hundred-Page Machine Learning Book,** por Andriy Burkov. Disponível em: <http://themlbook.com/wiki/doku.php>
- ❑ PAIM, André. **Introdução à inteligência computacional:**
Apresentação da Disciplina. 01 jan. 2017, 01 jun. 2017. Notas de Aula.

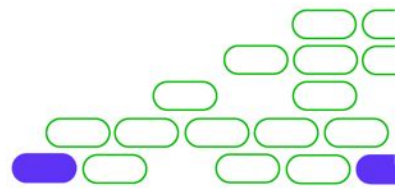


Próxima aula

- ❑ Introdução ao scikit-learn.



XPe





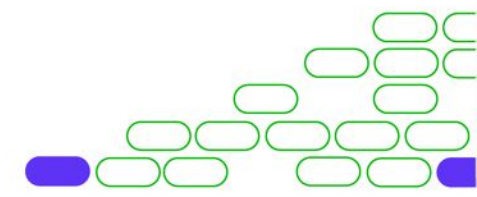
Faculdade



Introdução à Análise de Dados

AULA 4.2. INTRODUÇÃO AO
SCIKIT-LEARN

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Introdução ao scikit-learn.



Introdução ao scikit-learn

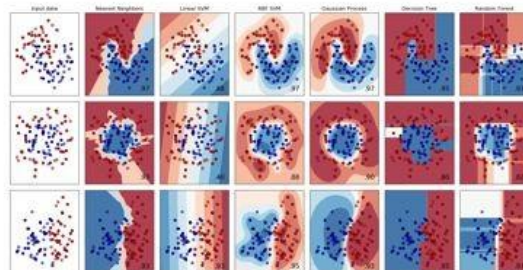
O [scikit-learn](https://scikit-learn.org) é um dos mais utilizados frameworks de aprendizado de máquinas em Python:

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

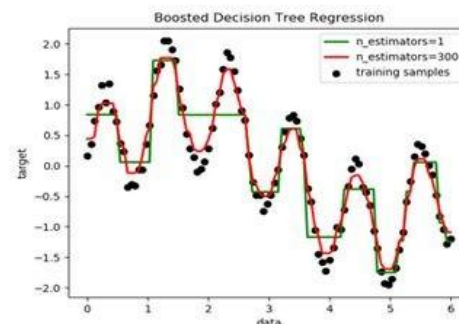


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Introdução ao scikit-learn

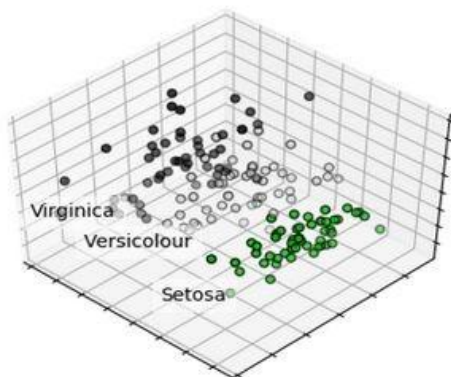
O [scikit-learn](https://scikit-learn.org) é um dos mais utilizados frameworks de aprendizado de máquinas em Python:

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

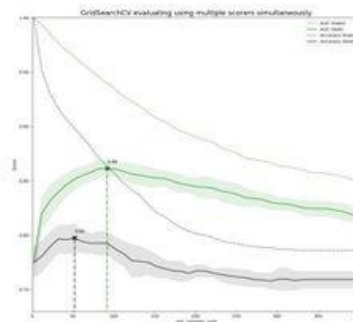


Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...



Introdução ao scikit-learn

O [scikit-learn](#) é um dos mais utilizados frameworks de aprendizado de máquinas em Python:

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross

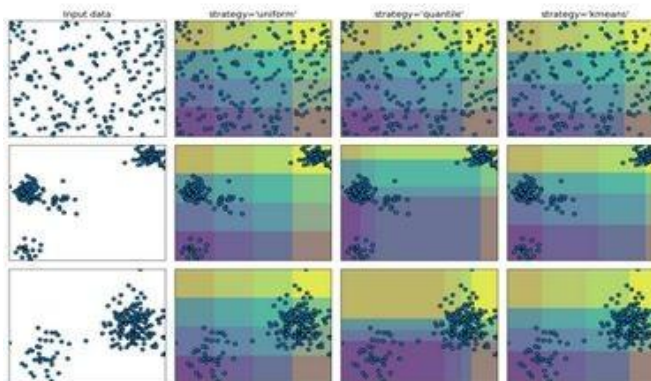


Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...



Introdução ao scikit-learn

- Open-source.



Introdução ao scikit-learn

- Open-source.
- Desenvolvido baseado no numpy, scipy e matplotlib.



Introdução ao scikit-learn

- Open-source.
- Desenvolvido baseado no numpy, scipy e matplotlib.
- Interface alto-nível de modelos complexos.



Introdução ao scikit-learn

- Open-source.
- Desenvolvido baseado no numpy, scipy e matplotlib.
- Interface alto-nível de modelos complexos.
- Instalação:
 - `pip install scikit-learn`.
 - `conda install scikit-learn`.



Introdução ao scikit-learn

- Open-source.
- Desenvolvido baseado no numpy, scipy e matplotlib.
- Interface alto-nível de modelos complexos.
- Instalação:
 - `pip install scikit-learn.`
 - `conda install scikit-learn.`
- Uso:

```
# pré-processamento
from sklearn.preprocessing import LabelEncoder

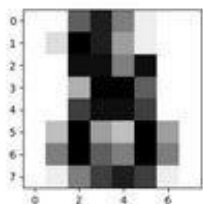
# modelo
from sklearn.linear_model import LogisticRegression
```

Introdução ao scikit-learn

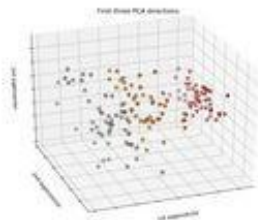
Possui diversos datasets disponíveis.

Dataset examples ¶

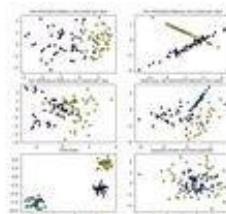
Examples concerning the `sklearn.datasets` module.



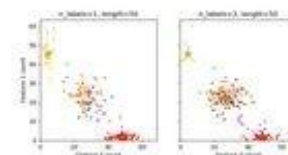
The Digit Dataset



The Iris Dataset



Plot randomly generated
classification dataset



Plot randomly generated
multilabel dataset

Introdução ao scikit-learn

Execução de um modelo complexo em poucas linhas:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

xx, yy = np.meshgrid(np.linspace(-3, 3, 500),
                     np.linspace(-3, 3, 500))

np.random.seed(0)
X = np.random.randn(300, 2)
Y = np.logical_xor(X[:, 0] > 0, X[:, 1] > 0)

# fit the model
clf = svm.NuSVC(gamma='auto')
clf.fit(X, Y)
```



Conclusão

- ✓ Visão geral do scikit-learn.



Referências

- ❑ **Scikit-learn.** Disponível em:
<<https://scikit-learn.org/stable/>>. Acesso em: 14 de
jul. de 2020.

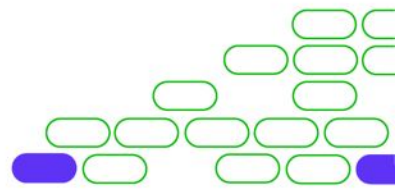


Próxima aula

- ❑ Classificação: conceitos básicos.



XPe





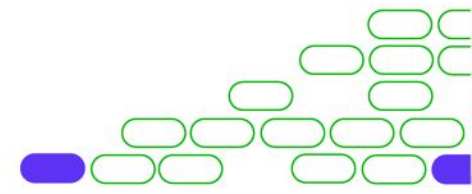
Faculdade



Introdução à Análise de Dados

**AULA 4.3. CLASSIFICAÇÃO DE PADRÕES:
CONCEITOS BÁSICOS**

PROF. MATHEUS MENDONÇA

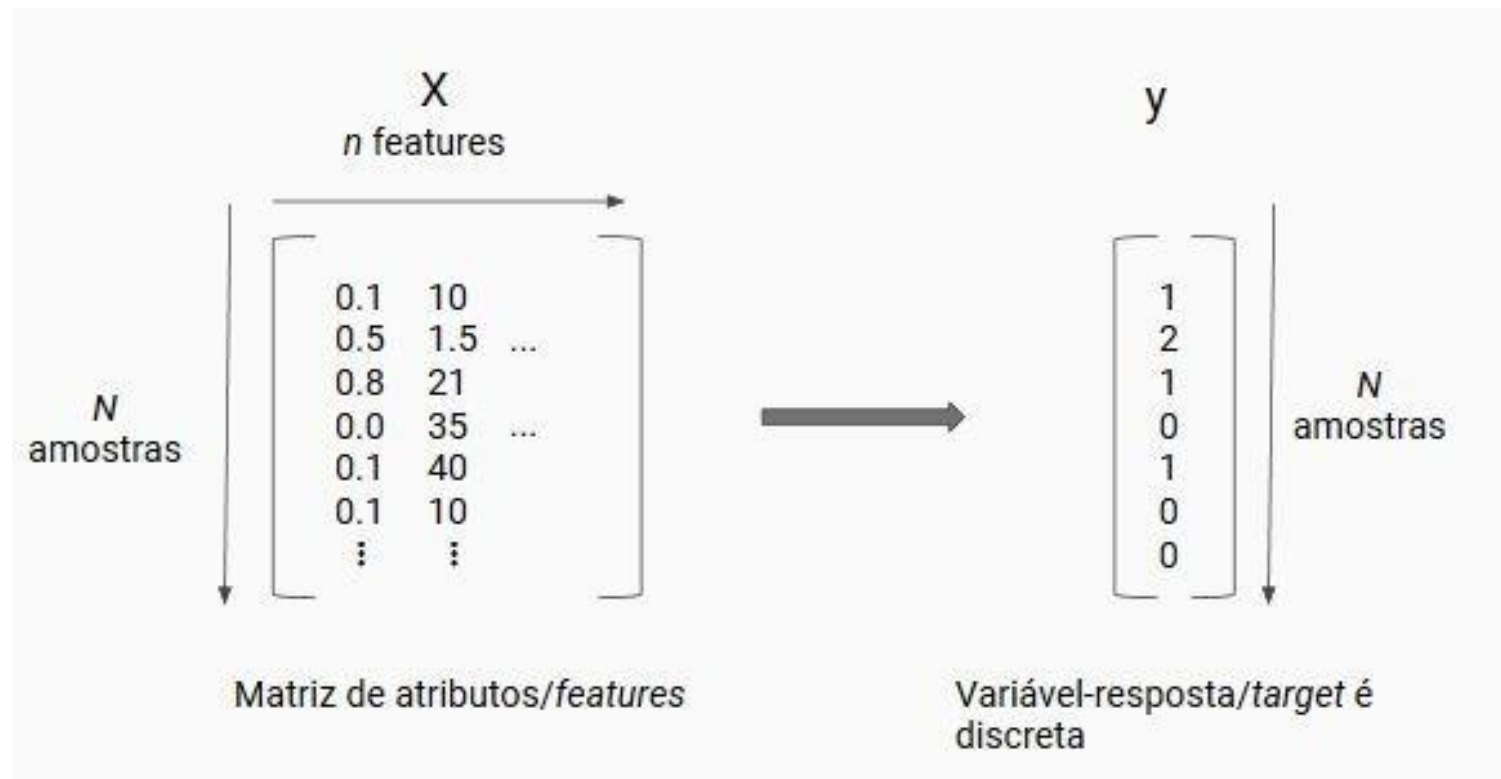


Nesta aula

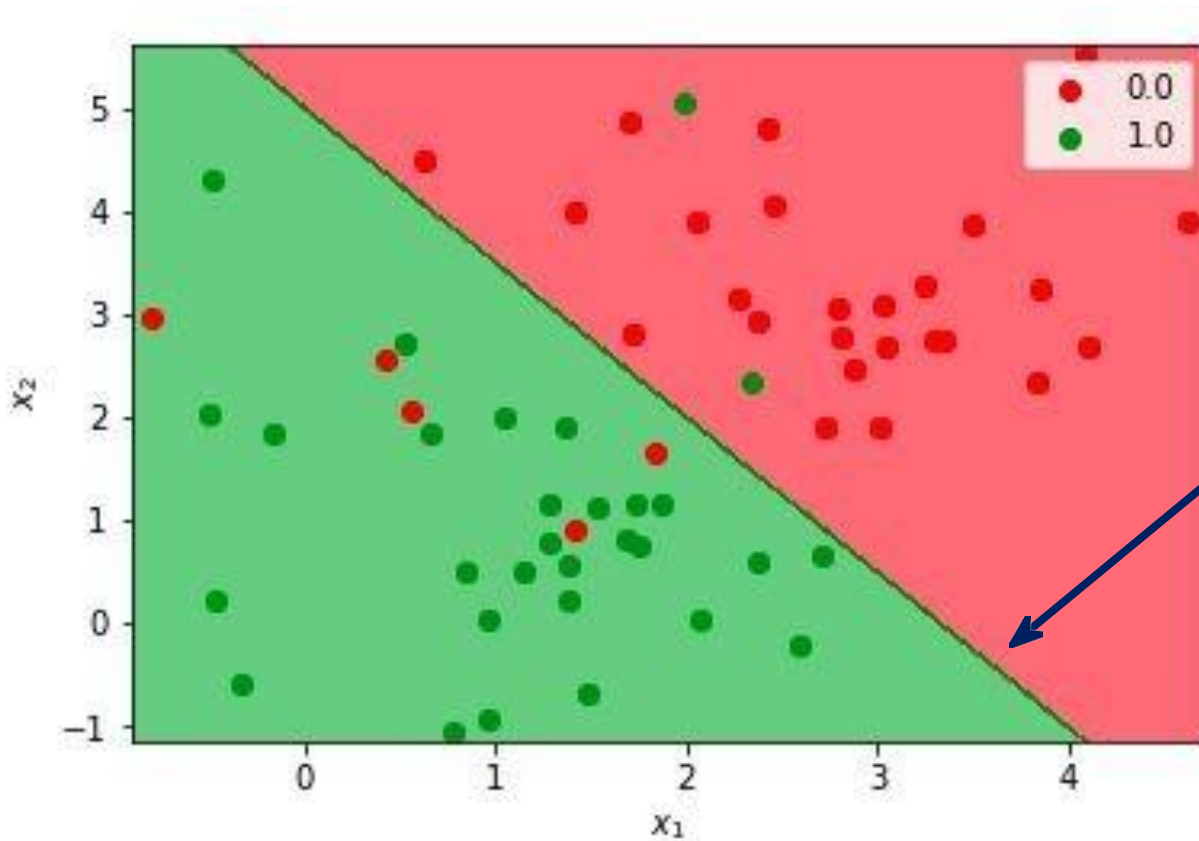
- ❑ Classificação de padrões: conceitos básicos.



Classificação



Classificação

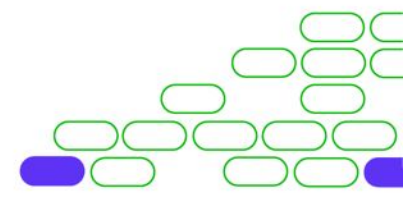


$g(X)$: Fronteira de decisão

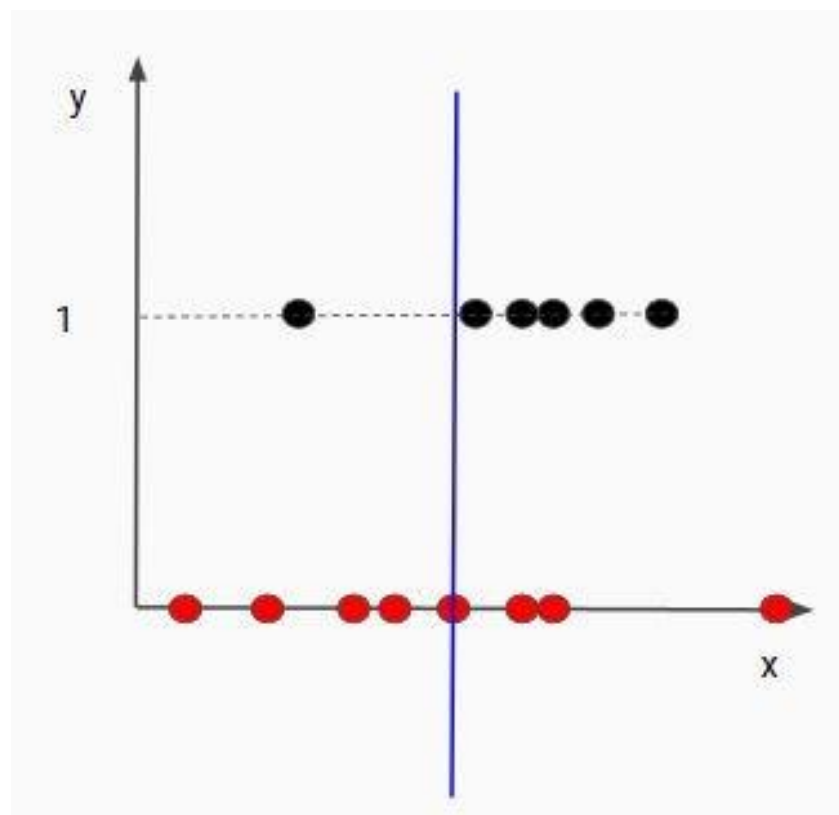
Classificação

- Um classificador binário se resume a:

$$y = \begin{cases} 1, & \text{caso } g(X) > \text{limiar} \\ 0, & \text{caso contrário} \end{cases}$$



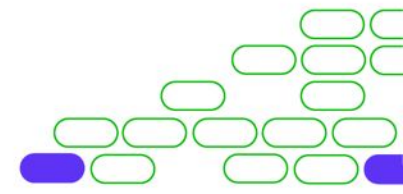
Fronteira de decisão



Conclusão

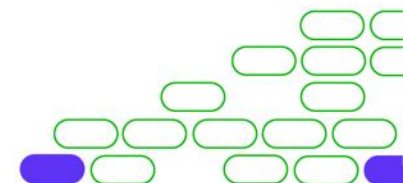
✓ Classificação de padrões:

✓ Conceitos básicos.



Referências

- ❑ **An Introduction to Statistical Learning: With Applications in R: 103,** por Gareth James. Disponível em: <http://www-bcf.usc.edu/~gareth/ISL/>
- ❑ **The Hundred-Page Machine Learning Book,** por Andriy Burkov. Disponível em: <http://themlbook.com/wiki/doku.php>
- ❑ PAIM, André. **Introdução à inteligência computacional: Apresentação da Disciplina.** 01 jan. 2017, 01 jun. 2017. Notas de Aula.



Próxima aula

- ❑ Classificação no scikit-learn – Prática.





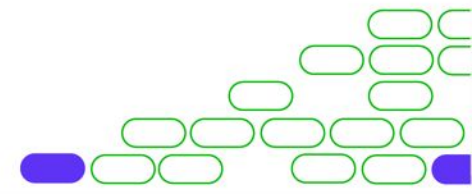
Faculdade



Introdução à Análise de Dados

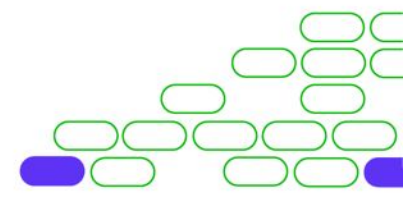
**AULA 4.4. CLASSIFICAÇÃO NO SCIKIT-LEARN
(PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Classificação de padrões no scikit-learn.



Conclusão

- ✓ Classificação de padrões no scikit-learn.



Referências

- ❑ **An Introduction to Statistical Learning: With Applications in R: 103,** por Gareth James. Disponível em: <http://www-bcf.usc.edu/~gareth/ISL/>
- ❑ **The Hundred-Page Machine Learning Book,** por Andriy Burkov. Disponível em: <http://themlbook.com/wiki/doku.php>
- ❑ **PAIM, André. Introdução à inteligência computacional: Apresentação da Disciplina.** 01 jan. 2017, 01 jun. 2017. Notas de Aula.



Próxima aula

- ❑ Regressão linear no scikit-learn I (Prática).





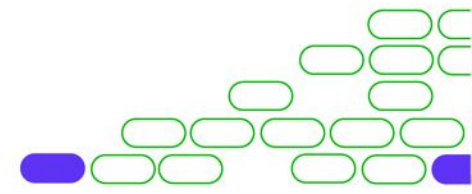
Faculdade



Introdução à Análise de Dados

**AULA 4.5. REGRESSÃO LINEAR NO SCIKIT-LEARN
I (PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

- ❑ Regressão linear no scikit-learn I.



Conclusão

✓ Regressão linear no
scikit-learn.



Referências

- ❑ **An Introduction to Statistical Learning: With Applications in R: 103,** por Gareth James. Disponível em: <http://www-bcf.usc.edu/~gareth/ISL/>
- ❑ **The Hundred-Page Machine Learning Book,** por Andriy Burkov. Disponível em: <http://themlbook.com/wiki/doku.php>
- ❑ **PAIM, André. Introdução à inteligência computacional: Apresentação da Disciplina.** 01 jan. 2017, 01 jun. 2017. Notas de Aula.



Próxima aula

- ❑ Regressão linear no scikit-learn II (Prática).





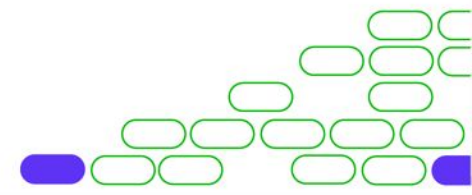
Faculdade



Introdução à Análise de Dados

**AULA 4.6. REGRESSÃO LINEAR NO SCIKIT-LEARN
II (PRÁTICA)**

PROF. MATHEUS MENDONÇA



Nesta aula

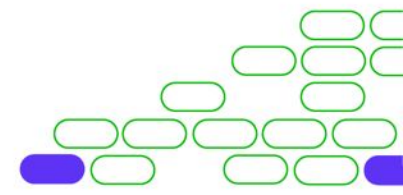
- ❑ Regressão linear no scikit-learn.



Conclusão

✓ Regressão linear no
scikit-learn:

✓ Métricas de avaliação.



Referências

- ❑ **An Introduction to Statistical Learning: With Applications in R: 103,** por Gareth James. Disponível em: <http://www-bcf.usc.edu/~gareth/ISL/>
- ❑ **The Hundred-Page Machine Learning Book,** por Andriy Burkov. Disponível em: <http://themlbook.com/wiki/doku.php>
- ❑ PAIM, André. **Introdução à inteligência computacional: Apresentação da Disciplina.** 01 jan. 2017, 01 jun. 2017. Notas de Aula.



Próxima aula

□ Fim!

