

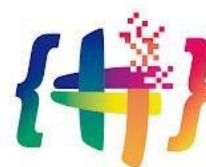


INTENSIVÃO DE PYTHON {#}

100% ONLINE & GRATUITO

Apostila Completa Aula 4

Aprenda a criar modelos de previsão usando
Ciência de Dados.
Impressionador do absoluto zero!



Parte 1

Introdução

O que vamos aprender

Na quarta aula do Intensivão de Python você vai aprender a criar um código de ciência de dados para criação de um modelo de previsão do **absoluto zero**. Para isso, vamos passar por conceitos como:

Jupyter Notebook

Etapas de um
projeto de Ciência
de Dados

Gráficos usando
Python

Uso de bibliotecas

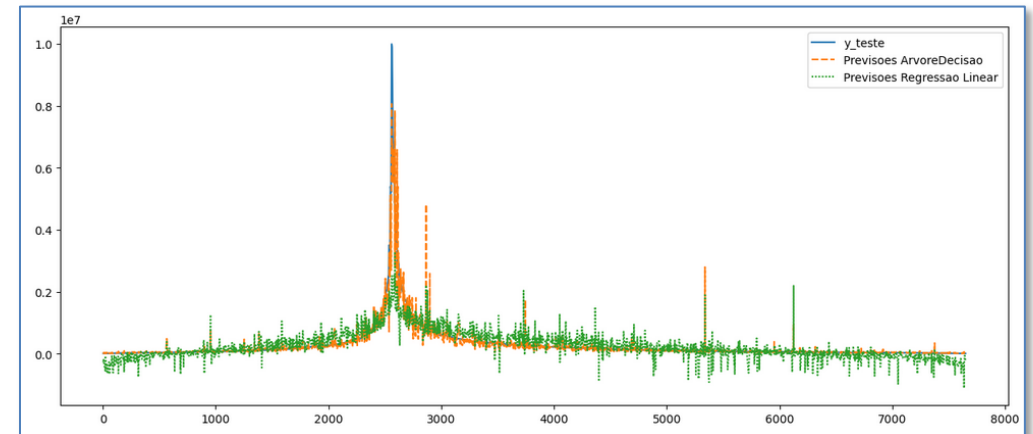
Modelos de
Inteligência Artificial

Após todos esses conhecimentos, seremos capazes de transformar uma tabela cheia de informações, nem um pouco fáceis de serem interpretadas ...

	A	B	C	D	E	F
1	Preco,TipoBarco,TipoVenda,Ano,Comprimento,Largura,Material					
2	3490	1,1	2020	4,1	5,1	
3	3399	2,1	2019	3.55	1.46	2
4	3600	3,2	1999	6.2	2.38	2
5	3500	2,2	1987	4.35	1.75	3
6	3500	4,2	2004	4.7	2,3	
7	4600	5,1	2020	4.4	1.65	2
8	4500	2,2	1997	3.72	1.33	2
9	4000	6,2	1981	8.2	84,4	
10	4000	7,2	2011	6.37	2.31	3
11	3999	2,1	2019	3.55	1.46	2
12	3930	4,1	2020	4.31	1.64	2
13	3900	4,2	1979	5.6	2.1	3



... em um modelo de previsão de preço utilizando Inteligência Artificial



Entendendo a base de dados

As informações que vão alimentar o nosso código, são informações de barcos para que possamos fazer a previsão de preços com base nessas informações.

A imagem ao lado, mostra base "crua". Apesar de não estar formatada de uma forma que nos ajuda a visualização, podemos perceber que existem 7 colunas (indicadas abaixo) separadas por vírgula:

Preço – TipoBarco – TipoVenda – Ano – Comprimento – Largura – Material

Temos muitas informações, mas vamos separá-las para facilitar o entendimento e verificar como podemos fazer a previsão de preço com base nessas informações.

Vamos então fazer a extração dos dados, fazer o tratamento adequado, fazer a análise exploratória, fazer a modelagem + algoritmos (inteligência artificial) e por fim vamos fazer a interpretação dos resultados!

	A	B	C	D	E	F
1	Preco,TipoBarco,TipoVenda,Ano,Comprimento,Largura,Material					
2	3490,1,1,2020,4,1.5,1					
3	3399,2,1,2019,3.55,1.46,2					
4	3600,3,2,1999,6.2,2.38,2					
5	3500,2,2,1987,4.35,1.75,3					
6	3500,4,2,2004,4.7,2,3					
7	4600,5,1,2020,4.4,1.65,2					
8	4500,2,2,1997,3.72,1.33,2					
9	4000,6,2,1981,8,2.84,4					
10	4000,7,2,2011,6.37,2.31,3					
11	3999,2,1,2019,3.55,1.46,2					
12	3930,4,1,2020,4.31,1.64,2					
13	3900,4,2,1979,5.6,2.1,3					

Entendendo a base de dados

Essa é uma imagem que está no código para que você possa entender o que são cada um dos números nas colunas, pois só saber que o material do primeiro barco é 1 não significa nada se não tivermos como traduzir essa informação.

Então aqui temos a tabela de materiais, tipo de venda e o tipo de barco para que fique mais fácil entender essas informações.

Lembrando que isso é necessário, pois no modelo não conseguimos trabalhar com informações de texto, por isso temos que ter essas informações no formato de número para que os algoritmos consigam trabalhar.

Materiais		TipoVenda		TipoBarco	
Thermoplastic	1	new boat from stock	1	Center console boat	1
Aluminium	2	Used boat	2	Fishing Boat	2
GRP	3	new boat on order	3	Catamaran	3
PVC	4	Display Model	4	Sport Boat	4
Plastic	5			Runabout	5
Wood	6			Pilothouse	6
Steel	7			Cabin Boat	7
Hypalon	8			Deck Boat	8
Carbon Fiber	9			Pontoon Boat	9
Reinforced concrete	10			Classic	10
Rubber	11			Bowrider	11
				Trawler	12
				Motor Yacht	13
				Working Boat	14
				Launch	15
				Flybridge	16
				Water ski	17
				Hardtop	18
				Offshore Boat	19
				House Boat	20
				Passenger boat	21
				Wakeboard/Wakesurf	22
				Ketch	23
				Mega Yacht	24
				Motorsailer	25

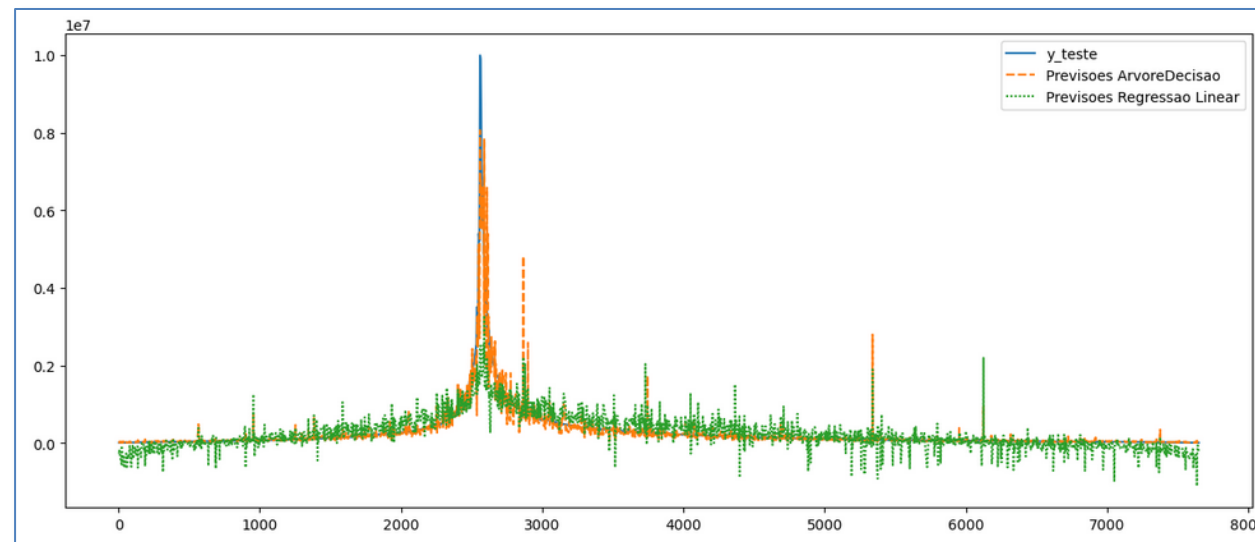
Entendendo a solução final

Nosso objetivo é criar um modelo de previsão de preço a partir do histórico de preço desses barcos.

Para isso, usaremos alguns conhecimentos de Ciência de Dados.

Baseado nesse modelo, poderemos projetar (com algum grau de incerteza) o preço desses barcos de acordo com as características deles.

A imagem inferior mostra essa previsão de preço com essas informações.



	TipoBarco	TipoVenda	Ano	Comprimento	Largura	Material
0	2	3	2023	3.0	1.5	1
1	2	1	2023	3.8	1.9	2
2	3	2	1999	6.2	3.0	2
[5979.2655 8816.2875 16208.65]						

Parte 2

O que é um projeto de Ciência de Dados

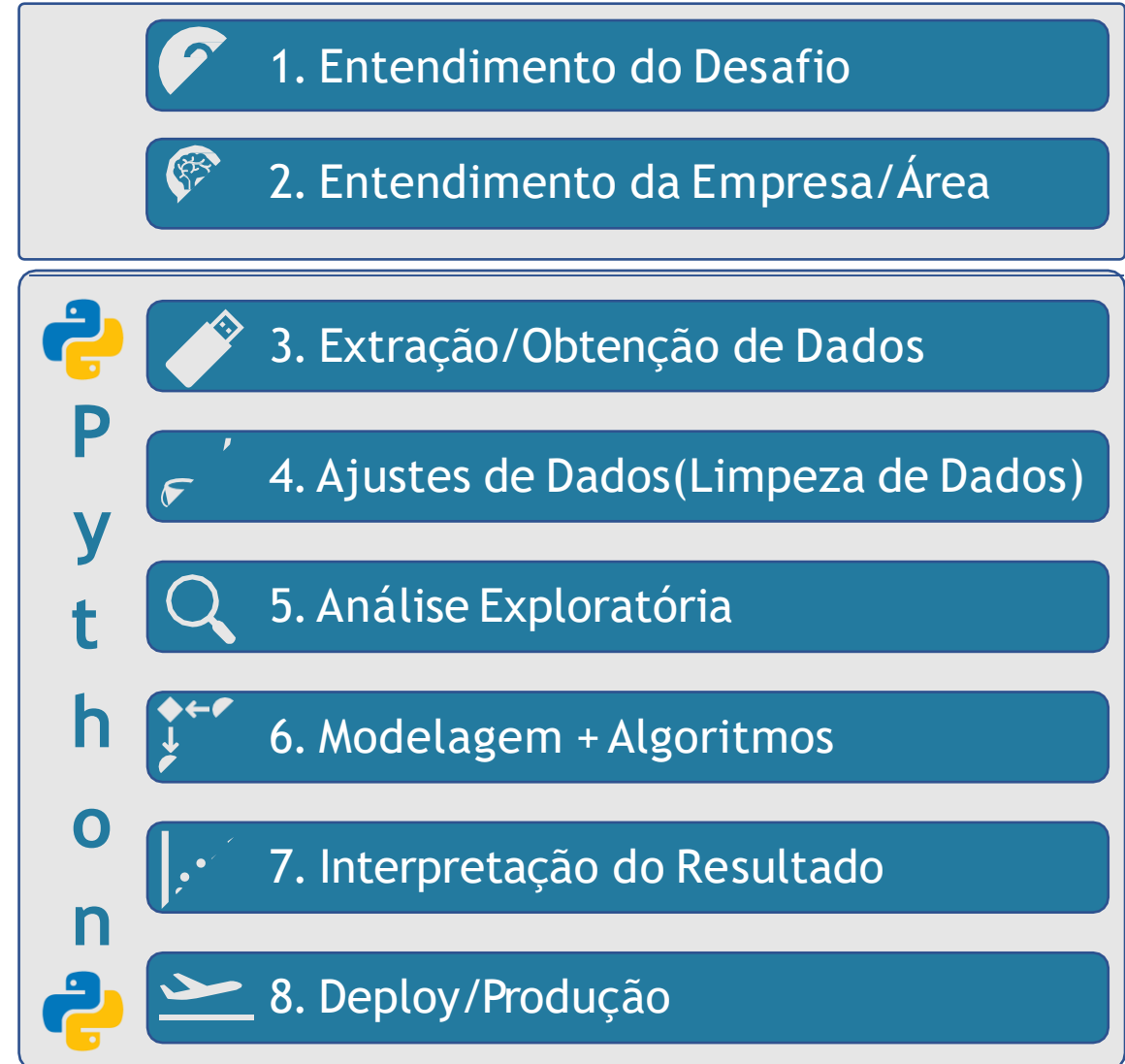
O que é um projeto de Ciência de dados

O conceito de Ciência de Dados cada dia passa a ser mais difundido nas empresas. Essencialmente, esse conceito, tem por objetivo gerar conclusões, Modelos, a partir de grande quantidade de dados.

Através de conhecimentos estatísticos alinhados ao conhecimento do negócio, a Ciência de dados permite “enxergar” além da superfície.

Para que isso aconteça, é muito interessante seguir alguns passos desse processos.

Preparamos um diagrama simples para você 😊



Parte 3

Jupyter Notebook

O que é? Como acesso?

Os códigos em Python precisam de uma plataforma para serem escritos.

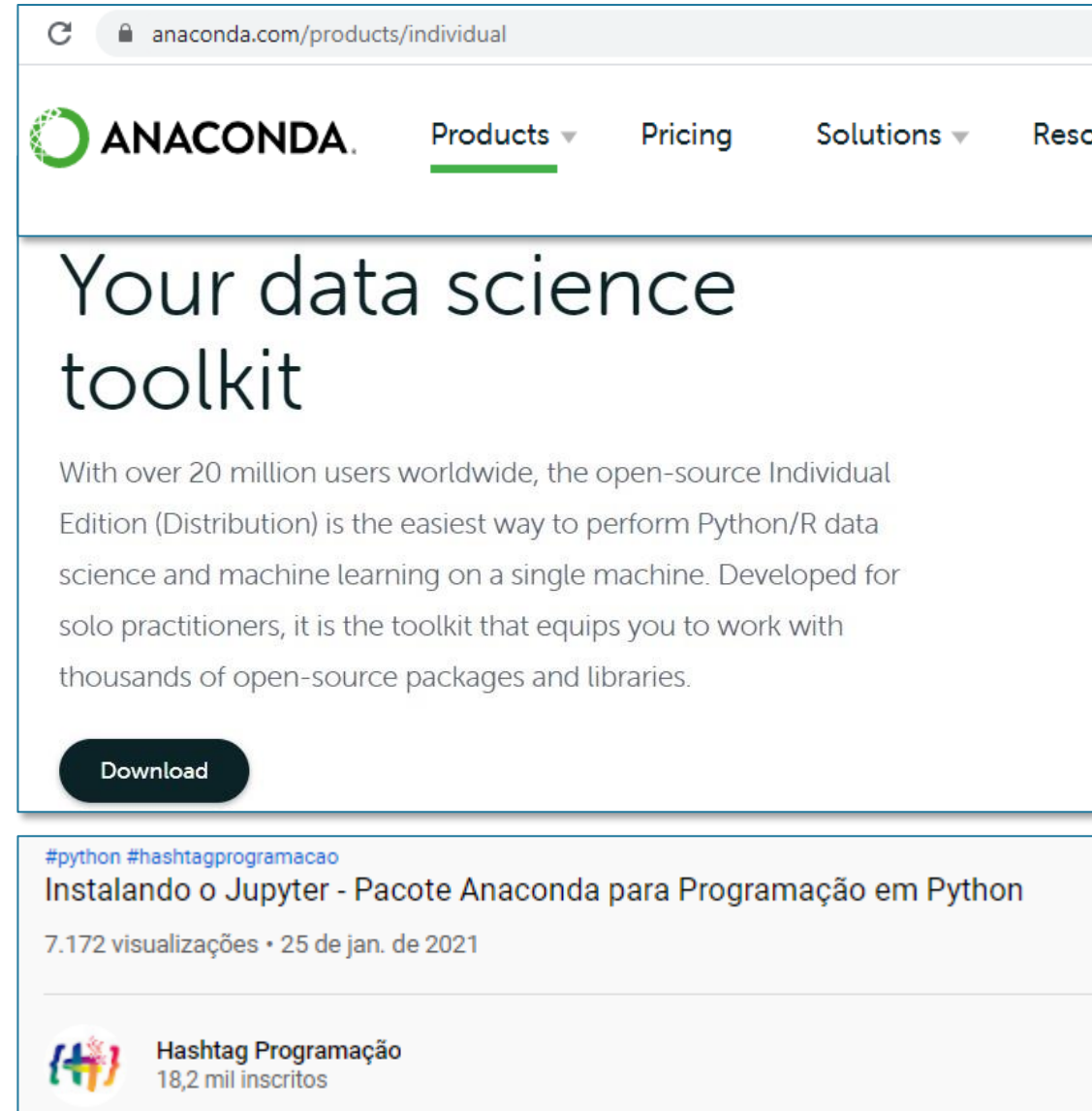
Essas plataformas na programação são chamadas de IDEs. Existem várias: Visual Studio, PyCharm, Atom, Google Colab, etc...

Todas podem ser utilizadas pra programação e execução dos códigos, usaremos o Jupyter Notebook que é uma ferramenta gratuita que existe dentro do Anaconda (uma espécie de grande caixa de ferramentas do Python).

Para usarmos o Jupyter Notebook, iremos instalar o Anaconda [\(Link\)](#).

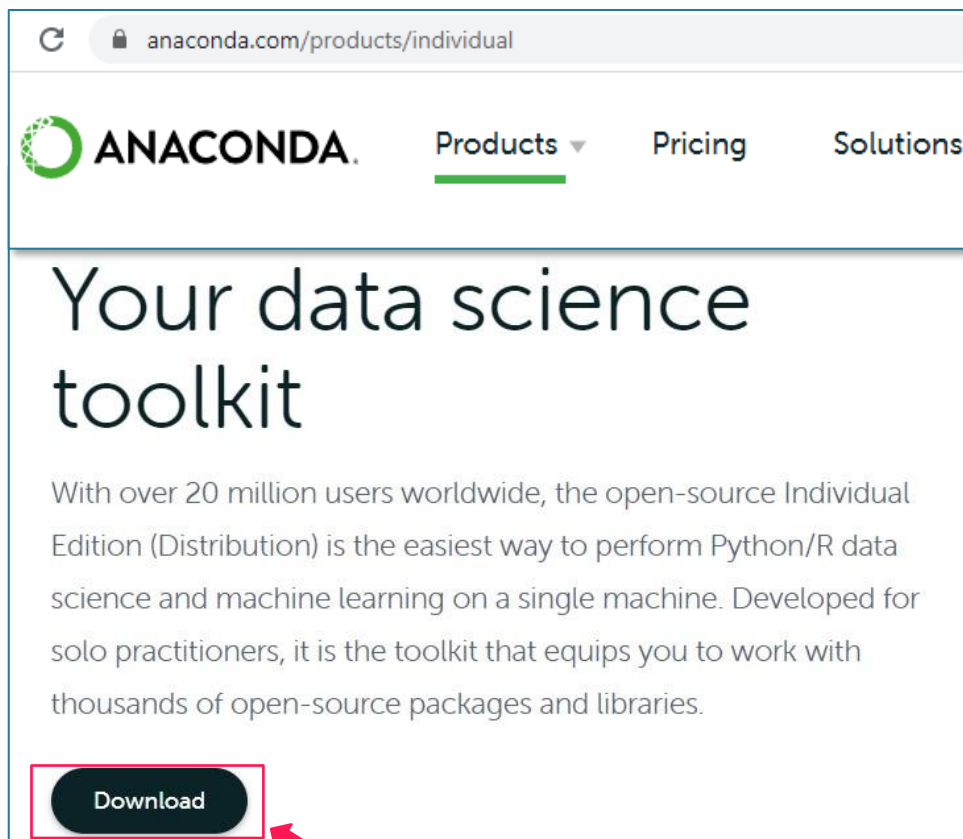
As próximas páginas desse capítulo são todos os passos para instalação correta do Jupyter mas caso você prefira, pode acessar nosso vídeo no Youtube explicando esse mesmo passo a passo:

[Instalando o Jupyter – Pacote Anaconda para Programação em Python](#)



The screenshot shows the Anaconda website's 'Products' page. The header includes the Anaconda logo and navigation links: Products, Pricing, Solutions, and Resources. The main content area features the heading 'Your data science toolkit' followed by a descriptive paragraph about the Individual Edition (Distribution) and a 'Download' button. Below this, there is a video thumbnail with the title 'Instalando o Jupyter - Pacote Anaconda para Programação em Python', showing 7,172 views and dated 25 de jan. de 2021. The video is associated with the hashtag #python and #hashtagprogramacao, and the channel 'Hashtag Programação' is mentioned with 18,2 mil inscritos.

1) No link indicado, clique em **Download**



anaconda.com/products/individual

ANACONDA. Products Pricing Solutions

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

2) **Escolha** a opção adequada para seu computador

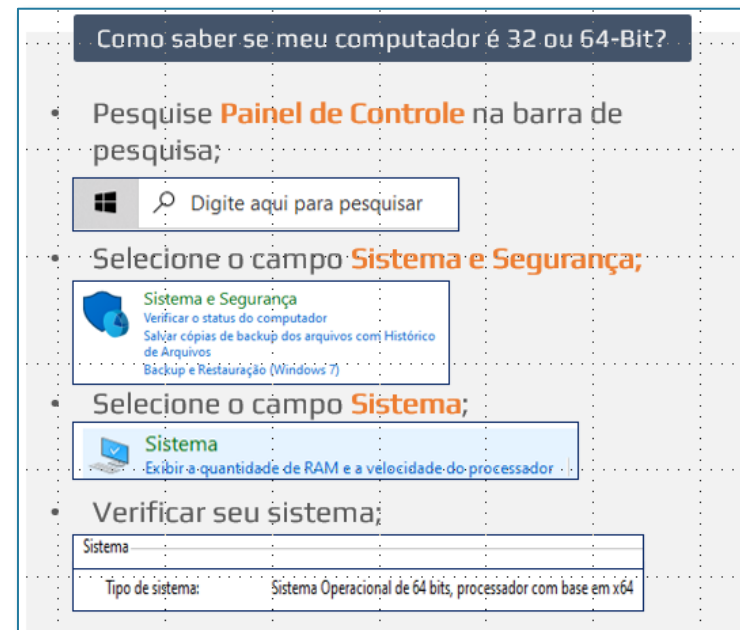


Windows

Python 3.8

64-Bit Graphical Installer (457 MB)

32-Bit Graphical Installer (403 MB)



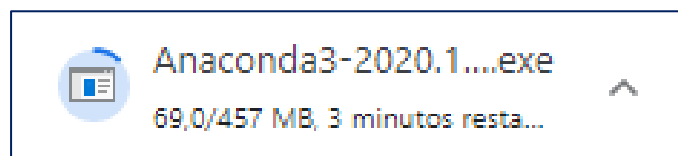
Como saber se meu computador é 32 ou 64-Bit?

- Pesquise **Painel de Controle** na barra de pesquisa;
- Selecione o campo **Sistema e Segurança**;
- Selecione o campo **Sistema**;
- Verificar seu sistema;

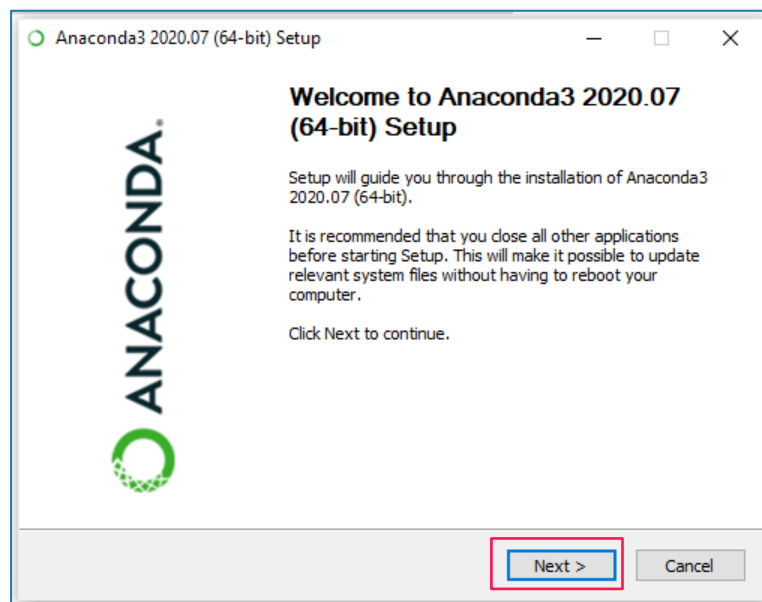
Tipo de sistema: Sistema Operacional de 64 bits, processador com base em x64

Jupyter Notebook Instalação

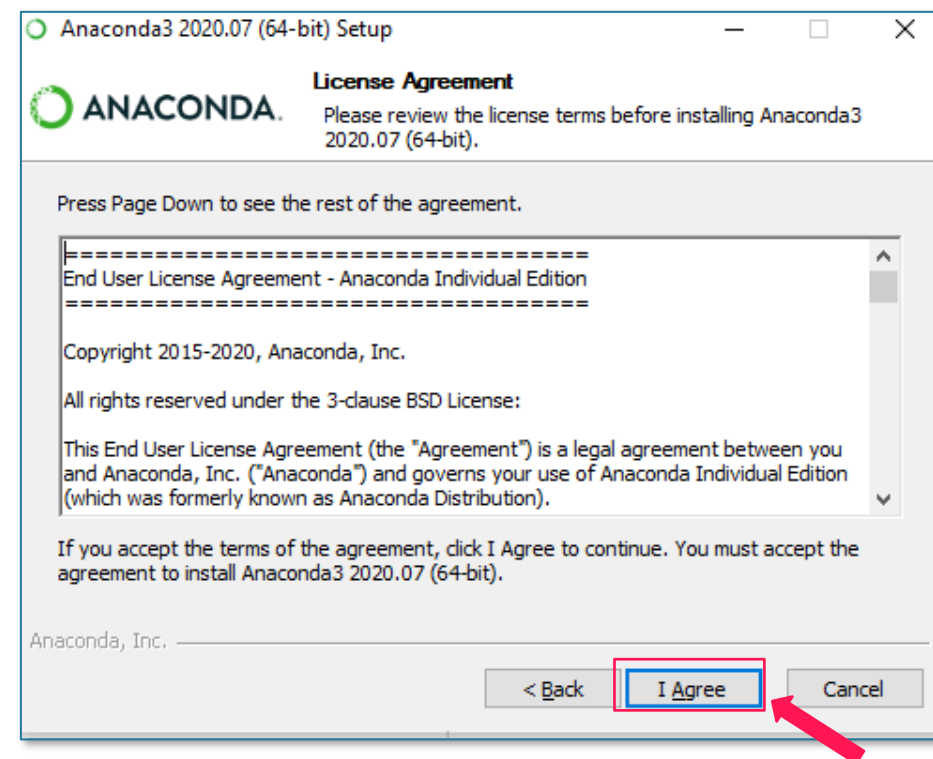
3) Fazer Download



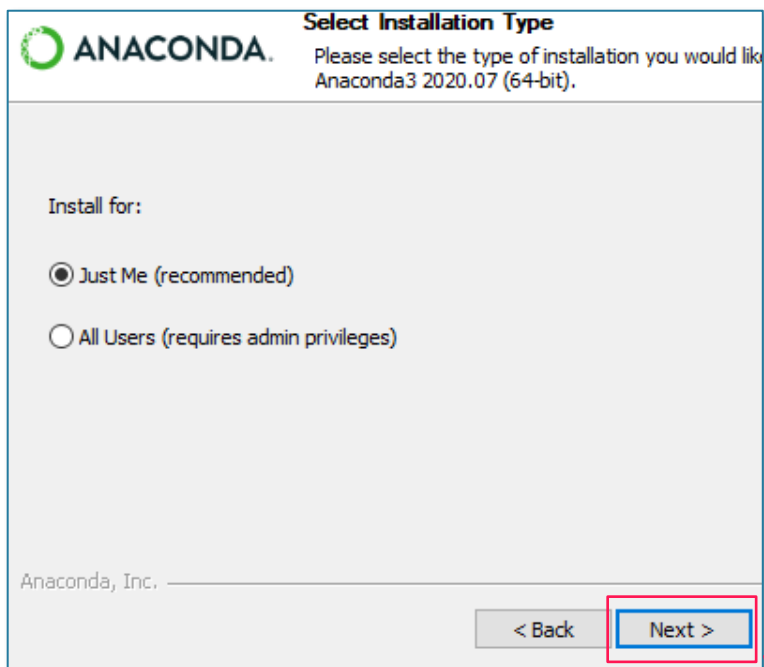
3) Abra o instalador do Anaconda



4) Aceite os termos de uso



5) No link indicado, clique em **Download**

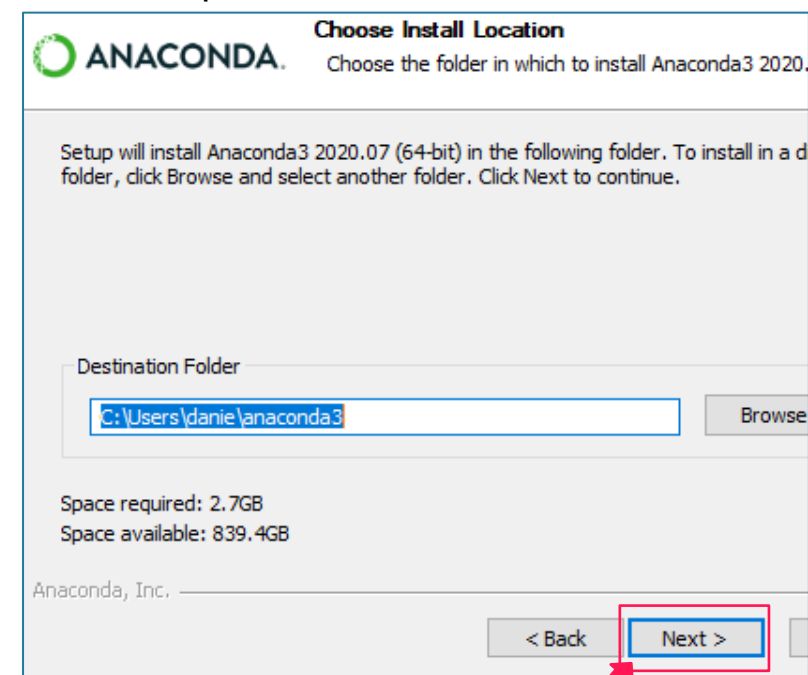


Nessa parte da instalação indicamos a opção **JUST ME** pois em teoria apenas o seu usuário precisa ter o Anaconda instalado.

PORÉM, em alguns casos, a instalação **JUST ME**, gera algumas falhas na inicialização do **JUPYTER NOTEBOOK** que vamos utilizar durante o curso.

Caso aconteça com você, reinstale e utilize a opção **ALL USERS**.

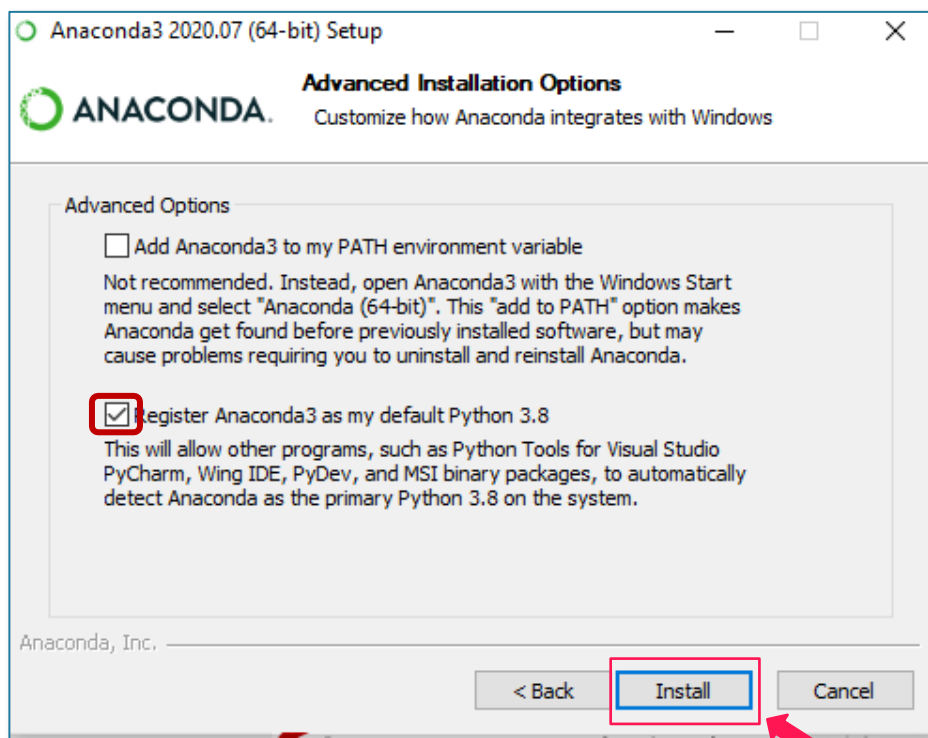
6) Aperte **Next** e siga a pasta padrão definida pelo Anaconda



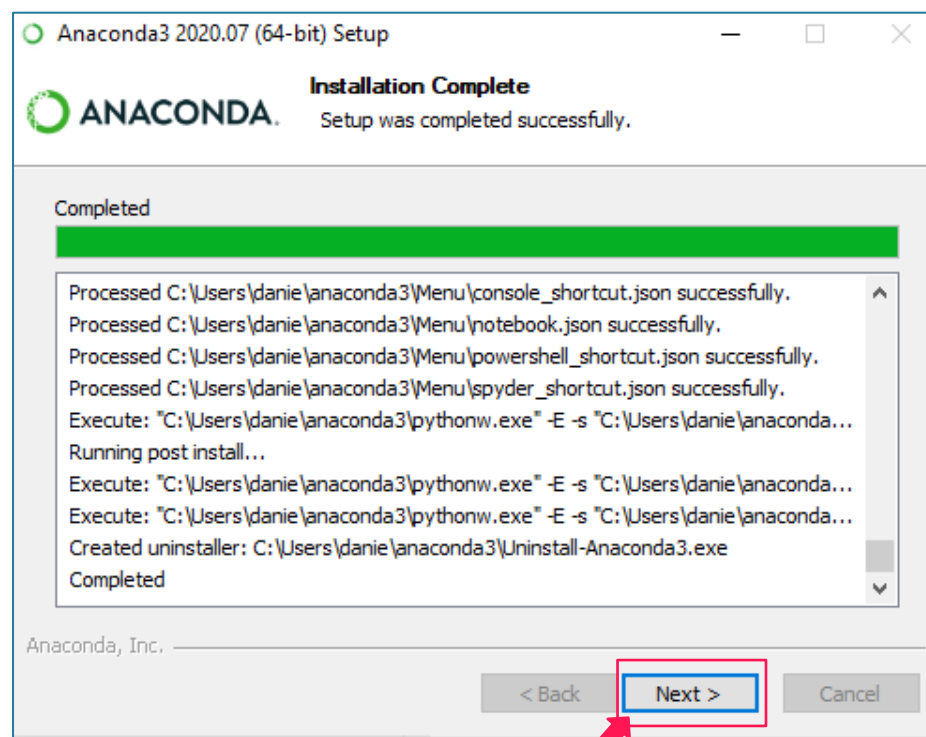
Jupyter Notebook

Instalação

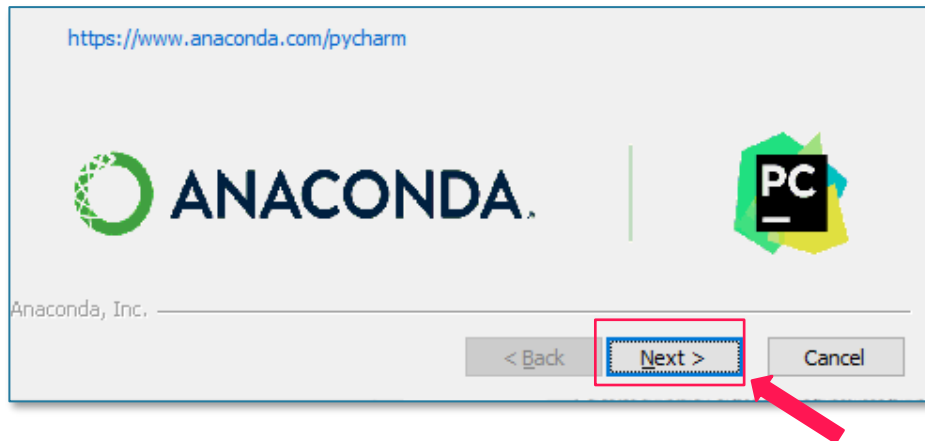
7) Defina o Anaconda como seu Python padrão e siga com a instalação clicando em **Install**



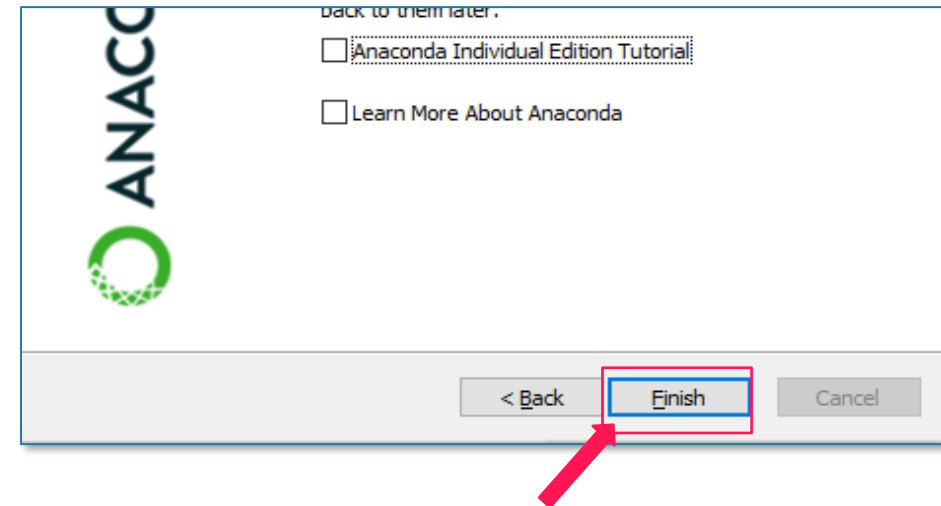
8) Ao fim da instalação clique em Next



9) Mais um **Next**



10) Clique em **Finish** para finalizar a instalação



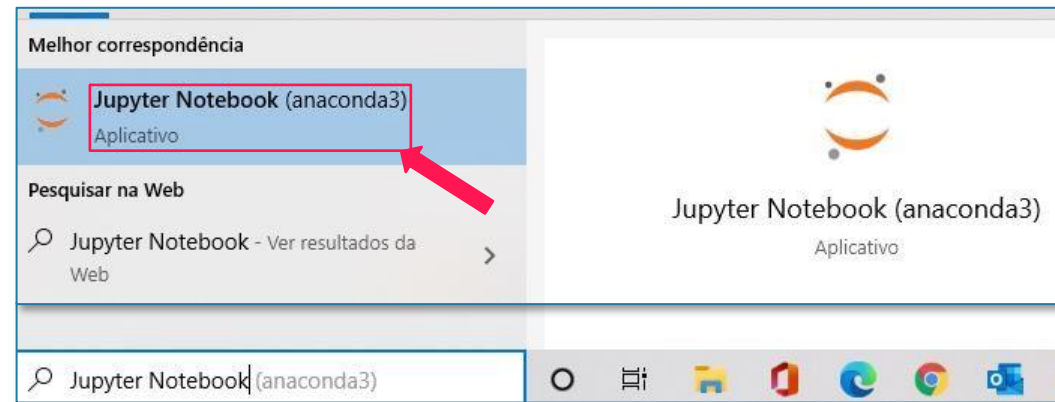
Pronto! Anaconda instalado. Agora vamos ver se está tudo OK para começarmos!

Jupyter Notebook

Inicializando o Jupyter

Para inicializarmos o Jupyter, basta digitar **Jupyter Notebook** no menu iniciar do seu computador.

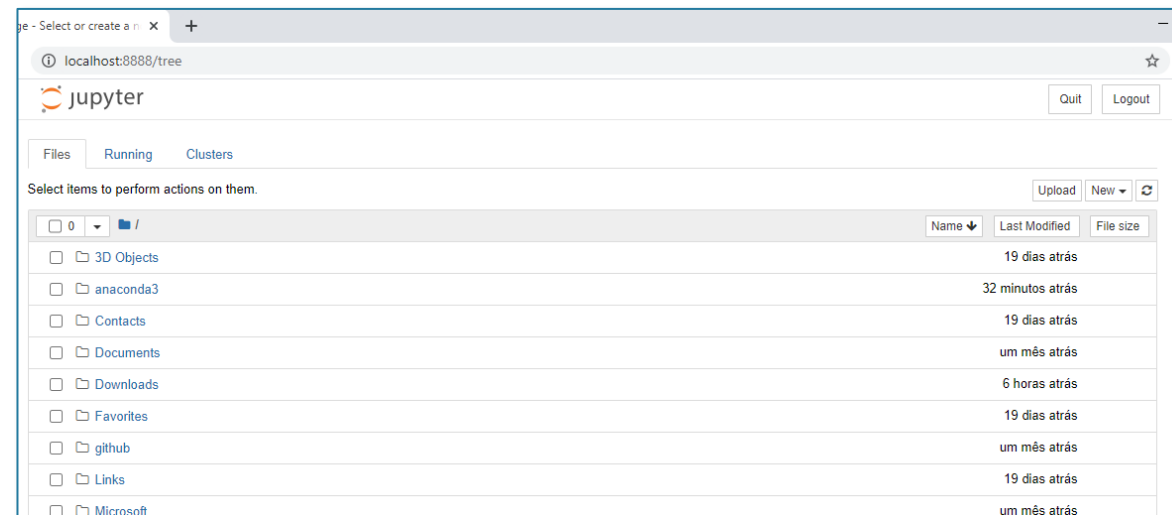
Uma nova janela será aberta no seu navegador padrão de internet.



ATENÇÃO !

Ao clicar no ícone do Jupyter seu navegador padrão deverá abrir o Jupyter Notebook como no print 2.

Além disso, uma janela preta com o símbolo do Jupyter irá abrir. **Não feche esta janela!** Ela é o Jupyter Notebook sendo rodado pelo seu computador.



Jupyter Notebook

Inicializando o Jupyter

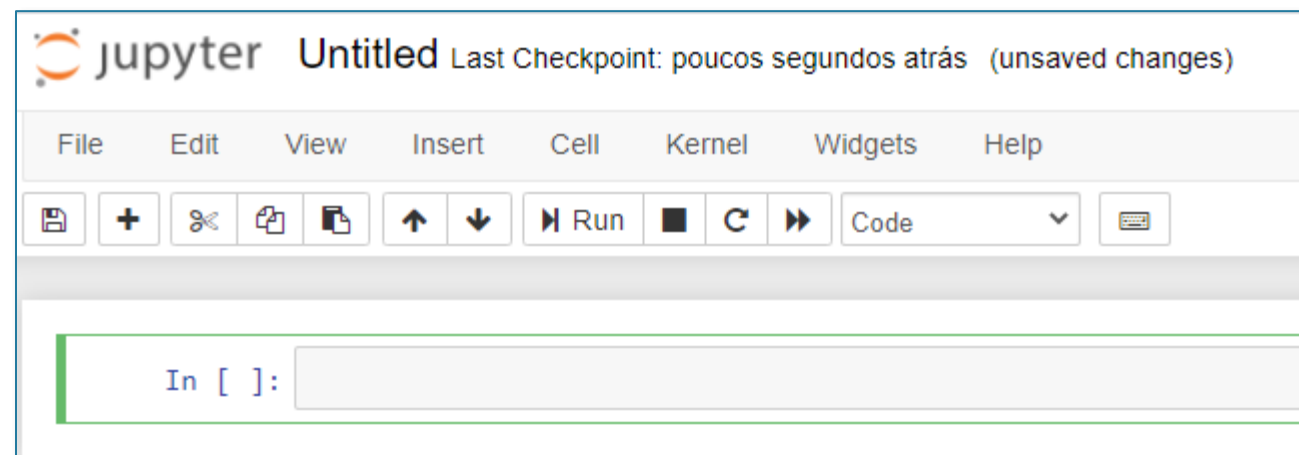
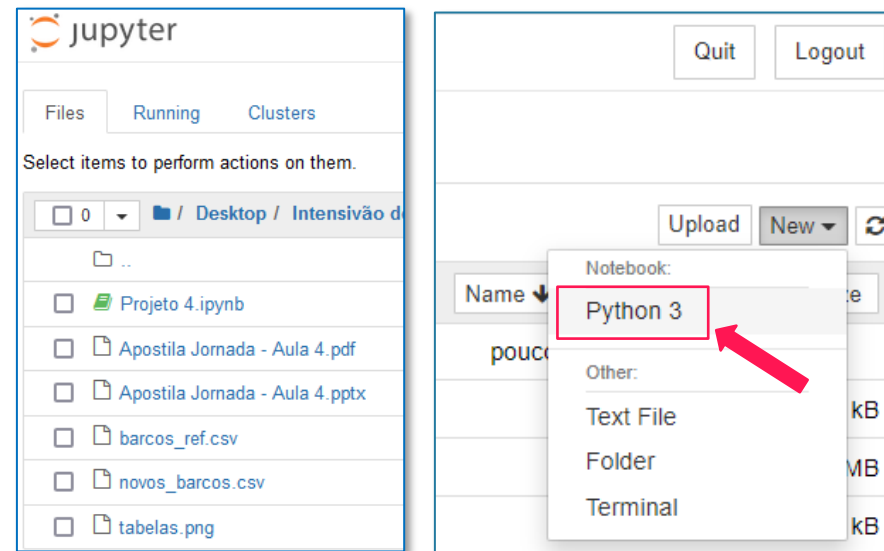
Após os preparativos, vamos acessar o nosso primeiro arquivo para criarmos nosso código em Python.

Esse arquivo se chama Notebook e possui o formato **.ipynb**. Ele só será aberto dentro de plataformas como o Google Colab ou Jupyter Notebook.

Para **criarmos um novo Notebook**, devemos clicar em **New > Python3**, assim como apresentado na figura ao lado.

Ao criar o novo Notebook, o Jupyter Notebook abrirá a janela ao lado. Aqui, é onde escreveremos nosso programa.

Mas antes de tudo, vamos entender a interface dessa plataforma.



Entendendo a interface

The image shows the Jupyter Notebook interface with several annotations in Portuguese:

- Nome do arquivo. Formato ipynb.**: Points to the **Untitled** tab title.
- Last Checkpoint: um minuto atrás (unsaved changes)**: Text next to the tab title.
- File Edit View Insert Cell Kernel Widgets Help**: The menu bar.
- +**: A button in the toolbar, annotated with **Cria nova célula** (Create new cell).
- Run**: A button in the toolbar, annotated with **Executa o código** (Execute the code).
- In []:**: The prompt for the code cell.
- Célula onde deve ser escrito o código**: Points to the code input area.
- Região de Output**: Points to the output area below the code cell.

Parte 5

Extração/Obtenção dos dados



INTENSIVÃO DE
PYTHON {#}
100% ONLINE & GRÁTUITO

Extração/Obtenção dos dados

Importando o Pandas

Como vimos no nosso diagrama, as partes 1 e 2 do nosso processo de Ciência de Dados é realizada FORA do Python. De forma bem simples, são as informações que introduzimos no Capítulo 1 dessa apostila.

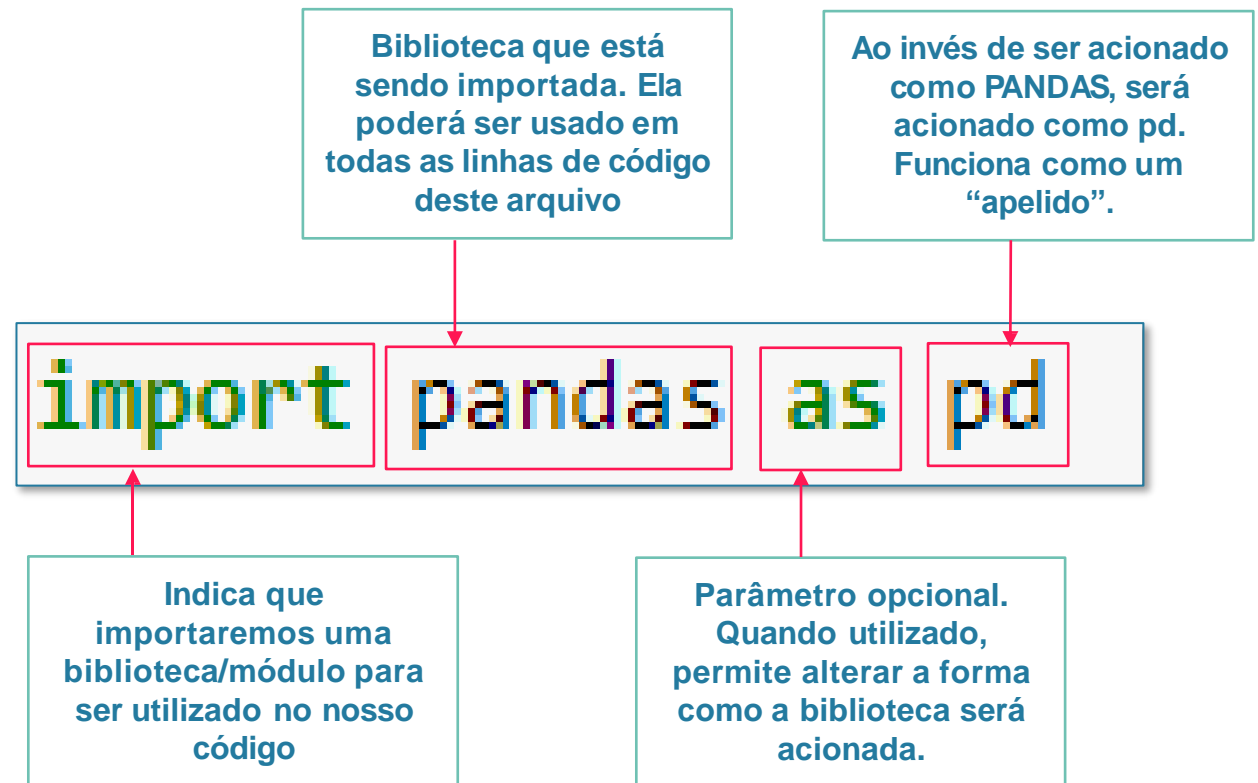
Agora vamos para a primeira etapa DENTRO do Python. A de obtenção de dados de fontes externas.

Apesar de agora ter um nome mais bonito, é algo que já vimos nas aulas 1 e 2 da Semana do Python. Usando o PANDAS podemos realizar essa importação dos dados do arquivo CSV e leva-los para o Jupyter Notebook.

Para realizar essa etapa, iremos usar a linha de código abaixo:

Import pandas as pd

Caso você ainda não conheça o Pandas, sugerimos dar uma olhadinha nas apostilas das aulas 1 e 2 😊.



Importando os dados de um arquivo .csv

Depois de importar o pandas já podemos importar e visualizar a nossa base de dados.

```
import pandas as pd

tabela = pd.read_csv("barcos_ref.csv")
display(tabela)
```

	Preco	TipoBarco	TipoVenda	Ano	Comprimento	Largura	Material
0	3490	1	1	2020	4.00	1.50	1
1	3399	2	1	2019	3.55	1.46	2
2	3600	3	2	1999	6.20	2.38	2
3	3500	2	2	1987	4.35	1.75	3
4	3500	4	2	2004	4.70	2.00	3
...
7644	5000	7	2	1974	7.20	2.70	3
7645	4999	4	2	1980	6.00	2.10	3
7646	4950	2	2	1984	6.00	1.62	5
7647	4499	4	1	2020	4.40	1.80	3
7648	3780	2	1	2019	3.60	1.60	3

7649 rows × 7 columns

É sempre bom lembrar que ao importar uma base de dados é necessário verificar como ela está, se temos colunas vazias, se temos linhas vazias.

Se de fato vamos utilizar todos os dados para fazer nossa análise. Isso é muito importante para que você não tenha retrabalho e acaba tendo resultados inesperados.

Então fazer esse procedimento logo no início é o ideal para evitar alguns erros futuros.

Como essa nossa base de dados já está certa não vamos precisar fazer os tratamentos, mas você pode repetir alguns dos passos que vimos nas aulas anteriores para treinar aqui também!

Parte 6

Análise Exploratória



INTENSIVÃO DE
PYTHON {#}
100% ONLINE & GRÁTUITO

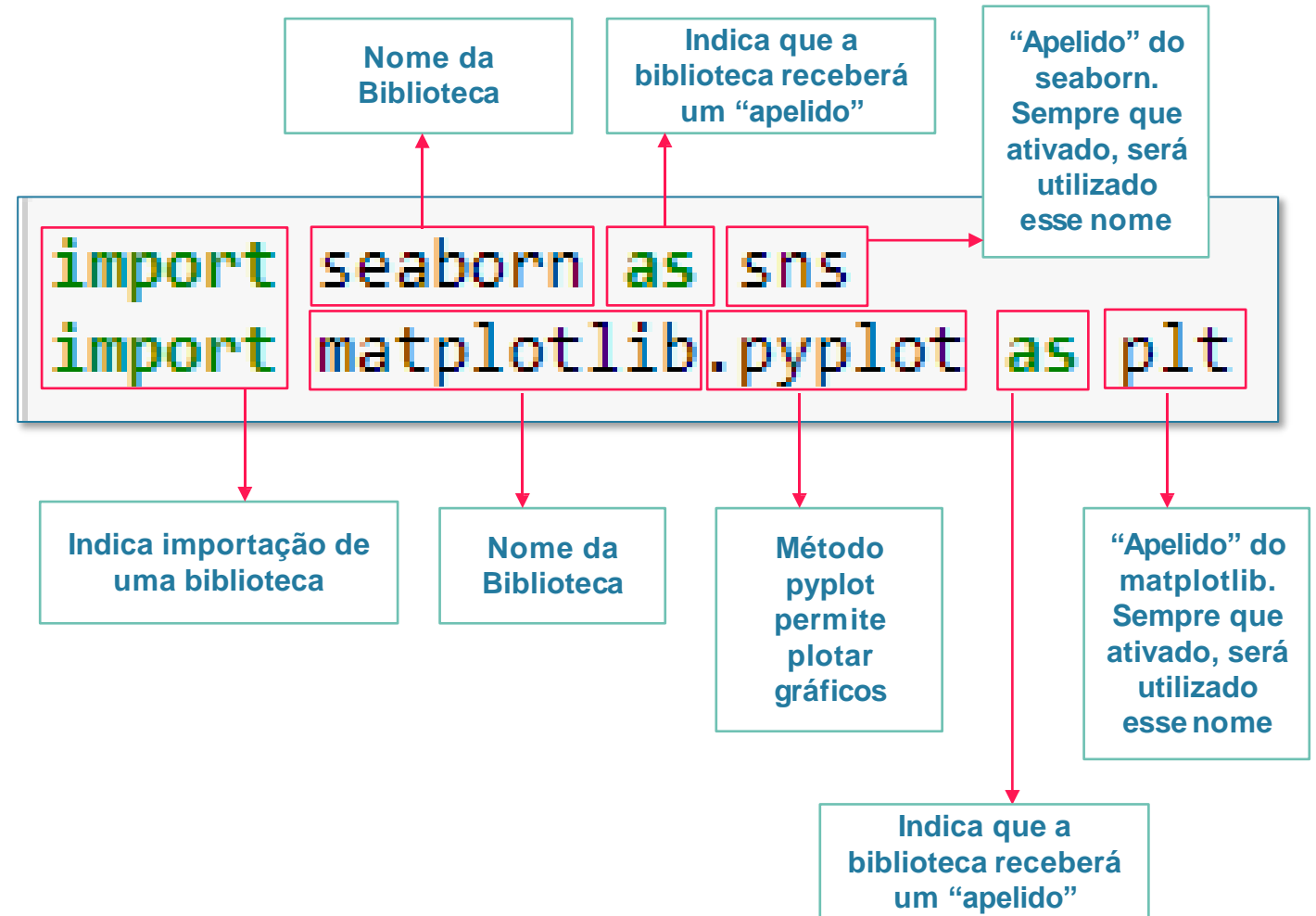
Importando as bibliotecas gráficas

Com o uso do Python e do Pandas, vamos poder gerar uma série de análises estatísticas/gráficas que nos permita entender melhor como esses dados se comportam.

Para isso, vamos importar 2 bibliotecas que nos auxiliam a criar gráficos:

- Seaborn ([link](#));
- Matplotlib ([link](#));

Assim como fizemos na aula 2 com o Plotly para utilizarmos essas bibliotecas, é necessário importa-las para nosso código. Para isso usaremos os comandos ao lado.

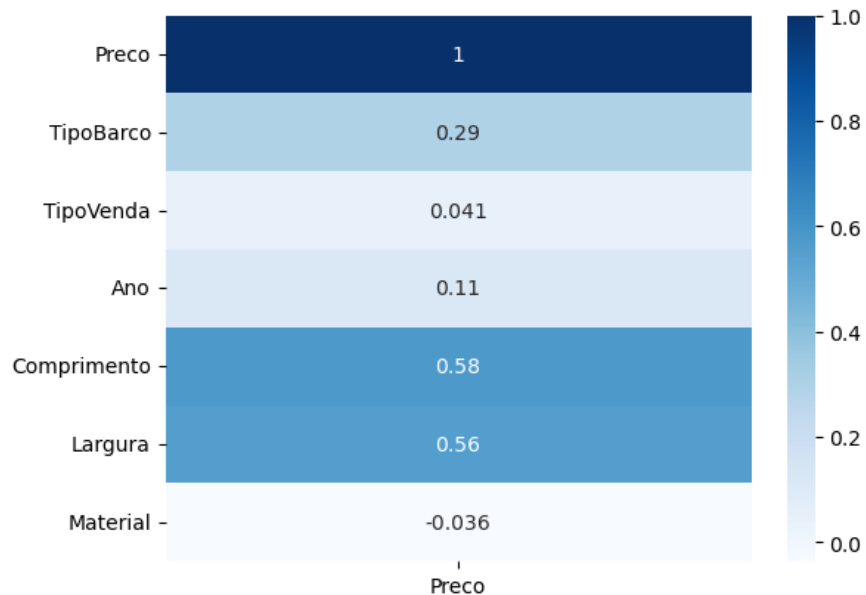


Análise Exploratória dos dados

Análise Gráfica

```
print(tabela.corr()["Preco"])  
  
# optional  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
sns.heatmap(tabela.corr()[["Preco"]], annot=True, cmap="Blues")  
plt.show()
```

```
Preco      1.000000  
TipoBarco  0.293137  
TipoVenda  0.041266  
Ano        0.113385  
Comprimento 0.578353  
Largura    0.557621  
Material   -0.036476  
Name: Preco, dtype: float64
```



A primeira análise que vamos fazer é da correlação, ou seja, vamos ver a relação entre os dados da nossa base com o preço, pois é isso que queremos prever.

A correlação varia entre -1 e 1, sendo que quanto mais próximo de 0 menos relação nós temos.

Agora quanto mais longe de 0, ou seja, mais próximo de 1 ou -1 significa que essa relação é maior, podendo ser tanto positiva quanto negativa.

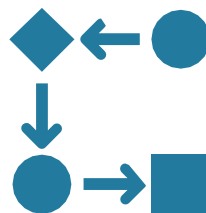
Então por esse gráfico inicial você nota a informação “mais relevante” para o preço do barco é o **comprimento** e a **largura**.

Agora se você observar o material ele possui uma correlação muito próxima de 0, assim como o tipo de venda.

Só com isso já podemos verificar que essas duas informações não serão tão relevantes no preço final.

Parte 7

Modelagem + Algoritmos



Criando base de dados para modelo (1/4)

Vamos agora para fase de criação de um modelo.

O que precisamos identificar são:

- Os inputs do modelo (eixo X);
- Os outputs do modelo (eixo Y);

Lembrando que nosso objetivo é criar um modelo de previsão de preços baseado nos dados que temos na nossa tabela.

Nossos inputs são as informações do barco, enquanto o nosso output será o preço do barco.

Vamos iniciar este bloco de código informando exatamente isso. Usaremos para essa etapa uma nova biblioteca. E como sempre, vamos precisar importa-las.

`sklearn.model_selection`

Esta é outra forma de importar uma biblioteca. Ao invés de importarmos toda a biblioteca, indicamos apenas a parte dela que temos interesse. Como se definíssemos a biblioteca e qual dos livros vamos pegar.

Ex: Da(from) biblioteca nacional(sklearn.model_selection) pegue(import) Dom Casmurro (train_test_split)

```
from sklearn.model_selection import train_test_split  
  
y = tabela["Preço"]  
x = tabela.drop("Preço", axis=1)
```

Receberá os valores que irão compor o eixo X (inputs)

Indica que todo o dataframe com exceção da coluna 'Preço' vai compor os valores de X (inputs)

Receberá os valores que irão compor o eixo Y (outputs)

Receberá os valores que irão compor o eixo Y (outputs)

```
y = tabela["Preço"]
```


Criando base de dados para modelo (2/4)

Agora temos claro nas nossas variáveis X e Y quais são os Inputs e quais são os Outputs.

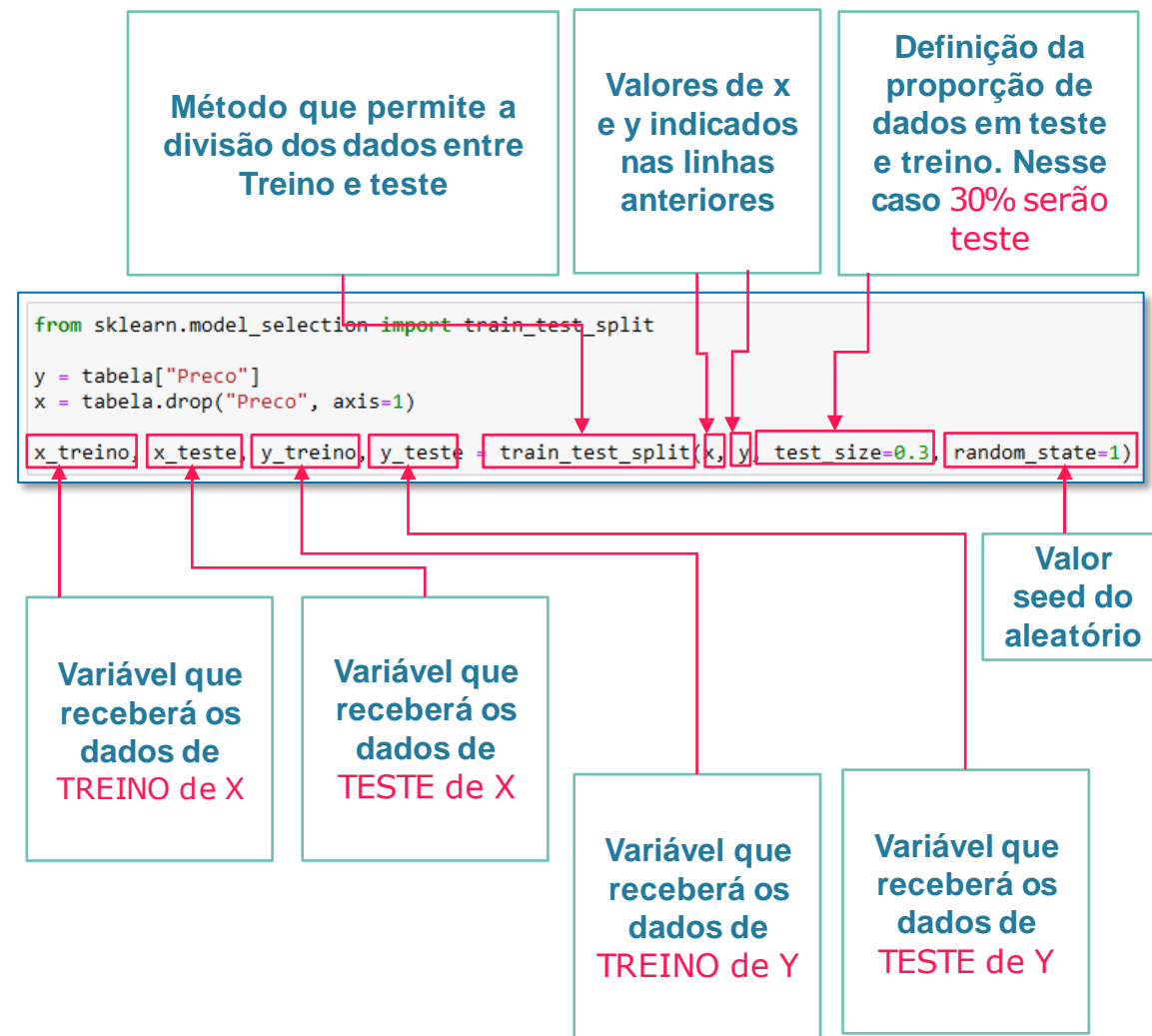
Para criarmos um modelo de previsão usando inteligência artificial vamos precisar de dados para:

- 1) Treinar o modelo;
- 2) Testar o modelo.

Fazemos isso, pois precisamos de dados para treinar o nosso modelo e depois para testá-lo, assim vamos garantir que ele funciona da forma correta.

Não adianta eu utilizar a mesma base para treinar o modelo e para testá-lo, pois ele pode estar simplesmente decorando as informações.

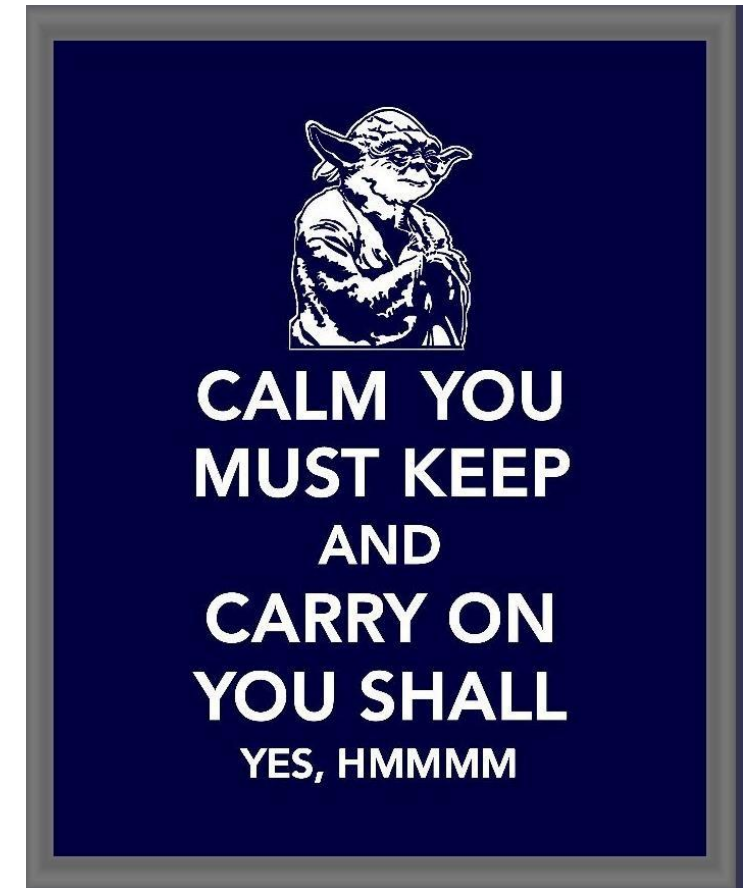
Por isso precisamos de bases diferentes de treino e teste. Você não precisa criar ou procurar uma nova base, basta dividir a sua base de dados para usar uma parte em treino e a outra em teste.



PAUSA PARA REVISAR

Sabemos que a última etapa pode não ser tão direta em um primeiro momento, mas vamos revisar o que fizemos até agora e pensar o que vamos fazer para frente.

- 1) Importamos a base de dados do arquivo .CSV para nosso programa;
- 2) Pensamos em dar uma limpeza nos dados, mas como tudo estava OK, passamos para a próxima etapa;
- 3) Analisamos graficamente nossa base usando a correlação para verificar quais informações são mais relevantes no preço.
- 4) Começamos a pensar no nosso modelo de previsão que se utilizará de inteligência artificial;
- 5) Para gerar uma inteligência artificial precisamos treiná-la e testá-la. Como fazemos isso? Através de dados!
- 6) Criamos 4 variáveis(**x_train**, **x_test**, **y_train**, **y_test**) com objetivo de fornecê-las a nossa inteligência artificial;
- 7) Cá estamos! Usando nossa biblioteca vamos informar essas variáveis e iniciar o treinamento do nosso modelo. Feito isso, testar esse modelo.



Criando base de dados para modelo (3/4)

Iniciamos por novas importações de dados. Aqui usaremos dois métodos distintos de regressão:

- `RegressãoLinear`;
- `RandomForestRegressor`;

Para isso temos duas partes do código, a primeira é criando as inteligências artificiais utilizando esses dois métodos.

Em seguida temos o treino dessas das inteligências. Aqui é importante notar que estamos utilizando apenas as bases de treino, pois estamos treinando a nossa inteligência para conseguir prever o preço do barco de acordo com as suas características.

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# cria as inteligencias aritificiais
modelo_regressaolinear = LinearRegression()
modelo_arvoredecisao = RandomForestRegressor()

# treina as inteligencias artificias
modelo_regressaolinear.fit(x_treino, y_treino)
modelo_arvoredecisao.fit(x_treino, y_treino)
```

Criando base de dados para modelo (4/4)

Depois de criar e treinar as inteligências artificiais nós vamos criar as previsões e comparar esses dois modelos.

Assim podemos verificar qual deles é melhor nesse caso específico.

Na parte superior temos os códigos para criar as previsões utilizando seus respectivos métodos.

Já na parte inferior nós estamos printando a métrica **R2**, que é o **coeficiente de determinação**.

Esse coeficiente é que vai definir se o nosso modelo é bom ou não. Lembrando que o maior valor é 1, que seria o valor excelente.

Com esses dois valores você já consegue visualizar que o método da **Árvore de Decisão** é bem melhor para esse caso dos barcos.

```
from sklearn import metrics

# criar as previsoes
previsao_regressaolinear = modelo_regressaolinear.predict(x_teste)
previsao_arvoredecisao = modelo_arvoredecisao.predict(x_teste)

# comparar os modelos
print(metrics.r2_score(y_teste, previsao_regressaolinear))
print(metrics.r2_score(y_teste, previsao_arvoredecisao))

0.4490324760735813
0.8665231703483982
```

IMPORTANTE: O método aqui escolhido nem sempre vai ser o melhor para todos os casos, aqui ele funcionou bem, mas pode ser que em outro caso outros métodos sejam melhores.

Por isso é que temos que fazer testes e comparações, para garantir que estamos utilizando o método correto para cada situação.

Parte 9

Interpretação do Resultado



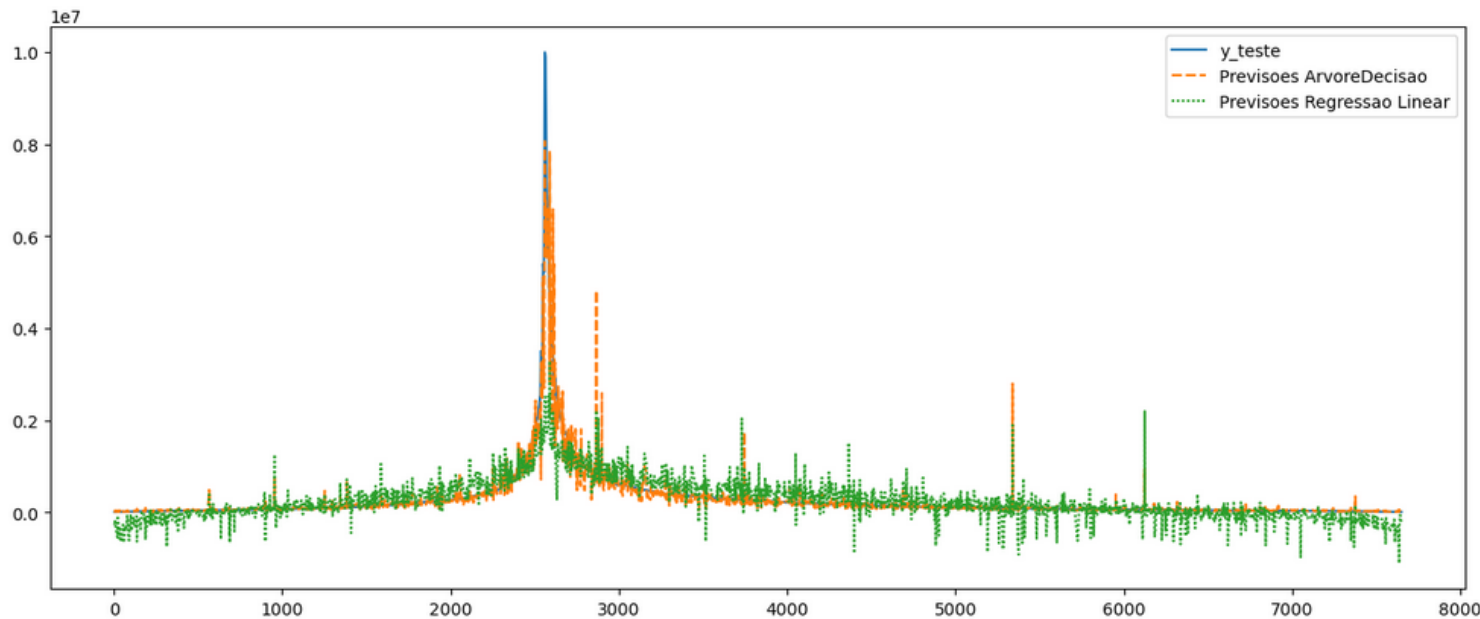
INTENSIVÃO DE
PYTHON {#}
100% ONLINE & GRATUITO

Interpretação dos resultados

Análise Gráfica

```
tabela_auxiliar = pd.DataFrame()
tabela_auxiliar["y_teste"] = y_teste
tabela_auxiliar["Previsoes ArvoreDecisao"] = previsao_arvoredecisao
tabela_auxiliar["Previsoes Regressao Linear"] = previsao_regressaoilinear

plt.figure(figsize=(15,6))
sns.lineplot(data=tabela_auxiliar)
plt.show()
```



Nós já sabemos que o método de Árvore de Decisão para esse caso seria a melhor escolha, mas é interessante visualizar como as informações se comportam.

No gráfico nós temos os resultados da previsão da **Árvore de Decisão**, da **Regressão Linear** e da nossa base de **Teste**.

Claro que aqui não vamos ter uma previsão 100% exata, até porque os modelos não são ideais.

Mas você consegue notar que os dois modelos seguem bem a nossa curva de teste.

De fato temos que o modelo de Regressão Linear acaba tendo muito mais variações do que o modelo de Árvore de Decisão.

Interpretação dos resultados

Previsão de Preços

Lembra que na fase exploratória percebemos que tanto o comprimento quanto a largura eram as características que mais afetavam os preços.

Utilizando o arquivo "**novos_barcos.csv**" vamos fazer a previsão de preço e verificar se de fato o nosso modelo segue esse mesmo padrão que vimos na análise exploratória.

Como o comprimento e a largura são as características mais "relevantes" para o preço do barco, claramente conseguimos notar que quanto maior o barco maior o seu preço.

E pela nossa previsão é exatamente isso que temos, o último barco, mesmo sendo do ano de 1999 acaba sendo o maior e por isso possui o maior preço entre os 3.

```
nova_tabela = pd.read_csv("novos_barcos.csv")
display(nova_tabela)
previsao = modelo_arvoredecisao.predict(nova_tabela)
print(previsao)
```

	TipoBarco	TipoVenda	Ano	Comprimento	Largura	Material
0	2	3	2023	3.0	1.5	1
1	2	1	2023	3.8	1.9	2
2	3	2	1999	6.2	3.0	2


```
[ 5979.2655  8816.2875 16208.65 ]
```

Parte 10

Deploy/Produção



INTENSIVÃO DE {#}
PYTHON{#}
100% ONLINE & GRATUITO

Deploy/Produção

E o Deploy? Ou melhor, o que é o Deploy?

Depois de criar um modelo e testar sua eficácia, chega a hora de colocar aquele modelo para rodar. Isso é o Deploy, ou colocar em modo de Produção.

Aqui no Intensivão de Python, não vamos ver no detalhe ☹️

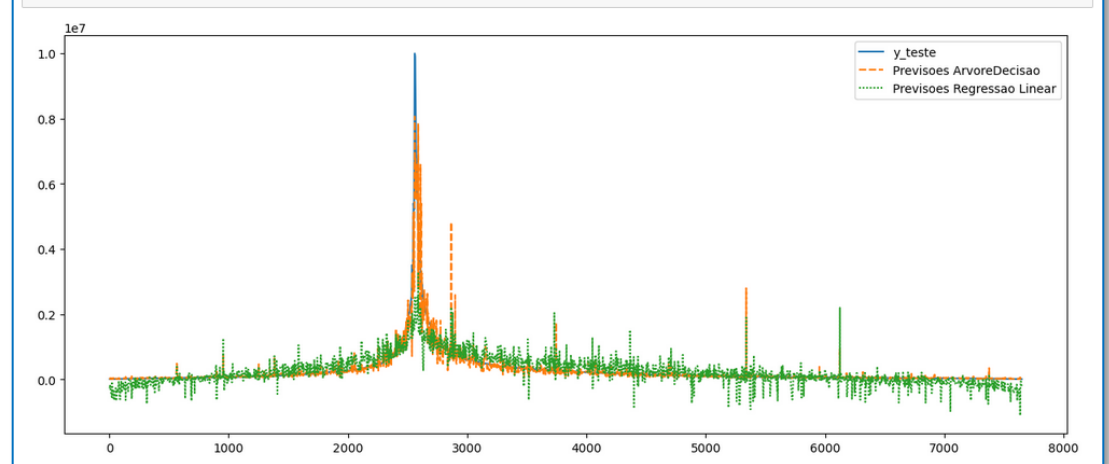
Mas em linhas gerais, o que precisamos fazer agora é criar uma espécie de sistema que possa ser executado por qualquer um que precise.

Assim você torna seu modelo escalável dentro da sua empresa.

Se tiver interesse de saber mais sobre Ciência de Dados e outros temas envolvendo Python fica de olho no nosso YouTube OU aproveita e já se inscreve no Python Impressionador 😊

```
tabela_auxiliar = pd.DataFrame()
tabela_auxiliar["y_teste"] = y_teste
tabela_auxiliar["Previsoes ArvoreDecisao"] = previsao_arvoredecisao
tabela_auxiliar["Previsoes Regressao Linear"] = previsao_regressao_linear

plt.figure(figsize=(15,6))
sns.lineplot(data=tabela_auxiliar)
plt.show()
```



```
nova_tabela = pd.read_csv("novos_barcos.csv")
display(nova_tabela)
previsao = modelo_arvoredecisao.predict(nova_tabela)
print(previsao)
```

	TipoBarco	TipoVenda	Ano	Comprimento	Largura	Material
0	2	3	2023	3.0	1.5	1
1	2	1	2023	3.8	1.9	2
2	3	2	1999	6.2	3.0	2

[5979.2655 8816.2875 16208.65]

INTENSIVÃO DE PYTHON {#}

100% ONLINE & GRATUITO

Ainda não segue a gente no Instagram e nem é inscrito no nosso canal do Youtube? Então corre lá!



@hashtagprogramacao



youtube.com/hashtag-programacao

