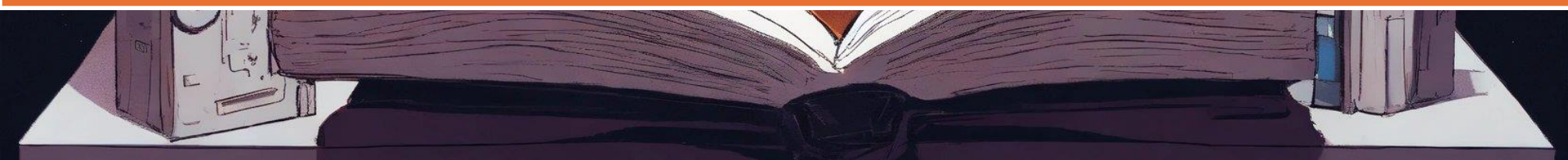


RAFAEL NEVES DE OLIVEIRA



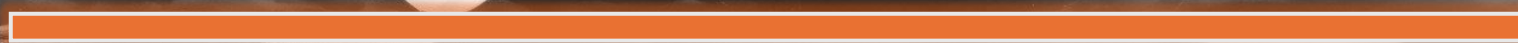
Python para iniciantes

CIÊNCIA DE DADOS SEM COMPLICAÇÃO



01

COMECE POR
AQUI!



Conheça Python!



Você já se perguntou como transformar uma pilha de dados em insights valiosos? Se sim, este e-book é para você! Vamos começar nossa jornada aprendendo Python, uma das linguagens mais populares e poderosas para a Ciência de Dados.

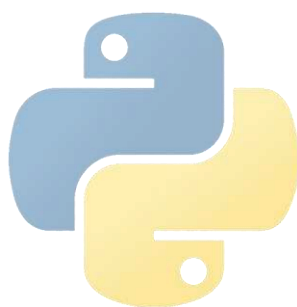
O que é Python?

Python é uma linguagem de programação de alto nível, conhecida por sua simplicidade e legibilidade. É amplamente utilizada em diversas áreas, especialmente em Ciência de Dados, devido às suas bibliotecas poderosas e sua comunidade ativa.

Instalando o Python

Antes de começarmos a programar, precisamos instalar o Python. Aqui estão os passos:

- 1. Baixar o Python:** Visite python.org e baixe a versão mais recente.
- 2. Instalar:** Siga as instruções de instalação, certificando-se de marcar a opção "Add Python to PATH".

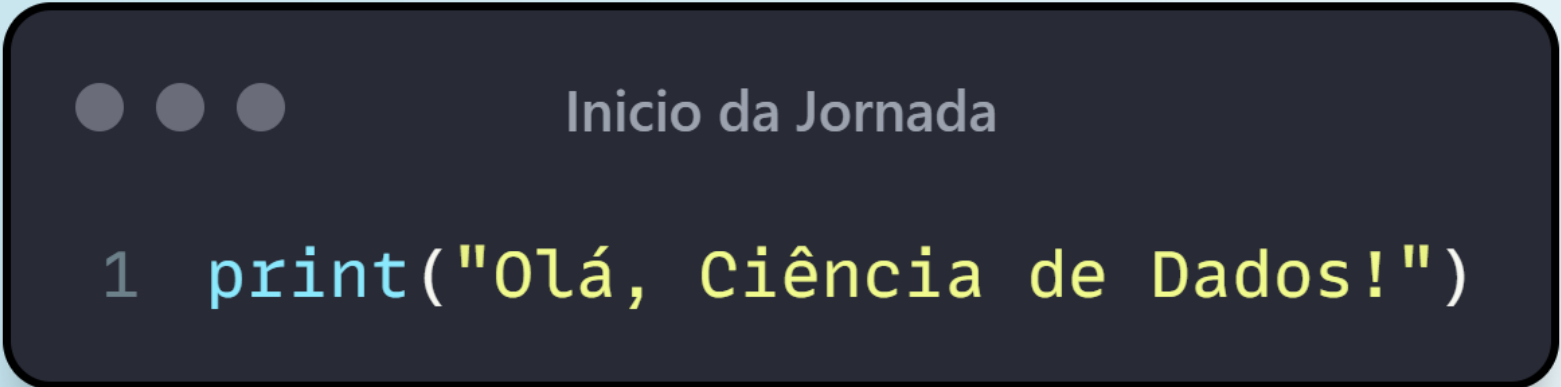


Mão na massa



Seu Primeiro Programa

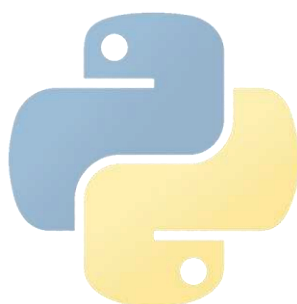
Abra o editor de texto do Python (IDLE) ou um ambiente de desenvolvimento integrado (IDE) como Jupyter Notebook, e digite o seguinte código:

A dark-themed code editor window with a title bar containing three window control buttons and the text "Inicio da Jornada". The editor contains a single line of Python code:

```
1 print("Olá, Ciência de Dados!")
```

```
1 print("Olá, Ciência de Dados!")
```

Salve e execute o arquivo. Parabéns, você escreveu seu primeiro programa em Python!



Mão na massa



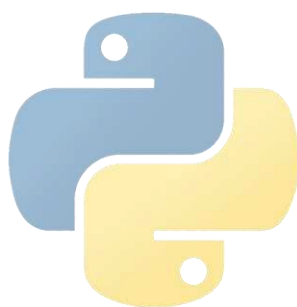
Variáveis e Tipos de Dados

Vamos explorar variáveis e tipos de dados básicos:.

```
● ● ● Início da Jornada
1 nome = "Ana"
2 idade = 30
3 altura = 1.75
4 dados = True
5
6 print(f"Nome: {nome}, Idade: {idade}, Altura: {altura}, Dados: {dados}")
7
```

Exercício Prático

- Crie variáveis para armazenar seu nome, idade e altura.
- Use o print para exibir uma frase apresentando-se.



02

Manipulação de Dados com Pandas

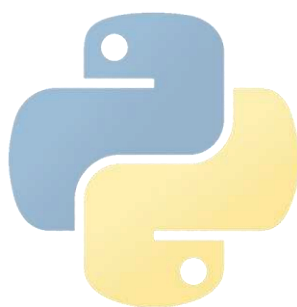
Introdução ao Pandas



Pandas é uma biblioteca poderosa para manipulação e análise de dados em Python. Vamos começar importando a biblioteca e criando um Data Frame:

```
● ● ● Início da Jornada

1 import pandas as pd
2
3 dados = {
4     "nome": ["Ana", "Bruno", "Carlos"],
5     "idade": [23, 35, 45],
6     "nota": [8.5, 7.0, 9.0]
7 }
8
9 df = pd.DataFrame(dados)
10 print(df)
11
12
```



Leitura de Arquivos

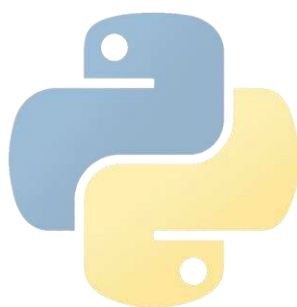


Vamos aprender a ler arquivos CSV, uma forma comum de armazenar dados em excel:

Inicio da Jornada

```
1 df = pd.read_csv('seu_arquivo.csv')
2 print(df.head())
```

**Note que usamos o (df.head())
para que seja possível visualizar os primeiros elementos
do Data Frame**



Limpeza de Dados

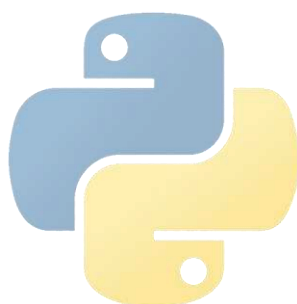


A limpeza de dados é uma parte essencial da Ciência de Dados. Vamos ver como identificar e tratar valores ausentes:

Início da Jornada

```
1 # Verificar valores nulos
2
3 df.isnull().sum()
4
5 # Preencher valores nulos com a média
6
7 df['coluna'].fillna(df['coluna'].mean(), inplace=True)
```

“#” significa um comentário. Isso não afeta o código!



Filtragem e Seleção de Dados

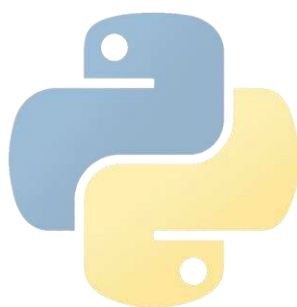
Vamos filtrar e selecionar dados específicos:

Início da Jornada

```
1 df_filtrado = df[df['idade'] > 30]
2 print(df_filtrado)
3
```

Exercício Prático

- Crie um Data Frame com dados fictícios sobre seus amigos (nome, idade, cidade).
- Leia um arquivo CSV usando pandas.
- Limpe os dados preenchendo valores ausentes.
- Filtre o DataFrame para exibir apenas amigos com mais de 25 anos.



The background is a cosmic-themed image. A large black triangle is centered, with a bright, glowing sphere (resembling a planet or moon) positioned in front of it. The sphere is partially obscured by the triangle's edges. The background is filled with stars and nebulae. A grid of white lines, resembling a constellation map or a data visualization, is overlaid on the lower half of the image. A horizontal orange line is positioned below the main title.

03

Visualização com Matplotlib e Seaborn

Introdução à Visualização de Dados

Visualizações ajudam a entender melhor os dados.

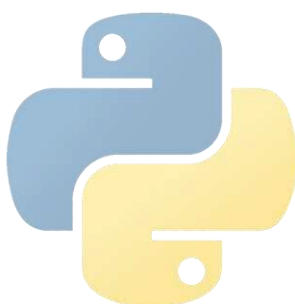
Vamos usar Matplotlib e Seaborn, duas bibliotecas populares para visualização.

Criando Gráficos Simples com Matplotlib

Vamos criar um gráfico de barras:

```
Inicio da Jornada

1 import matplotlib.pyplot as plt
2
3 dados = {
4     "nomes": ["Ana", "Bruno", "Carlos"],
5     "notas": [8.5, 7.0, 9.0]
6 }
7
8 plt.bar(dados["nomes"], dados["notas"])
9 plt.xlabel("Nomes")
10 plt.ylabel("Notas")
11 plt.title("Notas dos Alunos")
12 plt.show()
```



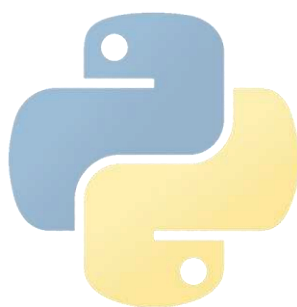
Visualizações Avançadas com Seaborn

Seaborn facilita a criação de gráficos estatísticos.

Vamos criar um gráfico de dispersão:

```
Inicio da Jornada

1 import seaborn as sns
2
3 sns.scatterplot(x='idade', y='nota', data=df)
4 plt.xlabel("Idade")
5 plt.ylabel("Nota")
6 plt.title("Idade vs Nota")
7 plt.show()
8
```



Explorando Dados com Boxplot

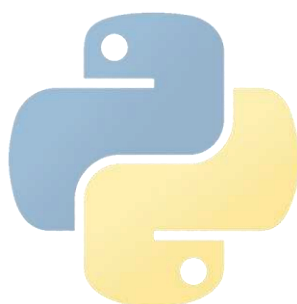
Vamos utilizar o boxplot para explorar a distribuição de uma variável e compará-la entre diferentes categorias:

Início da Jornada

```
1 sns.boxplot(x='idade', y='nota', data=df)
2 plt.xlabel("Idade")
3 plt.ylabel("Nota")
4 plt.title("Distribuição das Notas por Idade")
5 plt.show()
```

Exercício Prático

- Crie um gráfico de barras para exibir as notas dos alunos.
- Crie um gráfico de dispersão para mostrar a relação entre idade e nota.
- Use um boxplot para explorar a distribuição das notas por idade.



04

Estatísticas Descritivas e Análise Exploratória

Estatísticas Descritivas



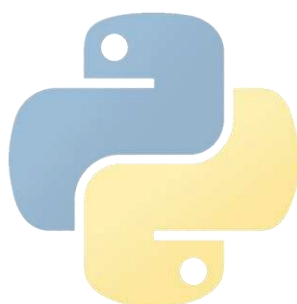
Vamos usar Pandas para calcular estatísticas descritivas básicas:

```
● ● ● Início da Jornada  
1 print(df.describe())
```

Análise Exploratória de Dados (EDA)

EDA é o processo de resumir as principais características dos dados. Vamos explorar um conjunto de dados real:

```
● ● ● Início da Jornada  
  
1 import seaborn as sns  
2  
3 # Carregar o dataset Iris  
4 df = sns.load_dataset('iris')  
5 print(df.head())  
6  
7 # Estatísticas descritivas  
8 print(df.describe())
```



Visualização de Distribuições

Vamos visualizar a distribuição de uma variável:

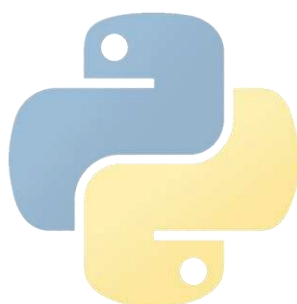
Início da Jornada

```
1 sns.histplot(df['sepal_length'], kde=True)
2 plt.xlabel("Comprimento da Sépala")
3 plt.title("Distribuição do Comprimento da Sépala")
4 plt.show()
```

Boxplot para Comparar Variáveis

Início da Jornada

```
1 sns.boxplot(x='species', y='sepal_length', data=df)
2 plt.xlabel("Espécie")
3 plt.ylabel("Comprimento da Sépala")
4 plt.title("Comprimento da Sépala por Espécie")
5 plt.show()
```

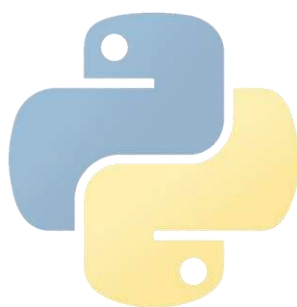


Visualização de Distribuições



Exercício Prático

- Carregue um conjunto de dados real, como o Iris Dataset.
- Calcule estatísticas descritivas básicas.
- Visualize a distribuição de uma variável de interesse.
- Crie um boxplot para comparar uma variável entre diferentes grupos.





05

Modelagem Preditiva com Scikit-Learn

Introdução ao Scikit-Learn

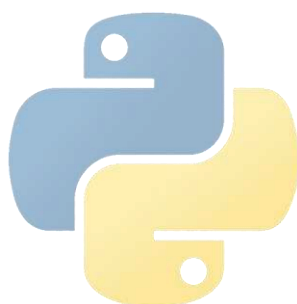
Scikit-Learn é uma biblioteca poderosa para aprendizado de máquina em Python.

Vamos usá-la para construir um modelo preditivo.

Preparação dos Dados

Primeiro, precisamos preparar os dados:

```
1 from sklearn.model_selection import train_test_split
2
3 X = df.drop('species', axis=1)
4 y = df['species']
5
6 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



Introdução ao Scikit-Learn

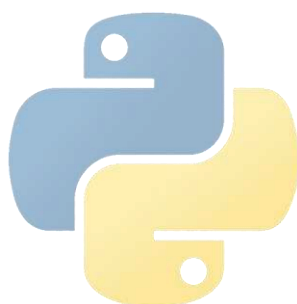


Construindo um Modelo de Regressão Logística

Vamos construir e treinar um modelo de Regressão Logística:

Início da Jornada

```
1 from sklearn.linear_model import LogisticRegression
2
3 modelo = LogisticRegression()
4 modelo.fit(X_train, y_train)
```



Introdução ao Scikit-Learn



Avaliando o Modelo

Vamos avaliar a performance do modelo:

```
● ● ● Início da Jornada

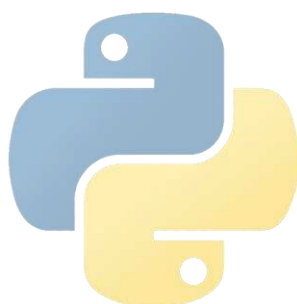
1 from sklearn.metrics import accuracy_score
2
3 predicoes = modelo.predict(X_test)
4 acuracia = accuracy_score(y_test, predicoes)
5
6 print(f"Acurácia do modelo: {acuracia * 100:.2f}%")
```

Melhorando o Modelo

Vamos explorar técnicas para melhorar o modelo, como ajuste de hiperparâmetros e validação cruzada:

```
● ● ● Início da Jornada

1 from sklearn.model_selection import GridSearchCV
2
3 parametros = {'C': [0.1, 1, 10, 100]}
4 grid = GridSearchCV(LogisticRegression(), parametros, cv=5)
5 grid.fit(X_train, y_train)
6
7 print(f"Melhor parâmetro: {grid.best_params_}")
```



Introdução ao Scikit-Learn



Exercício Prático

- Carregue o Iris Dataset ou outro conjunto de dados disponível.
- Prepare os dados para treinamento e teste.
- Construa e treine um modelo de Regressão Logística.

Avalie a performance do modelo.

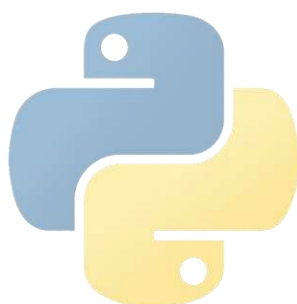
Explore técnicas para melhorar o modelo, como ajuste de hiperparâmetros:

Parabéns!

Você completou sua introdução ao Python para Ciência de Dados.

Continue praticando e explorando, e logo você será um especialista em analisar dados e extrair insights valiosos.

Boa sorte em sua jornada!



Agradecimientos

Continue Praticando!

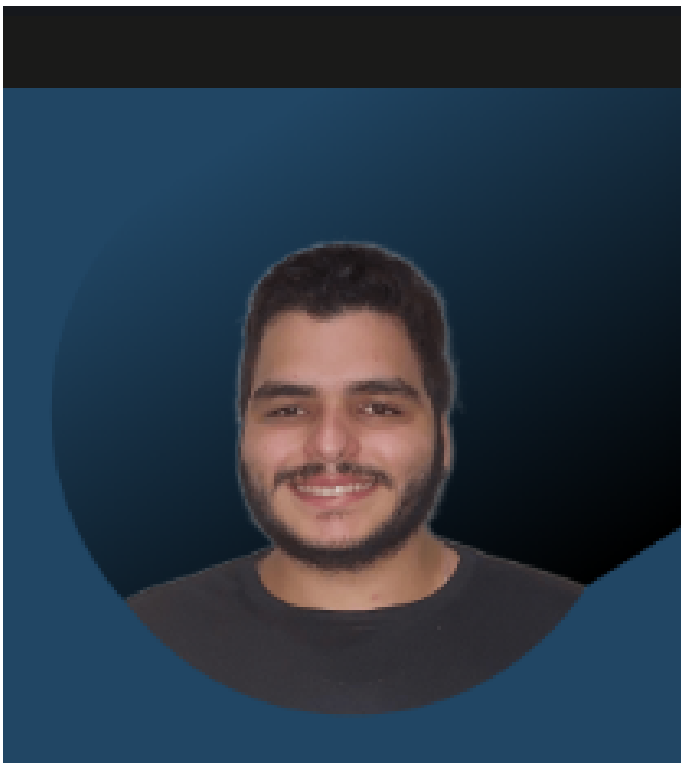


Esse E-Book foi gerado por IA e diagramado por humano.

O passo a passo se encontra no Github.

Esse conteúdo foi gerado com fins didáticos, de forma generalista e introdutória, pode apresentar erros pela IA.

- LinkedIn: <https://www.linkedin.com/in/rafael-neves-de-oliveira/>
- Github: <https://github.com/foreverpro700>



SOBRE MIM

Sou um estudante de TI com conhecimento tanto em tecnologia quanto no mercado financeiro brasileiro e internacional. Tenho experiência em metodologias ágeis como Scrum e Kanban, e minhas habilidades técnicas incluem proficiência em Python, Power BI, HTML e CSS. Estou comprometido em impulsionar a inovação e a eficiência em projetos de TI.

Possuo forte capacidade de adaptação rápida, colaboração eficaz e um desejo de aprendizado contínuo. Estou sempre buscando novas maneiras de melhorar minhas habilidades e contribuir de forma significativa para equipes de alto desempenho. Sou um comunicador eficaz e estou preparado para enfrentar desafios e superar expectativas em um ambiente dinâmico de TI.

