

Apostila Prática

Fundamentos de Python para Ciência de Dados e Machine Learning

Objetivo da Apostila

Capacitar o aluno a **programar em Python, analisar dados e aplicar modelos básicos de Machine Learning**, conectando teoria e prática com foco em resolução de problemas reais.

Público-alvo

- Iniciantes em programação
 - Estudantes de tecnologia, dados e engenharia
 - Pessoas migrando para Ciência de Dados e ML
-

PARTE 1 — INTRODUÇÃO À CIÊNCIA DE DADOS

1. A Era dos Dados

Vivemos em um mundo orientado por dados. A **Ciência de Dados** transforma grandes volumes de informação em **decisões estratégicas**.

- ◆ Ciência de Dados = Programação + Estatística + Conhecimento de Negócio
 - ◆ Papel do Cientista de Dados: - Coletar dados - Analisar padrões - Gerar insights acionáveis
-

2. Por que Python?

Python é o padrão da Ciência de Dados moderna.

-  Sintaxe simples e legível
-  Comunidade ativa
-  Bibliotecas poderosas

 Principais bibliotecas: - **NumPy** → Computação numérica - **Pandas** → Análise de dados - **Matplotlib / Seaborn** → Visualização - **Scikit-learn** → Machine Learning

3. Preparando o Ambiente

 Ferramenta recomendada: **Anaconda**

Inclui: - Python - Bibliotecas essenciais - Ambientes virtuais

■ Ambiente de execução: - **IDLE** ou **Jupyter Notebook**

PARTE 2 — FUNDAMENTOS DE PYTHON

4. Variáveis e Tipos de Dados

⌚ Variável = espaço de memória com nome

Tipos básicos: - `int` → números inteiros - `float` → números decimais - `str` → textos

```
x = 10  
y = 3.5  
nome = "Python"
```

5. Operadores

Matemáticos

```
+ - * /
```

🔍 Comparação

```
< > <= >= == !=
```

Resultado sempre booleano: `True` ou `False`

6. Strings (Textos)

💻 Operações essenciais: - Indexação (`texto[0]`) - Comprimento (`len()`) - Concatenação (`+`) - Caracteres de escape (`\`)



PARTE 3 — ESTRUTURAS DE DADOS

7. Listas

⌚ Coleções ordenadas e mutáveis

```
numeros = [1, 2, 3]
numeros.append(4)
```

- ✓ Permite duplicatas
 - ✓ Acesso por índice
-

8. Dicionários

- Estrutura chave → valor

```
aluno = {'nome': 'Ana', 'idade': 20}
```

- ✓ Busca rápida
 - ✓ Chaves únicas
-

9. Conjuntos (Sets)

- ☒ Não permitem duplicatas

```
s = {1, 2, 3}
```

- ✓ Muito eficientes para testes de pertencimento
-



PARTE 4 — CONTROLE DE FLUXO

10. Condicionais

```
if idade >= 18:
    print("Maior de idade")
else:
    print("Menor de idade")
```

11. Laços de Repetição

- ⌚ while

Executa enquanto a condição for verdadeira



Itera sobre sequências

```
for letra in "Python":  
    print(letra)
```

12. Funções

♻️ Reutilização de código

```
def soma(a, b):  
    return a + b
```



PARTE 5 — PROJETO PRÁTICO: CRIPTOGRAFIA

13. Cifra de César

🚫 Desloca caracteres usando uma chave

✓ Trabalha com: - Strings - Laços - Condicionais

14. Força Bruta

Testa todas as chaves possíveis

✓ Introdução a: - Busca exaustiva - Pensamento algorítmico



PARTE 6 — ANÁLISE DE DADOS COM PANDAS

15. DataFrame

(DbContext) Estrutura tabular avançada

```
import pandas as pd  
df = pd.read_csv('dados.csv')
```

16. Exploração Inicial (EDA)

Ferramentas essenciais: - `.head()` - `.info()` - `.describe()`

17. Seleção e Filtros

```
df[df['preco'] > 100]
```



PARTE 7 — VISUALIZAÇÃO DE DADOS

18. Gráficos Essenciais

- Histograma
- Linha
- Boxplot
- Dispersão

Ferramentas: - Matplotlib - Seaborn



PARTE 8 — MACHINE LEARNING

19. Conceitos Fundamentais

Aprender padrões a partir de dados

Tipos: - Supervisionado - Não supervisionado

Tarefas: - Classificação - Regressão

20. Pré-processamento

6 Dados limpos = modelos melhores

✓ Imputação ✓ Normalização ✓ One-Hot Encoding

21. Modelos de Classificação

-  KNN
 -  Árvores de Decisão
 -  Random Forest
-

22. Avaliação de Modelos

-  Métricas: - Acurácia - Precisão - Recall - AUC-ROC
-

23. Modelos de Regressão

-  Regressão Linear
 -  Random Forest Regressor
-

Avaliação: - MSE - R² - Gráfico de resíduos

CONCLUSÃO

Você construiu uma **base sólida em Python e Ciência de Dados**.

 Próximos passos: - Deep Learning - NLP - APIs e Web Scraping - Algoritmos e Big-O

 **A melhor forma de aprender é praticar.** Escolha um dataset, explore, erre, ajuste e evolua.

Boa jornada no mundo dos dados!