O DESPERTAR DO ENGENHEIRO DE DADOS

Jornada do Zero a Mestre com Python

DANIEL FARNEY

Vol.01

Introdução

Bem-vindo ao mundo fascinante da Engenharia de Dados! Se você é um entusiasta de tecnologia, programador iniciante ou um desenvolvedor experiente que deseja expandir suas habilidades, este Ebook é para você. A Engenharia de Dados é uma das áreas mais dinâmicas e essenciais no campo da tecnologia da informação, responsável por coletar, armazenar e processar dados de maneira eficiente para transformá-los em informações valiosas.

Neste Ebook, vamos guiá-lo desde os fundamentos básicos de Python, a linguagem de programação preferida para manipulação de dados, até técnicas avançadas usadas no dia a dia de um engenheiro de dados. Dividimos o conteúdo em três capítulos principais para facilitar seu aprendizado:

Fundamentos de Python

Python é uma linguagem de programação versátil e amplamente usada na engenharia de dados. Sua sintaxe simples e vasta biblioteca de módulos tornam-na ideal para manipulação de dados, automação de tarefas e construção de pipelines de dados.

Fundamentos de Python para Engenharia de Dados

1.1: O que é Python?

Python é uma linguagem de programação versátil e amplamente usada na engenharia de dados. Sua sintaxe simples e vasta biblioteca de módulos tornam-na ideal para manipulação de dados, automação de tarefas e construção de pipelines de dados.

1.2: Variáveis e Tipos de Dados

Python é uma linguagem de programação versátil e amplamente usada na engenharia de dados. Sua sintaxe simples e vasta biblioteca de módulos tornam-na ideal para manipulação de dados, automação de tarefas e construção de pipelines de dados.

```python
# Exemplo de uso de variáveis
nome = "Ana"
idade = 25
altura = 1.68
print(f"Nome: {nome}, Idade: {idade}, Altura: {altura}")

```

Os tipos de dados básicos em Python incluem inteiros, flutuantes, strings e booleanos.

```python
# Exemplos de diferentes tipos de dados
idade = 30  # inteiro
altura = 1.75  # flutuante
nome = "João"  # string
ativo = True  # booleano

print(type(idade))
print(type(altura))
print(type(nome))
print(type(ativo))
```

1.3: Estruturas de Controle

As estruturas de controle, como if-else e loops, permitem que você tome decisões e repita ações com base em condições específicas.

```python
# Exemplo de if-else com múltiplas condições
idade = 20
if idade < 18:
    print("Você é menor de idade")
elif idade < 60:
    print("Você é adulto")
else:
    print("Você é idoso")
```

```python
# Exemplo de loop for iterando sobre uma lista
nomes = ["Ana", "Pedro", "João"]
for nome in nomes:
    print(nome)

# Exemplo de loop while
contador = 0
while contador < 3:
    print(nomes[contador])
    contador += 1
```

Sessão 1.4: Funções

Funções são blocos de código reutilizáveis que realizam tarefas específicas. Elas ajudam a organizar e modularizar seu código.

```python
# Exemplo de função com parâmetros e retorno
def calcular_media(numeros):
    total = sum(numeros)
    quantidade = len(numeros)
    media = total / quantidade
    return media

notas = [7.5, 8.0, 9.5]
media = calcular_media(notas)
print(f"Média: {media}")
```

Sessão 1.5: Estruturas de Dados

Listas armazenam coleções ordenadas de itens e dicionários armazenam pares de chave-valor.

```python
# Exemplo de lista e operações com listas
numeros = [1, 2, 3, 4, 5]
numeros.append(6)  # adiciona um elemento à lista
print(numeros)

numeros.remove(3)  # remove o elemento 3 da lista
print(numeros)

print(f"O primeiro elemento da lista é: {numeros[0]}")
print(f"O último elemento da lista é: {numeros[-1]}")
```

```python
# Exemplo de dicionário e operações com dicionários
pessoa = {"nome": "Ana", "idade": 28, "cidade": "São Paulo"}
print(pessoa)

pessoa["idade"] = 29  # atualiza o valor associado à chave "idade"
print(pessoa)

print(f"O nome da pessoa é: {pessoa['nome']}")
print(f"A idade da pessoa é: {pessoa['idade']}")
```

02

Manipulação de Dados com Python

Exploraremos técnicas de leitura e escrita de arquivos, introdução ao uso da biblioteca Pandas para manipulação de grandes volumes de dados, e métodos de tratamento de dados como filtragem e agregação.

2.1 Leitura e Escrita de Arquivos

A leitura de arquivos permite importar dados de diferentes fontes, enquanto a escrita de arquivos permite exportar dados..

```python
# Exemplo de leitura de arquivo linha por linha
with open('dados.txt', 'r') as arquivo:
    for linha in arquivo:
        print(linha.strip())  # strip() remove espaços em branco extras
```

```python
# Exemplo de escrita de múltiplas linhas em um arquivo
linhas = ["Primeira linha", "Segunda linha", "Terceira linha"]
with open('saida.txt', 'w') as arquivo:
    for linha in linhas:
        arquivo.write(linha + '\n')
```

2.2: Introdução ao Pandas

Pandas é uma biblioteca essencial para manipulação de dados em Python. DataFrames são estruturas de dados bidimensionais que facilitam a manipulação de dados.

```python
import pandas as pd

# Exemplo de criação de DataFrame a partir de um dicionário
dados = {'Nome': ['João', 'Maria', 'Pedro'],
         'Idade': [28, 22, 35],
         'Cidade': ['São Paulo', 'Rio de Janeiro', 'Belo Horizonte']}
df = pd.DataFrame(dados)
print(df)
```

Pandas permite a leitura de arquivos CSV para manipulação de dados.

```python
# Exemplo de leitura de arquivo CSV
df = pd.read_csv('dados.csv')
print(df.head())  # Exibe as primeiras 5 linhas do DataFrame
```

2.3: Tratamento de Dados com Pandas

Filtrar dados permite focar em subconjuntos específicos de dados. A agregação de dados ajuda a calcular estatísticas resumidas.

```python
# Exemplo de filtragem de dados
df_filtrado = df[df['Idade'] > 25]
print(df_filtrado)
```

```python
# Exemplo de agregação de dados
media_idade = df['Idade'].mean()
print(f"A média de idade é: {media_idade}")

# Exemplo de soma de valores
total_vendas = df['Vendas'].sum()
print(f"O total de vendas é: {total_vendas}")
```

03

Visualização de Dados

Veremos como criar visualizações de dados eficazes utilizando a biblioteca Matplotlib, e concluiremos com uma revisão dos conceitos aprendidos e um encorajamento para continuar sua jornada na Engenharia de Dados.

3.1: Introdução ao Matplotlib

Matplotlib é uma biblioteca de visualização que ajuda a criar gráficos a partir dos seus dados.

3.2: Gráficos de Barras e de Linha

Gráficos de barras são úteis para comparar categorias de dados, enquanto gráficos de linha são ideais para mostrar tendências ao longo do tempo.

```python
import matplotlib.pyplot as plt

# Exemplo de gráfico de barras
df['Idade'].plot(kind='bar')
plt.title('Idades dos Participantes')
plt.xlabel('Participantes')
plt.ylabel('Idade')
plt.show()
```

```python
# Exemplo de gráfico de linha
df['Vendas'].plot(kind='line')
plt.title('Vendas ao Longo do Tempo')
plt.xlabel('Tempo')
plt.ylabel('Vendas')
plt.show()
```

Agradecimentos

PYTHON PARA ENG. DE DADOS
DANIEL FARNEY

OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por um humano

Dominar Python é um passo crucial para se tornar um engenheiro de dados eficaz. Continue praticando e explorando as bibliotecas disponíveis para expandir suas habilidades.

Este Ebook é apenas o começo da sua jornada na Engenharia de Dados. Que a Força dos Dados esteja com você, até o próximo capítulo dessa jornada!