

PYTHON ALOUIMISTA

A arte de transformar dados brutos em ouro analítico



Aprenda a explorar as aplicações mais relevantes do Python no universo da análise de dados

GABRIELA SOARES

PYTHON E A ALQUIMIA DOS DADOS

A era dos dados chegou, e com ela, a necessidade de dominar ferramentas que transformam números, textos e imagens aparentemente desorganizados em insights valiosos. Python se destaca como a verdadeira "varinha de condão" dos cientistas de dados e entusiastas da análise. Vamos explorar as aplicações mais relevantes do Python no universo da análise de dados, abordando não apenas as ferramentas mais utilizadas, mas também como aplicá-las em cenários práticos para extrair o máximo de valor dos dados.



01

Coletando Dados de várias fontes

A Magia da Web Scraping com BeautifulSoup

COLETANDO DADOS DA WEB

Os dados estão espalhados por toda a internet, e Python nos ajuda a coletá-los com ferramentas como o BeautifulSoup. A coleta automatizada de informações possibilita reunir grandes volumes de dados que podem ser utilizados para análises detalhadas, desde monitoramento de preços até estudos de mercado.

```
import requests
from bs4 import BeautifulSoup

# URL de exemplo
url = "https://quotes.toscrape.com/"
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

# Extraíndo citações
quotes = [quote.text for quote in soup.find_all('span', class_='text')]
for i, quote in enumerate(quotes, start=1):
    print(f"Citação {i}: {quote}")
```

O que o código faz: Ele acessa uma página web de exemplo, extrai as citações presentes e as exibe no console. O requests realiza a requisição à URL, e o BeautifulSoup interpreta o HTML para encontrar os dados desejados. Essa abordagem permite transformar conteúdo web em fontes valiosas de informação.

02

Limpeza de dados: da bagunça ao brilho

Corrigindo Inconsistências com pandas

LIMPEZA DE DADOS

Nem todos os dados são perfeitos. Dados incompletos, formatos inconsistentes e entradas duplicadas podem comprometer a qualidade das análises. Felizmente, Python facilita a limpeza e organização usando a biblioteca pandas. Esse processo é essencial para garantir que os dados estejam preparados para serem analisados de maneira eficiente.

```
import pandas as pd

# Criando um DataFrame com dados desorganizados
data = {
    'Nome': ['Alice', 'Bob', None, 'Eve'],
    'Idade': ['25', 'trinta', '22', None],
    'Cidade': ['São Paulo', 'Rio de Janeiro', None, 'Curitiba']
}
df = pd.DataFrame(data)

# Limpando os dados
df['Idade'] = pd.to_numeric(df['Idade'], errors='coerce')
df.dropna(inplace=True)
print(df)
```

O que o código faz: Ele cria um DataFrame com dados inconsistentes, converte a coluna "Idade" para números e remove linhas contendo valores ausentes. Resultado: um DataFrame limpo e pronto para análise. Essas técnicas garantem que os dados estejam consistentes e relevantes.

03

Análise Exploratória: Conhecendo os Dados

Visualizando Tendências com matplotlib

CONHECENDO OS DADOS

Nada melhor do que um gráfico para enxergar padrões em um mar de dados. A visualização não apenas facilita o entendimento, mas também permite identificar tendências, outliers e relações importantes.

```
import matplotlib.pyplot as plt

# Dados de vendas mensais
meses = ['Jan', 'Fev', 'Mar', 'Abr', 'Mai']
vendas = [1000, 1200, 1100, 1500, 1700]

# Criando o gráfico
plt.plot(meses, vendas, marker='o')
plt.title('Vendas Mensais')
plt.xlabel('Meses')
plt.ylabel('Vendas (R$)')
plt.grid(True)
plt.show()
```

O que o código faz: Ele plota um gráfico de linha para representar as vendas ao longo de meses. O marcador 'o' destaca os pontos de dados, enquanto o grid torna a visualização mais clara. Essa abordagem oferece insights visuais que complementam análises numéricas.

04

Modelagem e Predição: A Alquimia Final

Treinando um Modelo de Machine Learning com scikit-learn

MODELAGEM E PREDIÇÃO

Uma análise realmente poderosa vai além dos dados presentes e tenta prever o futuro. Com Python e bibliotecas como scikit-learn, é possível criar modelos de aprendizado de máquina para resolver problemas complexos.

```
from sklearn.linear_model import LinearRegression
import numpy as np

# Dados de exemplo
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1) # Variável independente
y = np.array([100, 200, 300, 400, 500])     # Variável dependente

# Treinando o modelo
modelo = LinearRegression()
modelo.fit(X, y)

# Fazendo previsões
novo_X = np.array([6, 7, 8]).reshape(-1, 1)
previsoes = modelo.predict(novo_X)
print("Previsões:", previsoes)
```

O que o código faz: Ele cria um modelo de regressão linear para prever valores baseados em dados históricos. Neste exemplo, prevemos que os próximos pontos serão 600, 700 e 800. Esse tipo de modelo é amplamente utilizado para análises financeiras, previsões de demanda e outros cenários práticos.

05

Compartilhando Insights: Relatórios Automáticos

Gerando Relatórios com Jupyter Notebook

RELATÓRIOS AUTOMÁTICOS

Para cientistas de dados, não basta apenas obter insights; é preciso comunicá-los de forma clara e atraente. O Jupyter Notebook oferece uma plataforma ideal para combinar código, gráficos e textos explicativos em relatórios interativos.

```
# Exemplo dentro do Jupyter Notebook
# Título do relatório
"""
# Relatório de Análise de Vendas
Este relatório detalha as vendas mensais de forma gráfica e numérica.
"""

import pandas as pd

# Dados de vendas
data = {'Meses': ['Jan', 'Fev', 'Mar'], 'Vendas': [1000, 1200, 1100]}
df = pd.DataFrame(data)
df
```

O que o exemplo faz: Mostra como combinar Markdown e Python em um Jupyter Notebook para criar relatórios interativos e informativos, perfeitos para apresentações. Esse formato também facilita o compartilhamento de descobertas com equipes ou stakeholders.

AGRADECIMENTOS

OBRIGADA

Esse Ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu Github.

.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.



<https://github.com/GabrielaSR/prompts-recipe-to-create-a-ebook.git>

