Data Science

&

Inteligência Artificial

Fabiana L.Dias

1

Introdução a Data Science e Inteligência Artificial

São áreas em rápida ascensão, utilizadas para resolver problemas complexos em diversos setores. Este eBook tem como objetivo apresentar as principais bibliotecas e técnicas de maneira simples, com exemplos de códigos práticos.

2

Bibliotecas Fundamentais para Data Science com IA

Este capítulo explora as bibliotecas essenciais para manipulação de dados, criação de visualizações e implementação de modelos de aprendizado de máquina.

NumPy

É a base para operações matemáticas e manipulação de arrays multidimensionais, proporcionando eficiência e facilidade em cálculos científicos.

```python
import numpy as np

dados = [1, 2, 3, 4, 5]
array = np.array(dados)
print(array.mean())  # Média dos valores
```

Pandas

Especialista em manipulação de dados tabulares, como os encontrados em planilhas e bancos de dados. Permite uma análise rápida e eficiente.

```python
import pandas as pd

dados = {'Nome': ['Ana', 'Pedro'], 'Idade': [25, 30]}
df = pd.DataFrame(dados)
print(df.describe())  # Estatísticas descritivas
```

Matplotlib e Seaborn

As bibliotecas mais populares para criação de gráficos e visualizações personalizadas. Enquanto o Matplotlib é mais genérico, o Seaborn é voltado para visualizações estatísticas.

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
dados = [10, 20, 30, 40]
plt.plot(dados)
plt.title("Gráfico Simples")
plt.show()
```

Scikit-learn

Biblioteca robusta para aprendizado de máquina. Fornece algoritmos para classificação, regressão, clustering e muito

```
from sklearn.linear_model import LinearRegression

X = [[1], [2], [3]]  # Variável independente
y = [2, 4, 6]        # Variável dependente

modelo = LinearRegression()
modelo.fit(X, y)
print(modelo.predict([[4]]))  # Prevê o valor para X=4
```

TensorFlow e PyTorch

Focados em deep learning, essas bibliotecas ajudam na construção e treinamento de redes neurais para resolver problemas complexos.

```
import tensorflow as tf

model = tf.keras.Sequential([
    tf.keras.layers.Dense(units=1, input_shape=[1])
])
model.compile(optimizer='sgd', loss='mean_squared_error')

X = [1.0, 2.0, 3.0]
y = [2.0, 4.0, 6.0]
model.fit(X, y, epochs=500, verbose=0)
print(model.predict([4.0]))
```

3

CASE: Predição de Preços Imobiliários

Este case demonstra como usar regressão linear para estimar o preço de propriedades com base em dados como tamanho e localização.

Problema: Estimar o preço de casas com base em tamanho e localização.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Dados fictícios
dados = {'Tamanho (m2)': [50, 100, 150], 'Localização': [1, 2, 3],
         'Preço': [200000, 500000, 800000]}

df = pd.DataFrame(dados)
X = df[['Tamanho (m2)', 'Localização']]
y = df['Preço']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
modelo = LinearRegression()
modelo.fit(X_train, y_train)

preco_previsto = modelo.predict([[120, 2]])
print(f"Preço previsto: {preco_previsto[0]:.2f}")
```

4

CASE: Classificação de E-mails (Spam ou Não)

Mostra como implementar algoritmos de classificação para categorizar mensagens de e-mail como spam ou não, usando técnicas de processamento de linguagem natural.

Problema: Identificar se um e-mail é spam ou não.

```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

# Dados fictícios
emails = ["Compre agora", "Oferta exclusiva", "Reunião amanhã"]
labels = [1, 1, 0]  # 1 = Spam, 0 = Não Spam

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)

modelo = MultinomialNB()
modelo.fit(X, labels)

novo_email = ["Ganhe dinheiro agora"]
X_novo = vectorizer.transform(novo_email)
print(modelo.predict(X_novo))  # 1 = Spam
```

5

CASE:
Reconhecimento de Imagens Simples

Este case aborda o uso de redes neurais convolucionais para classificação de imagens, destacando conceitos fundamentais de deep learning.

Problema: Classificar imagens de gatos e cães.

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Supondo dados de treino já preparados
# model.fit(X_train, y_train, epochs=10)
```

6

Conclusão

Com ferramentas acessíveis e uma abordagem estruturada, você pode resolver problemas reais usando Data Science e IA. Teste os exemplos, explore dados do mundo real e construa projetos práticos para aperfeiçoar suas habilidades.