

The background image is a vertical, high-contrast photograph of a city street at night. The street is wet, reflecting the lights from buildings and street lamps. A digital rain effect, with green and blue vertical streaks, is overlaid on the image, creating a sense of data flow or digital rain. The perspective is looking down the street, with buildings on either side.

Matrix Learning

Dominando o Aprendizado de Máquinas

Guia Enxuto de Machine Learning

Introdução

Machine Learning (ML) é uma área da inteligência artificial que permite que computadores aprendam a realizar tarefas sem serem explicitamente programados. Neste eBook, abordaremos os principais tipos de aprendizado de máquina, suas metodologias, como preparar os dados e avaliar os resultados, com exemplos práticos em código usando Python.

1. Tipos de Aprendizado de Máquina

1.1 Aprendizado Supervisionado

No aprendizado supervisionado, o modelo é treinado com dados rotulados, ou seja, cada entrada vem acompanhada de uma resposta esperada.

Exemplo: Prever o preço de uma casa com base em atributos como tamanho e localização.

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

# Dados simulados
data = {
    'tamanho': [50, 60, 80, 100, 120],
    'localizacao': [1, 2, 2, 3, 3],
    'preco': [150, 180, 240, 300, 360]
}
df = pd.DataFrame(data)

# Preparar dados
X = df[['tamanho', 'localizacao']]
y = df['preco']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Treinar modelo
modelo = LinearRegression()
modelo.fit(X_train, y_train)

# Avaliar modelo
predicoes = modelo.predict(X_test)
mse = mean_squared_error(y_test, predicoes)
print(f'Erro Quadrático Médio: {mse:.2f}')
```

1.2 Aprendizado Não Supervisionado

O aprendizado não supervisionado trabalha com dados sem rótulos, buscando padrões ou agrupamentos.

Exemplo: Agrupar clientes em segmentos com base em seus comportamentos.

```
from sklearn.cluster import KMeans
import numpy as np

# Dados simulados
clientes = np.array([
    [20, 200],
    [25, 220],
    [30, 210],
    [60, 800],
    [65, 850],
])

# Treinar modelo
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(clientes)

# Resultados
print('Centroides:', kmeans.cluster_centers_)
print('Grupos:', kmeans.labels_)
```

1.3 Aprendizado por Reforço

Neste tipo, o modelo aprende interagindo com o ambiente e recebendo recompensas por suas ações.

Exemplo: Treinar um agente para jogar um jogo simples.

```
import numpy as np

# Simulação simples
estado = 0
recompensa_total = 0
acoes = [-1, 1]

for _ in range(10):
    acao = np.random.choice(acoes) # Escolha aleatória de ação
    estado += acao
    recompensa = 1 if estado == 5 else -1
    recompensa_total += recompensa
    print(f'Estado: {estado}, Recompensa: {recompensa}')

print(f'Recompensa Total: {recompensa_total}')
```

2. Preparação de Dados

2.1 Limpeza de Dados

Dados reais frequentemente contêm valores ausentes e inconsistências. A limpeza é essencial para modelos de qualidade.

```
# Exemplo de limpeza
import pandas as pd

# Dados simulados
dados = {
    'idade': [25, 30, None, 35, 40],
    'salario': [50000, 60000, 55000, None, 70000]
}
df = pd.DataFrame(dados)

# Preencher valores ausentes
print('Antes:
', df)
df['idade'].fillna(df['idade'].mean(), inplace=True)
df['salario'].fillna(df['salario'].median(), inplace=True)
print('Depois:
', df)
```

2.2 Normalização e Padronização

Modelos geralmente performam melhor com dados em escalas similares.

```
from sklearn.preprocessing import StandardScaler

# Dados simulados
valores = [[1, 1000], [2, 2000], [3, 3000]]

# Padronizar dados
scaler = StandardScaler()
dados_padronizados = scaler.fit_transform(valores)
print(dados_padronizados)
```

3. Avaliação de Resultados

3.1 Métricas para Modelos Supervisionados

Escolher a métrica certa ajuda a entender a performance do modelo.

- **Regressão:** Erro Quadrático Médio (MSE), Erro Absoluto Médio (MAE)
- **Classificação:** Acurácia, Precisão, Recall, F1-Score

```
from sklearn.metrics import classification_report

# Exemplo de classificação
verdadeiro = [1, 0, 1, 1, 0]
predito = [1, 0, 1, 0, 0]

print(classification_report(verdadeiro, predito))
```

3.2 Validação Cruzada

Validação cruzada é uma técnica para avaliar a generalização do modelo.

```
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier

# Dados simulados
X = [[1], [2], [3], [4], [5]]
y = [0, 0, 1, 1, 1]

# Modelo e validação
modelo = RandomForestClassifier()
scores = cross_val_score(modelo, X, y, cv=3)
print('Scores:', scores)
```


Conclusão

Machine Learning é um campo vasto, mas com fundamentos simples. Pratique com pequenos projetos e explore ferramentas e bibliotecas. Com o tempo, você dominará os conceitos e as aplicações mais complexas.