

# Módulo 8: Backend com C# e Python

## Dia 07: Python para IA - Machine Learning

João Quintino

29 de dezembro de 2025



- 1 O que é Machine Learning?
- 2 Regressão vs Classificação
- 3 Scikit-learn: A Biblioteca de ML
- 4 Pipeline de ML
- 5 Regressão Linear Simples
- 6 Prática: Predição de Preços de Imóveis
- 7 Encerramento

# Inteligência Artificial vs Machine Learning

- **IA:** sistemas que simulam inteligência humana.
- **ML:** subcampo da IA que aprende com dados (sem ser explicitamente programado).
- Exemplos: recomendação de filmes, detecção de fraudes, diagnóstico médico.

# Tipos de Aprendizado

## Principais categorias

- **Supervisionado:** treinar com dados rotulados (ex: preço de casas).
- **Não-supervisionado:** encontrar padrões sem rótulos (ex: segmentação de clientes).
- **Por reforço:** aprender por tentativa e erro (ex: jogos).

- Prever um **valor contínuo**.
- Exemplos: preço de imóveis, temperatura, vendas futuras.
- Algoritmos: Regressão Linear, Regressão Polinomial, etc.

- Prever uma **categoria/classe**.
- Exemplos: spam ou não-spam, gato ou cachorro, tipo de flor.
- Algoritmos: Regressão Logística, Árvores de Decisão, SVM, etc.

# Por que Scikit-learn?

## Biblioteca mais popular para ML clássico

- API simples e consistente.
- Dezenas de algoritmos prontos.
- Integração com NumPy e Pandas.
- Ótima documentação e comunidade.

## Instalar via pip

```
pip install scikit-learn
```



# Etapas de um Projeto de ML

- ❶ **Coleta de dados:** obter dataset (CSV, API, DB).
- ❷ **Preparação:** limpar, tratar nulos, normalizar.
- ❸ **Divisão:** treino e teste (ex: 80% treino, 20% teste).
- ❹ **Treinamento:** ajustar o modelo aos dados de treino.
- ❺ **Avaliação:** medir performance (MSE,  $R^2$ , acurácia).
- ❻ **Predição:** usar o modelo treinado em dados novos.

## Separando os dados

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

# O que é Regressão Linear?

- Modelo que tenta traçar uma **linha reta** pelos dados.
- Equação:  $y = a \cdot x + b$
- Objetivo: minimizar o erro entre valores reais e previstos.

# Exemplo: Treinar modelo

## Código básico

```
from sklearn.linear_model import LinearRegression

modelo = LinearRegression()
modelo.fit(X_train, y_train)
```

## Usando o modelo treinado

```
y_pred = modelo.predict(X_test)

print("Primeiras predicoes:", y_pred[:5])
```

## Métricas de avaliação

```
from sklearn.metrics import mean_squared_error,
    r2_score

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse:.2f}")
print(f"R2: {r2:.2f}")
```

$R^2$ : quanto mais próximo de 1, melhor o modelo explica os dados.

## Objetivo

Treinar um modelo de regressão linear para prever o preço de imóveis com base em:

- Área ( $\text{m}^2$ )
- Número de quartos
- Idade do imóvel

# Estrutura do dataset

## Exemplo de CSV (imoveis.csv)

```
area,quartos,idade,preco  
120,3,5,450000  
80,2,10,280000  
150,4,2,620000
```



# Pipeline completo

## Fluxo geral

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,
    r2_score

df = pd.read_csv("imoveis.csv")
X = df[["area", "quartos", "idade"]]
y = df["preco"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

modelo = LinearRegression()
modelo.fit(X_train, y_train)
y_pred = modelo.predict(X_test)
```

- Conceitos de IA e Machine Learning.
- Diferença entre regressão e classificação.
- Biblioteca Scikit-learn e pipeline de ML.
- Regressão Linear para prever valores contínuos.
- Prática: predição de preços de imóveis.

*Vamos construir IAs!*

- Explorar outros algoritmos (Decision Tree, Random Forest).
- Adicionar mais features ao modelo.
- Integrar predições de ML em APIs.