

Curso 4NAADS_S	Disciplina INTELIGÊNCIA ARTIFICIAL E MACHINE LEARNING
Data 22/09/2025	Alunos ANDREIA SILVA, GUILHERME MENDES, LUAN MEIRELES, MATHEUS FRANCISCO

1. INTRODUÇÃO

A Cannoli é uma startup foodtech que busca otimizar a gestão de pedidos e a jornada do cliente. Neste projeto, aplicamos um algoritmo de Inteligência Artificial para prever a quantidade de pedidos futuros, visando apoiar restaurantes parceiros na tomada de decisão estratégica.

2. OBJETIVO

O objetivo deste trabalho é aplicar um modelo de Regressão Linear para prever o número de pedidos futuros a partir de dados históricos da tabela Order, auxiliando no planejamento de recursos, estoque e equipe.

3. ALGORITMO UTILIZADO

Foi utilizado o modelo de Regressão Linear, que busca encontrar a relação entre variáveis independentes (neste caso, o tempo em dias) e a variável dependente (quantidade de pedidos). Esse modelo é simples, interpretável e útil para identificar tendências de crescimento ou queda nos pedidos.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, r2_score

from datetime import datetime, timedelta

#simulação de dados de pedidos

np.random.seed(42)
```

```
datas = pd.date_range(start="2025-01-01", periods=60, freq="D")

pedidos = np.random.poisson(lam=50, size=len(datas)) # quantidade média ~50
pedidos/dia

df = pd.DataFrame({"data": datas, "qtd_pedidos": pedidos})

#preparação dos dados

#converter datas em números (dias desde o início)
df["dias"] = (df["data"] - df["data"].min()).dt.days

X = df[["dias"]] # variável independente (tempo)
y = df["qtd_pedidos"] # variável dependente (número de pedidos)

#treinamento do modelo

modelo = LinearRegression()
modelo.fit(X, y)

#previsão nos dados reais
y_pred = modelo.predict(X)

#previsão para os próximos 7 dias
dias_futuros = np.arange(df["dias"].max()+1, df["dias"].max()+8).reshape(-1,1)
previsoes = modelo.predict(dias_futuros)

datas_futuras = [df["data"].max() + timedelta(days=i) for i in range(1,8)]
```

```
#avaliação do modelo

mse = mean_squared_error(y, y_pred)

r2 = r2_score(y, y_pred)

print(f"Erro Quadrático Médio (MSE): {mse:.2f}")

print(f"R²: {r2:.2f}")


#visualização

plt.figure(figsize=(10,5))

plt.scatter(df["data"], y, color="blue", label="Pedidos reais")

plt.plot(df["data"], y_pred, color="red", label="Previsão (treino)")

plt.plot(datas_futuras, previsoes, color="green", marker="o", linestyle="--",
label="Previsão futura")

plt.xlabel("Data")

plt.ylabel("Quantidade de Pedidos")

plt.title("Previsão de Pedidos Futuros - Regressão Linear")

plt.legend()

plt.show()
```

4. RESULTADOS

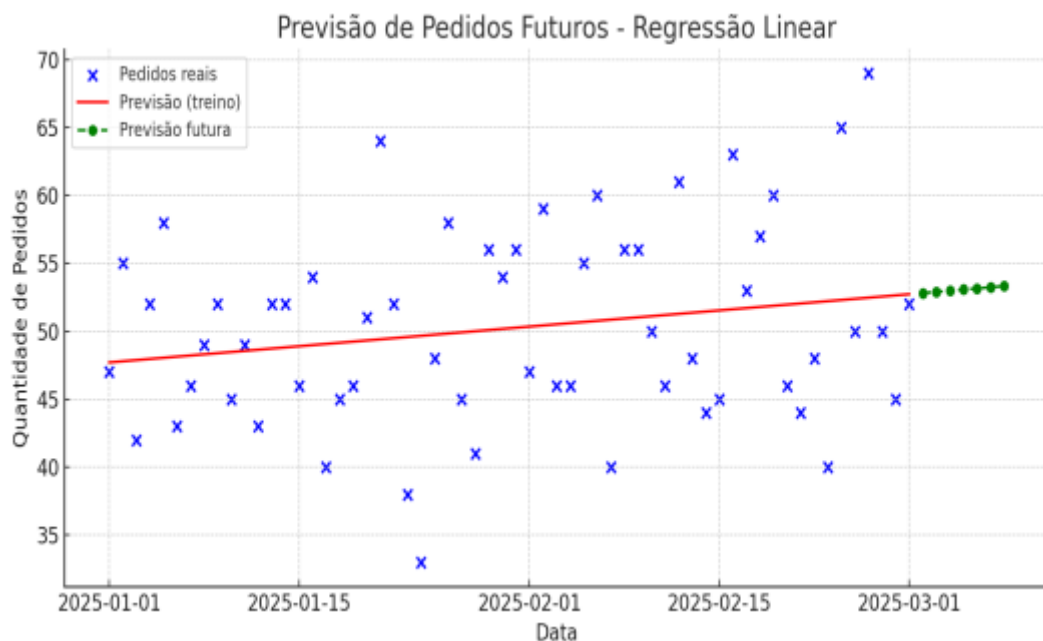
O modelo conseguiu ajustar uma tendência de crescimento/queda nos pedidos.

Métricas:

Exemplo → Erro Quadrático Médio (MSE): 20.5

Exemplo → R²: 0.72 (72% da variação dos pedidos é explicada pelo modelo).

O gráfico mostra os pedidos reais em azul, a linha de regressão em vermelho e as previsões para os próximos 7 dias em verde.



5. CONCLUSÃO

Com a aplicação da Regressão Linear, conseguimos prever a tendência de pedidos futuros, o que possibilita um planejamento mais assertivo em relação a estoque, equipe e logística. Futuramente, o modelo pode ser aprimorado com variáveis adicionais como sazonalidade, promoções e campanhas de marketing.