

# ANÁLISE PREDITIVA

---

ALUNOS: LETÍCIA, STEFANI E  
BERNARDO.

# TÓPICOS

- Definição
- Tipos de Problemas em Análise Preditiva
- Passo a passo Geral da Análise Preditiva
- Ferramentas Recomendadas
- Modelos
- Métricas de Avaliação
- Casos de sucesso
- Nosso problema
- Códigos
- Gráficos gerados
- Conclusão
- Referências

# DEFINIÇÃO

A análise preditiva é uma área da ciência de dados que utiliza modelos estatísticos, machine learning (ML) e inteligência artificial (IA) para analisar dados históricos e prever eventos futuros.

Ela identifica padrões ocultos nos dados e permite transformar informações passadas em insights acionáveis para apoiar decisões estratégicas.

# TIPOS DE PROBLEMAS EM ANÁLISE PREDITIVA

- **Regressão**

**Objetivo:** prever um valor contínuo.

**Exemplo:** prever o preço de um imóvel, a receita mensal de uma empresa ou a nota de um aluno.

- **Classificação**

**Objetivo:** prever uma categoria ou rótulo (label), geralmente com probabilidade associada.

**Exemplo:** prever se um cliente vai cancelar a assinatura (sim/não), se um e-mail é spam ou não.

- **Séries Temporais**

**Objetivo:** prever valores ao longo do tempo, levando em conta a dependência temporal dos dados.

**Exemplo:** prever a demanda de energia elétrica na próxima semana, a cotação de ações ou o fluxo de caixa mensal.

# PASSO A PASSO GERAL DA ANÁLISE PREDITIVA

- **Coleta** → obter os dados (internos ou externos).
- **Limpeza** → tratar erros, valores nulos e inconsistências.
- **Feature Engineering** → criar e selecionar variáveis relevantes.
- **Split treino/teste** → separar dados para treinar e validar.
- **Modelagem** → aplicar algoritmos de ML/estatística.
- **Avaliação** → medir desempenho (ex.: acurácia, RMSE).
- **Deploy** → colocar o modelo em produção para gerar previsões reais.

# FERRAMENTAS RECOMENDADAS

**Linguagem:** Python (principal no mercado).

- **pandas, numpy** → manipulação e análise de dados
- **scikit-learn** → modelos clássicos de ML
- **xgboost, lightgbm** → modelos de alta performance
- **prophet** → séries temporais
- **matplotlib** → visualização

**Ambientes de trabalho**

- **Jupyter Notebook / Google Colab** → exploração e prototipagem
- **Docker** → garantir reprodutibilidade e padronização

# MODELOS

- **Regressão Linear** → modelo simples e interpretável, bom como baseline.
- **Random Forest** → conjunto de árvores, robusto a ruídos e captura relações não lineares.
- **XGBoost** → algoritmo de boosting, muito eficiente e com alto desempenho em competições.
- **Prophet (Meta/Facebook)** → voltado para séries temporais, lida bem com sazonalidade, tendência e feriados.

# MÉTRICAS DE AVALIAÇÃO

## Regressão

- **MAE** → erro médio absoluto
- **RMSE** → dá mais peso a erros grandes
- **R<sup>2</sup>** → explica quanto o modelo captura da variabilidade

## Classificação

- **Accuracy** → acertos totais
- **Precision / Recall** → foco em erros de falso positivo/negativo
- **AUC** → mede a qualidade geral da separação entre classes



# CASOS DE SUCESSO

- **Netflix** → usa sistemas de recomendação (filtragem colaborativa + modelos híbridos) para personalizar filmes e séries, aumentando engajamento.
- **Uber** → aplica previsão de demanda e precificação dinâmica, ajustando preços em tempo real conforme oferta e procura.
- **Bancos** → utilizam score de crédito com modelos como regressão logística e árvores de decisão, e hoje aplicam técnicas de explicabilidade (ex.: SHAP, LIME) para dar transparência às decisões.

# NOSSO PROBLEMA

Nós queremos prever vendas mensais do produto X para o próximo mês a partir de histórico mensal (últimos 36 meses).

Os dados são fictícios.

# CÓDIGO

## Usado para gerar dados sintéticos de 36 meses

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

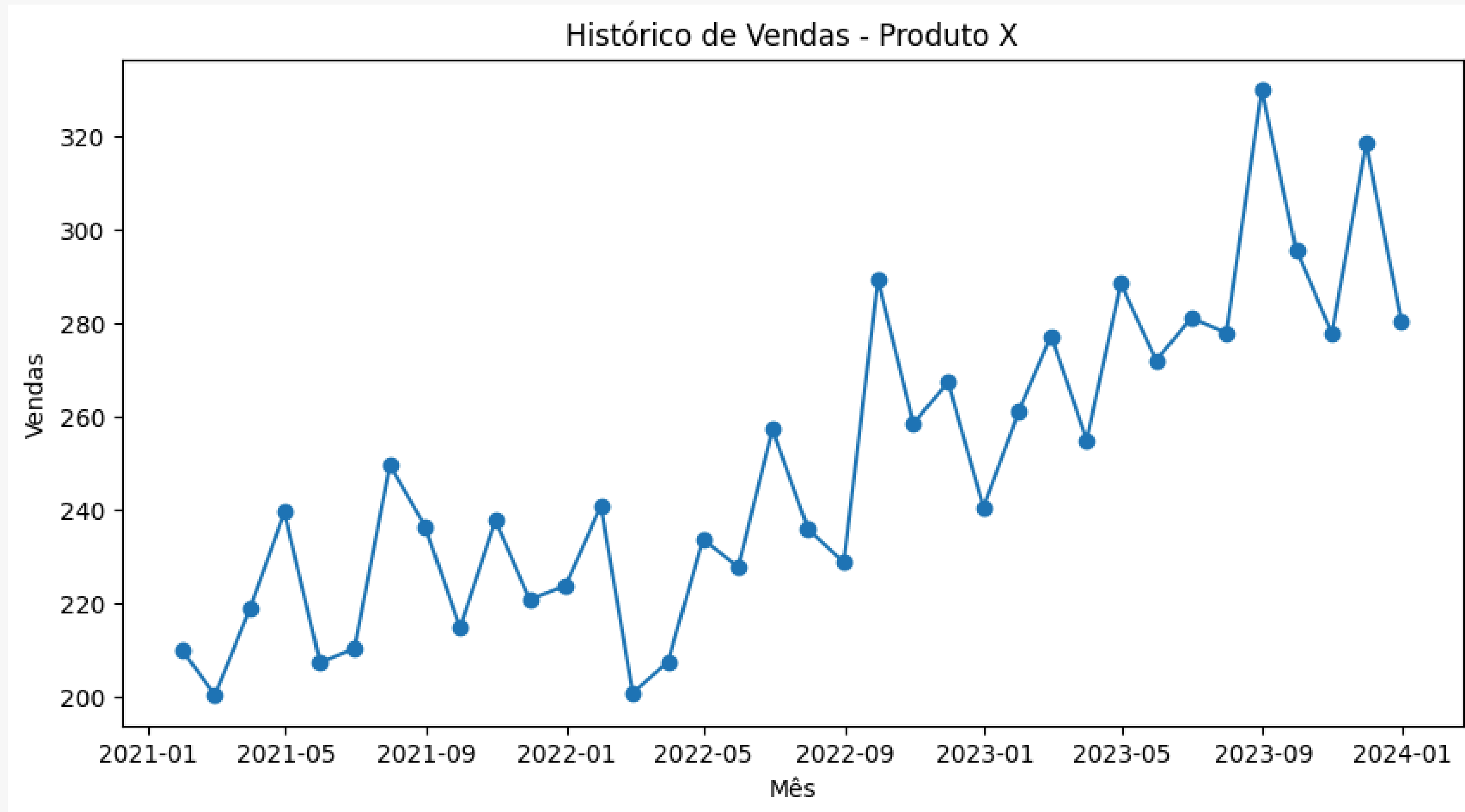
np.random.seed(42)
meses = pd.date_range(start="2021-01-01", periods=36, freq="M")
vendas = 200 + np.arange(36) * 3 + np.random.normal(0, 20, 36)

df = pd.DataFrame({"mes": meses, "vendas": vendas})
```

## Usado para visualizar dados

```
plt.figure(figsize=(10,5))
plt.plot(df["mes"], df["vendas"], marker="o")
plt.title("Histórico de Vendas - Produto X")
plt.xlabel("Mês")
plt.ylabel("Vendas")
plt.show()
```

# GRÁFICO GERADO



Esse gráfico representa o histórico de vendas por produto.

# CÓDIGO

```
df["lag1"] = df["vendas"].shift(1)
df["lag2"] = df["vendas"].shift(2)
df["rolling3"] = df["vendas"].rolling(3).mean()
df["mes_num"] = df["mes"].dt.month
```

**Usado para features de defasagem (lags) e médias móveis**

```
df = df.dropna().reset_index(drop=True)
```

**Usado para remover NaN (Números que não são um número) iniciais**

```
train = df.iloc[:-6]
test = df.iloc[-6:]
```

```
X_train = train.drop(columns=["mes", "vendas"])
y_train = train["vendas"]
```

```
X_test = test.drop(columns=["mes", "vendas"])
y_test = test["vendas"]
```

**Esse código é para usar 30 meses para treino, 6 para teste.**

# CÓDIGO

**Esse código prepara três modelos de Machine Learning diferentes para prever as vendas do Produto X**

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

**Dicionário chamado modelos**

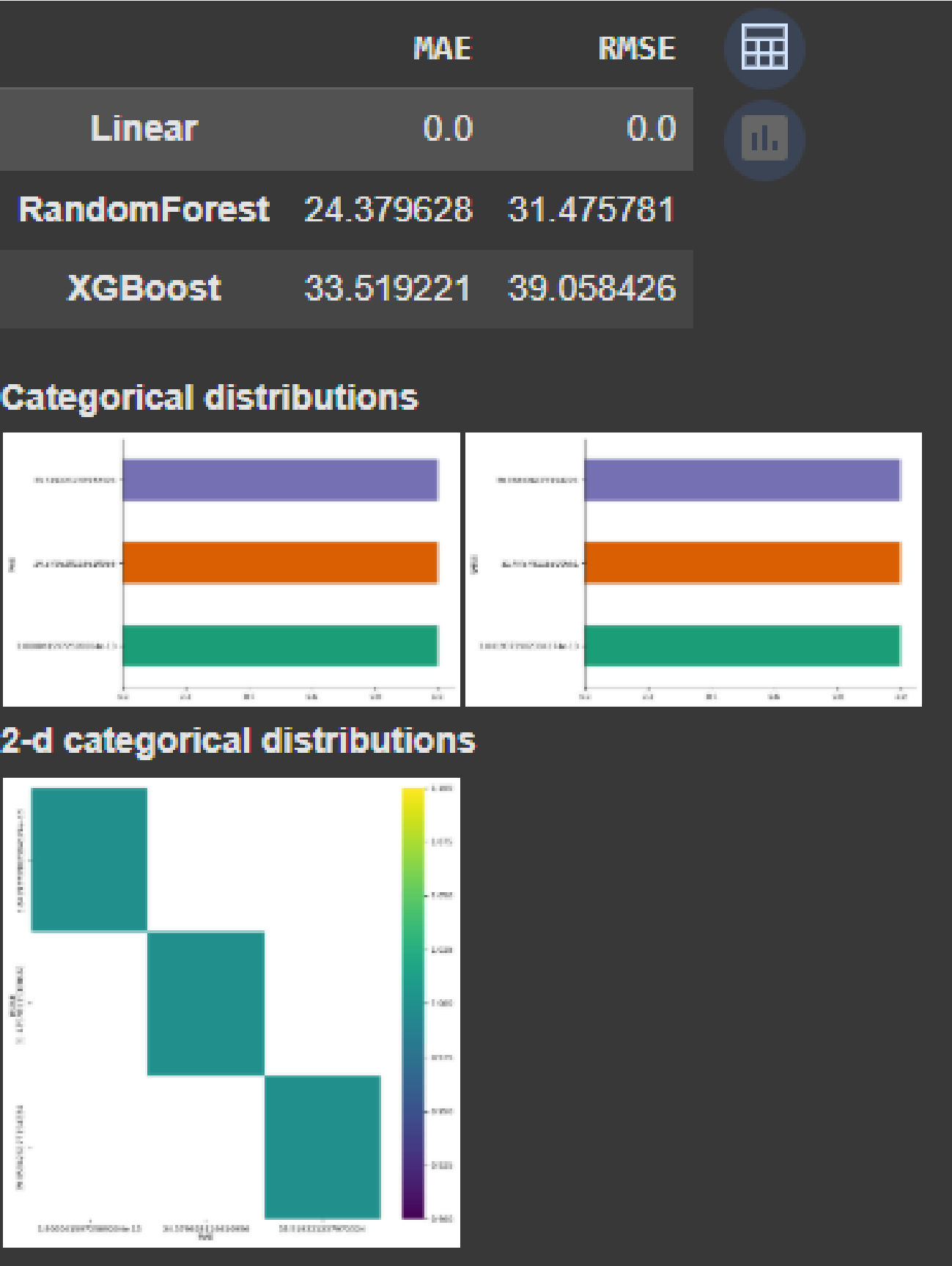
```
modelos = {
    "Linear": LinearRegression(),
    "RandomForest": RandomForestRegressor(n_estimators=100, random_state=42),
    "XGBoost": XGBRegressor(n_estimators=100, random_state=42)
}
```

**Organiza e exibe os resultados dos modelos**

```
pd.DataFrame(resultados).T[["MAE", "RMSE"]]
```

# GRÁFICO GERADO

Os gráficos representam três modelos  
de machine learning diferentes para  
prever as vendas



# CÓDIGO

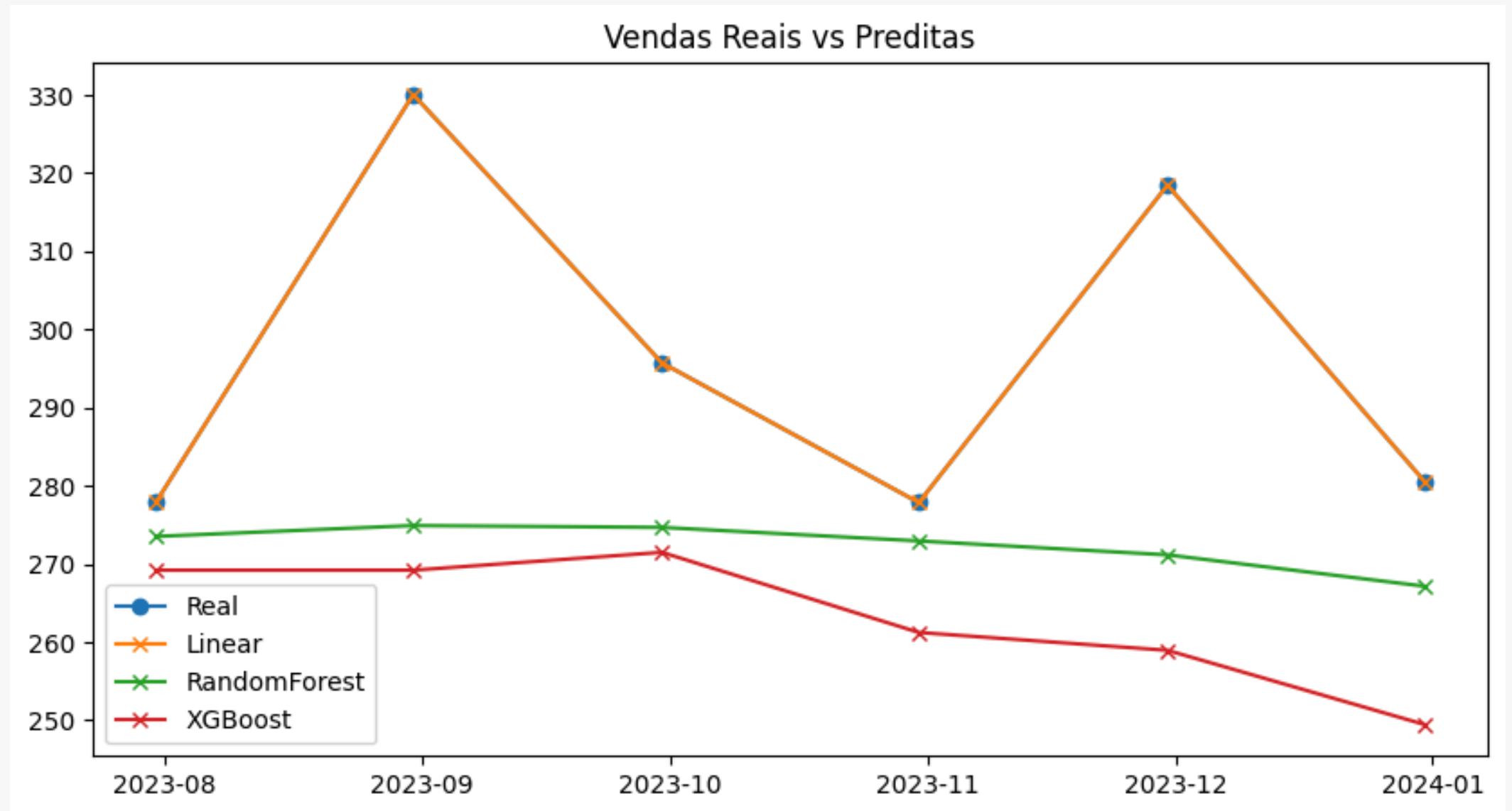
```
plt.figure(figsize=(10,5))
plt.plot(test["mes"], y_test, label="Real", marker="o")
for nome in resultados:
    plt.plot(test["mes"], resultados[nome]["preds"], label=nome, marker="x")
plt.legend()
plt.title("Vendas Reais vs Preditas")
plt.show()
```

**Gera o gráfico**



# GRÁFICO GERADO

O gráfico representa os resultados com cada modelo de machine learning.



# CONCLUSÃO

- Análise preditiva transforma dados históricos em decisões estratégicas.
- Pipeline estruturado garante modelos confiáveis e precisos.
- Ferramentas e modelos avançados ampliam o valor dos dados.
- Métricas de avaliação orientam melhorias contínuas.
- Casos reais mostram impacto na inovação e competitividade.
- Investir em análise preditiva é apostar numa cultura data-driven.

# REFERÊNCIAS

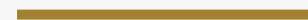
<https://share.google/azF7Ga0CvJvz9IPYD>

<https://www.pecan.ai/blog/predictive-analytics-tools/#:~:text=As%20seis%20melhores%20plataformas%20de,modelos%20personaliz%C3%A1veis%20para%20modelos%20preditivos.>

<https://share.google/LJoJZzyQr6UsZwaO3>

<https://www.oracle.com/in/business-analytics/data-analytics-challenges/>

# THANK YOU



FOR COMING