

PREVISÃO DE LONGO PRAZO DE SÉRIES TEMPORAIS DIÁRIAS UTILIZANDO MODELOS BASEADOS EM ÁRVORES DE REGRESSÃO

Getro Lucas da Silva Filho

Universidade Federal Rural do Semi-Árido
Av. Francisco Mota, 572, Mossoró-RN, CEP: 59625-900
getrolucasfilho@gmail.com

Amanda Gondim De Oliveira

Universidade Federal Rural do Semi-Árido
Av. Francisco Mota, 572, Mossoró-RN, CEP: 59625-900
amandagondim@ufersa.edu.br

Angélica Felix De Castro

Universidade Federal Rural do Semi-Árido
Av. Francisco Mota, 572, Mossoró-RN, CEP: 59625-900
angelica@ufersa.edu.br

RESUMO

Este trabalho busca avaliar a aplicação de modelos de regressão baseados em árvores de decisão, como Florestas Aleatórias e XGBoost, na previsão de longo prazo de séries temporais diárias. Foram comparadas abordagens clássicas como Naive Sazonal, SARIMA e ETS com os métodos de *machine learning*, utilizando dados reais de venda de um e-commerce. O estudo incluiu etapas de análise exploratória, engenharia e seleção de características e otimização de hiperparâmetros. Os resultados mostraram que os modelos baseados em árvores apresentaram melhor desempenho em termos de raiz do erro quadrático médio (REQM), superando os modelos estatísticos clássicos, especialmente em séries com sazonalidades complexas. A análise reforça a importância do uso de variáveis explicativas bem construídas para melhorar as previsões de longo prazo de séries temporais.

PALAVRAS CHAVE. Séries Temporais de Longo Prazo, Random Forest, Gradient Boosting.

AS&DS – Análise e Ciência de Dados; EST&AM – PO Analytics em Estatística e Aprendizado de Máquina

ABSTRACT

This study aims to evaluate the application of tree-based regression models, such as Random Forest and XGBoost, in the long-term forecasting of daily time series. Classical approaches such as Seasonal Naive, SARIMA, and ETS were compared with machine learning methods using real sales data from an e-commerce platform. The study included exploratory analysis, feature engineering and selection, and hyperparameter optimization. The results showed that tree-based models outperformed classical statistical models in terms of root mean squared error (RMSE), particularly in series with complex seasonality. The analysis highlights the importance of well-constructed explanatory variables in improving long-term time series forecasts.

KEYWORDS. Long-term Time Series, Random Forest, Gradient Boosting.

AS&DS – Analysis and Data Science; EST&AM – OP Analytics in Statistics and Machine Learning

1. Introdução

A análise e previsão de séries temporais desempenham um papel crucial e são amplamente aplicadas em vários campos de pesquisa e setores da indústria, incluindo meteorologia, economia, engenharias, logística e marketing. O entendimento correto do processo gerador das séries e produção de previsões acuradas permitem antecipar eventos e tomar melhores decisões, garantindo estabilidade, segurança e redução de custos. Para este fim, modelos clássicos de séries temporais, como Autorregressivo Integrado de Médias Móveis (ARIMA) e sua versão para séries sazonais (SARIMA), ou modelos baseados em Suavização Exponencial são amplamente utilizados, tanto pela eficácia quanto pelo baixo custo computacional, principalmente em contextos em que há a necessidade de previsão de múltiplas séries com atualização frequente. Mais recentemente, com o avanço da computação e a consequente popularização das técnicas de *machine learning*, novas propostas aos modelos estatísticos clássicos têm surgido, como modelos de regressão baseados em árvores de decisão, assim como diversas técnicas de engenharia de características e identificação e tratamento de anomalias têm sido desenvolvidas com uso de *machine learning*.

Séries temporais são sequências de observações ordenadas e espaçadas em intervalos de tempo geralmente iguais [Wei 2006]. Essas características fazem com que a análise e a previsão desses dados se distingam de outros problemas de regressão, uma vez que há uma dependência temporal inerente às observações. Diferentemente dos modelos clássicos de séries temporais, os modelos baseados em árvores não pressupõem estrutura nos dados, por isso, seu uso na modelagem de séries temporais exige um esforço adicional na construção de variáveis de regressão capazes de capturar adequadamente essas dependências, tornando esses modelos mais adequados à previsão desse tipo de dado.

Este artigo tem como objetivo comparar o desempenho de abordagens clássicas com modelos de *machine learning* baseados em árvores na previsão de longo prazo de séries temporais de vendas diárias. Foram implementados três modelos tradicionais: Naive Sazonal, ARIMA e ETS. Dentre os modelos baseados em árvores, foram utilizados Árvore de Decisão, Florestas Aleatórias e *Extreme Gradient Boosting Regressor* (XGBoost). Além disso, o estudo incluiu etapas fundamentais para a construção e validação dos modelos de *machine learning*, como a construção e seleção de variáveis preditoras e otimização de hiperparâmetros.

O artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica dos modelos utilizados no experimento. A Seção 3 descreve a metodologia adotada, incluindo a análise e o pré-processamento dos dados, a engenharia e seleção de *features*, e a otimização de hiperparâmetros. Por fim, a Seção 4 apresenta e discute os resultados obtidos, comparando o desempenho dos modelos e avaliando a aplicabilidade de técnicas de *machine learning* no contexto prático da previsão de séries temporais.

2. Fundamentação Teórica

2.1. Naive Sazonal

Os métodos ingênuos (Naive) para séries temporais são os métodos mais simples e rápidos de serem implementados, os quais resumem a previsão como repetição dos últimos valores observados – algumas variações do método usam a média dos n últimos valores. A implementação sazonal do método ingênuo (Naive Sazonal) usa os últimos valores observados em períodos semelhantes [Hyndman e Athanasopoulos 2021], e pode ser representado matematicamente da seguinte forma:

$$\hat{y}_{t+1} = y_{t-m},$$

onde t é o índice temporal atual, $\hat{y}_{t+1}$ é o valor estimado para o próximo período, m é a frequência sazonal dos dados e y_{t-m} é o último valor observado para o mesmo período do tempo [Svetunkov 2023]. Apesar da simplicidade, o método é bastante eficiente na previsão de séries com sazonalidade bem definida e com baixa variação, servindo de ponto de partida para diversas análises e comparação de modelos.

2.2. SARIMA

Autorregressivo Integrado de Médias Móveis Sazonal (SARIMA) é um dos mais populares métodos estatísticos e um dos modelos de referência para avaliação de novas abordagens preditivas em séries temporais [Ng et al. 2023]. O modelo é expresso pela combinação linear das defasagens da própria série (AR), erros residuais passados associados a um modelo de médias móveis (MA), aplicados à série diferenciada (I), cada um calculado de maneira sazonal e não-sazonal [Hyndman e Athanasopoulos 2021]. A equação geral do modelo pode ser escrita como:

$$\hat{y}'_t = c + \sum_{n=1}^p \alpha_n y'_{t-n} + \sum_{n=1}^q \theta_n \varepsilon_{t-n} + \sum_{n=1}^P \phi_n y'_{t-ns} + \sum_{n=1}^Q \eta_n \varepsilon_{t-ns} + \varepsilon_t,$$

onde $\hat{y}'_t$ é o valor estimado no tempo t , c é o intercepto do modelo e y'_t é o valor da série diferenciado de ordem d (não-sazonal) e D (sazonal), que são os termos de integração (I). α_n e θ_n são os coeficientes associados aos termos AR e MA das partes não sazonais, enquanto ϕ_n e η_n associam-se à parte sazonal de período s . Os parâmetros p , q , P e Q indicam as ordens dos termos AR e MA não sazonais e sazonais. Os termos de erro associados às médias móveis sazonais e não sazonais são dados por ε_t , enquanto ε_t é o erro aleatório da regressão. A seleção dos parâmetros do modelo SARIMA pode ser realizada de forma automática por meio de algoritmos de otimização baseados em critérios de informação, como Critério de Informação de Akaike (AIC), que testa diversas combinações de parâmetros e escolhe aquela que minimiza o AIC.

2.3. ETS

O modelo de Suavização Exponencial com Erro, Tendência e Sazonalidade (ETS), assim como o ARIMA, é uma das abordagens clássicas amplamente utilizadas na previsão de séries temporais. Sua principal característica é a decomposição da série em três componentes — erro, tendência e sazonalidade — que podem ser combinados de forma aditiva ou multiplicativa, garantindo flexibilidade para captar e representar séries com diferentes padrões. A previsão é realizada por meio de uma média ponderada dos valores históricos, em que os pesos atribuídos diminuem exponencialmente ao longo do tempo de acordo com as constantes de suavização.

Dados os diferentes tipos de componentes possíveis para o modelo e suas interações, o ETS pode assumir 30 combinações diferentes [Svetunkov 2023]. Como exemplo, podemos definir o modelo com componentes de tendência e sazonalidade aditivos por:

$$\begin{aligned}\hat{y}_t &= l_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t, \\ l_t &= l_{t-1} + b_{t-1} + \alpha \varepsilon_t, \\ b_t &= b_{t-1} + \beta \varepsilon_t, \\ s_t &= s_{t-m} + \gamma \varepsilon_t,\end{aligned}$$

onde l_t representa o nível da série, b_t a tendência, s_t o componente sazonal, m a frequência sazonal e ε_t o erro no tempo t . Os parâmetros α , β e γ controlam a suavização de cada componente e são estimados a partir dos dados. A melhor combinação de componentes depende das características da série e pode, de maneira semelhante ao ARIMA, ser automaticamente escolhido através de critérios de informação.

2.4. Árvores de Decisão

Árvores de Decisão usadas para regressão é uma técnica não paramétrica que emprega um conjunto de variáveis explicativas para prever uma variável objetivo. O método particiona um espaço de variáveis X em regiões distintas e disjuntas: $R_1, \dots, R_j$. A previsão para um novo conjunto de covariáveis é dada por:

$$\hat{y}(x_t) = \frac{1}{|\{i: x_i \in R_j\}|} \sum_{i: \{x_i \in R_j\}} y_i,$$

sendo $\hat{y}_t$ é a média dos valores y observados na região R_j a qual a observação x_t pertence. A construção da árvore ocorre de maneira recursiva encontrando a divisão dos dados que minimize a soma dos erros quadrados dentro de cada região R_j [Izbicki e Santos 2020].

Devido a esta característica, o modelo produz resultados facilmente interpretáveis, mas é sensível às variações nos dados de treino e pode apresentar viés se a árvore for muito profunda.

2.5. Florestas Aleatórias

Para contornar a limitação das Árvores de Decisão de apresentar viés e ter baixo poder preditivo existe o método de Florestas Aleatórias [Breiman 2001], que combina diversas árvores treinadas com amostras *bootstrap* dos dados. Esta abordagem é nomeada *Bagging (Bootstrap Aggregating)* e visa reduzir o viés dos modelos mais simples [Spiliotis 2022] combinando-os para construir um modelo mais robusto e menos propenso ao *overfitting*, que ocorre quando um modelo se ajusta demais aos dados de treino e não consegue generalizar.

Portanto, a previsão final do método pode ser expressa por:

$$\hat{y}(x_t) = \frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}(x_t),$$

onde B é o número de árvores treinadas e $\hat{y}^{(b)}$ é o valor estimado em cada árvore, resultando na média dos valores estimados. Em cada árvore, um subconjunto aleatório das variáveis é selecionado. O resultado não é tão interpretável quando de uma árvore simples, mas uma medida de importância de cada covariável pode ser calculada pela redução da soma dos erros quadráticos médios de cada divisão realizada, assim, tem-se uma medida para avaliar a importância de cada atributo [Izbicki e Santos 2020], que traz certa explicabilidade ao modelo, além de permitir selecionar os atributos mais importantes e reduzir a complexidade.

2.6. XGBoost

O XGBoost [Chen e Guestrin 2016] é uma variação dos métodos de *boosting* projetado para ser computacionalmente eficiente, rápido e escalável, uma vez que é executado em paralelo. *Boostings* são modelos que consistem em agregar estimadores de regressão mais fracos, como árvores de regressão de profundidade limitada, para compor um estimador forte. Diferente do método Random Forest, o *boosting* constrói o modelo geral a partir do incremento recursivo dos modelos mais fracos, de forma que cada estimador $f(x)$ é ajustado pela adição de uma nova função que prevê os resíduos r_i multiplicada por uma taxa de aprendizado λ , evitando *overfitting* [Izbicki e Santos 2020]. A atualização dos estimadores e dos resíduos é realizada por:

$$f_i(x) = f_{i-1}(x) + \lambda f^{(b_i)}(x), \\ r_i = y_i - f(x),$$

onde b é o *boosting round*, $f_i(x)$ é a função de predição atualizada, $f_{i-1}(x)$ é a função gerada pelo *boosting round* anterior, $f^{(b_i)}(x)$ é a função incremental que visa minimizar os resíduos r_i .

Diferentes hiperparâmetros podem ser configurados, como quantidade de árvores e profundidade delas, taxa de aprendizado, além do percentual de *features* e amostras utilizadas em cada estimador.

3. Metodologia

3.1. Análise Exploratória dos Dados

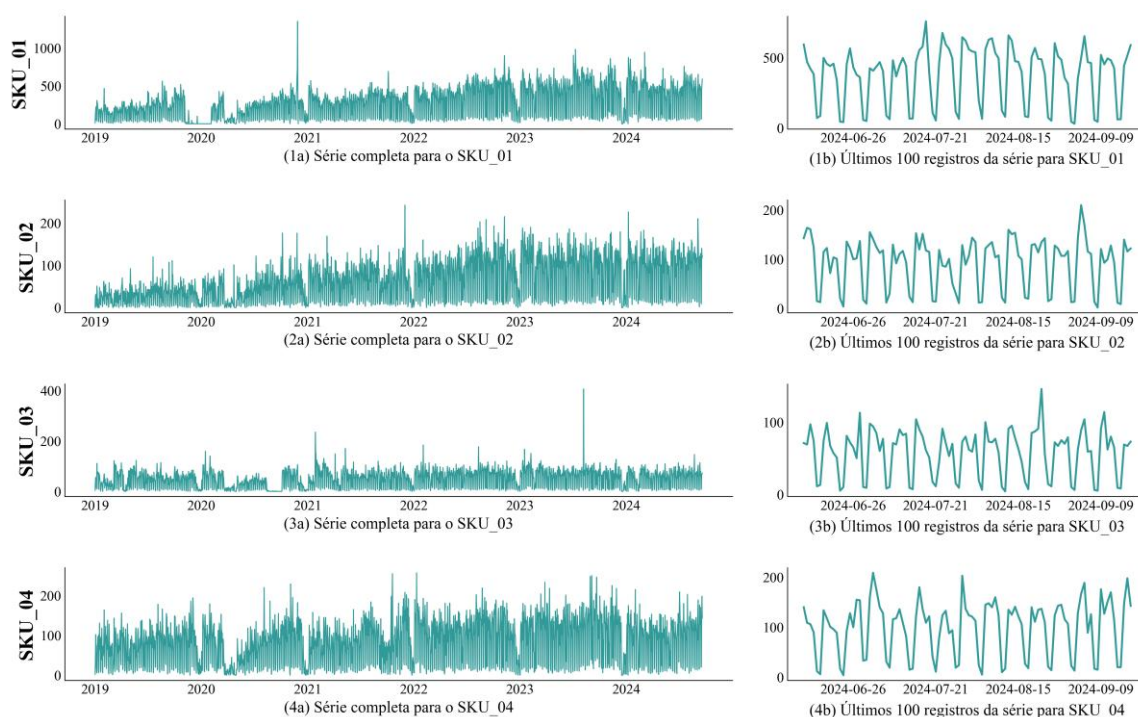
O estudo usou dados reais de vendas de um *e-commerce* de insumos médicos e odontológicos. O *dataset* contém dados diários de vendas de 4 produtos distintos, com vendas a partir de 1 de janeiro de 2019 até 18 de setembro de 2024, totalizando 2088 registros para cada produto, os quais recebem os nomes genéricos de SKU para manter a privacidade das informações fornecidas. Por motivos estratégicos e de segurança, dados de preço e estoque não foram incluídos, os quais poderiam melhorar a análise e predição, considerando que são variáveis que geralmente apresentam alta correlação com o volume de vendas. Devido à ausência de variáveis explicativas internas ao negócio, uma parte essencial da pesquisa se dá na extração de variáveis explicativas a partir dos próprios dados de venda, bem como de outras informações disponíveis externamente, e avaliar o poder preditivo dessas informações. A Tabela 1 representa a descrição dos dados de vendas para cada produto.

Tabela 1: Estatísticas Descritivas dos Dados

SKU	Média	Desvio Padrão	Mínimo	25%	50%	75%	Máximo
SKU_01	271,55	205,25	0	66	265	431	1350
SKU_02	61,24	48,57	0	15	53	102	242
SKU_03	47,15	35,64	0	11	51	75	406
SKU_04	81,45	55,58	0	22	88	126	257

A primeira e mais básica etapa do processo de análise exploratória de uma série temporal é a visualização da série por meio de gráficos de linha [Joseph e Tackes 2024], sendo possível notar padrões sazonais, tendências e variações, bem como notar a presença de outliers. A Figura 1 mostra as séries completas como também os últimos cem registros para cada produto.

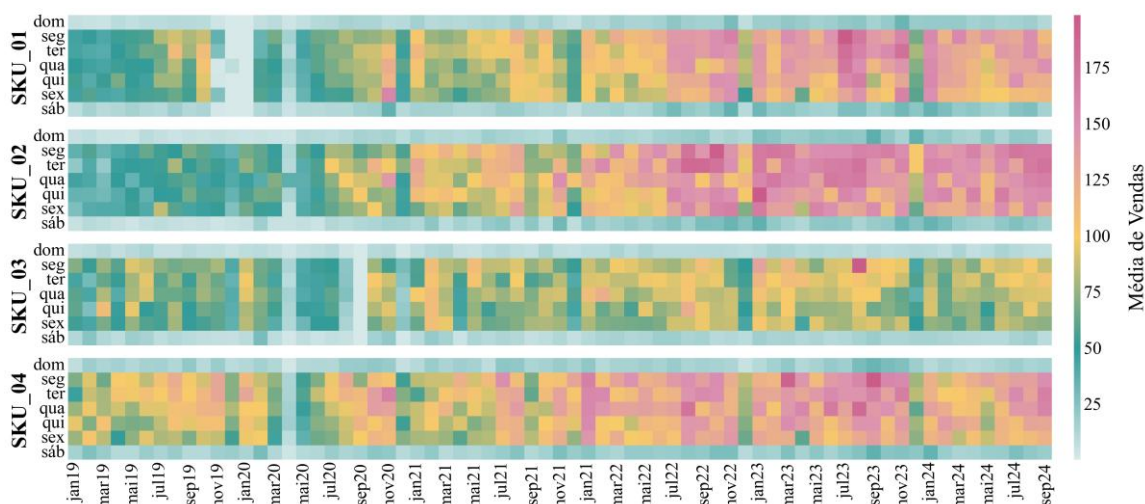
Figura 1: Séries completas e últimos cem registros por produto



Os gráficos de linha permitiram identificar, de maneira intuitiva, algumas características nas séries, tais como: (1) tendência de crescimento nos produtos 01 e 02; (2) forte padrão sazonal em todas as quatro séries; e (3) presença de outliers, o que direciona a análise exploratória para uma investigação mais detalhada desses padrões.

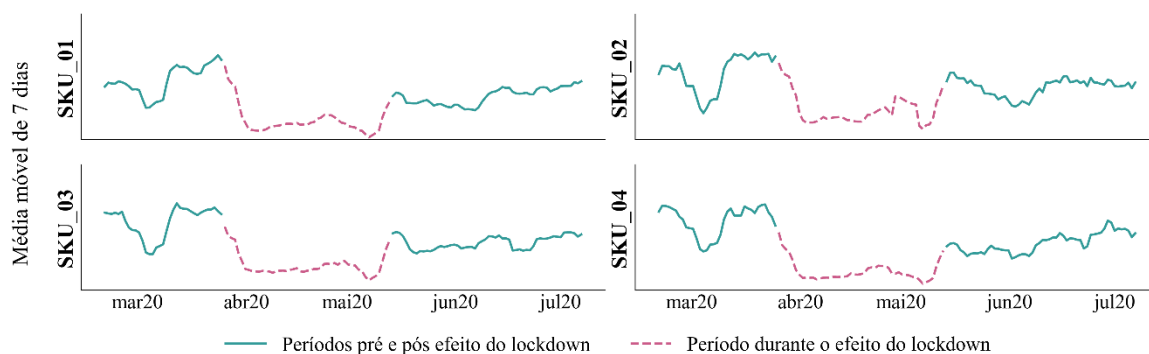
Para aprofundar a análise, a Figura 2 apresenta mapas de calor para séries temporais [Lye e Hirschberg 2020], em que cada ponto representa a média das vendas por dia da semana, segmentadas por mês e ano. A partir do gráfico é possível notar a tendência de crescimento nas séries 01, 02 e 04. O padrão sazonal também pode ser percebido, especialmente aos sábados e domingos, onde há cores mais suaves em todas as séries, indicando vendas mais baixas nestes dias.

Figura 2: Mapa de calor da venda média por dia da semana, mês e ano



Um ponto adicional relevante é a queda significativa nas médias observadas em abril de 2020, período que coincide com o início dos lockdowns no Brasil em decorrência da pandemia de COVID-19. A Figura 3 apresenta a média móvel de ordem 7 entre as datas 07 de fevereiro e 1º de julho de 2020, onde fica clara a mudança de tendência no mês de abril, quando houve o início do lockdown. Essa variação poderá ser incorporada como uma variável explicativa, contribuindo potencialmente para a melhoria da performance dos modelos.

Figura 3: Efeito da pandemia nas vendas



3.2. Pré-processamento dos Dados

Para a modelagem e avaliação do desempenho dos modelos, cada série temporal foi dividida em duas partes: o ano mais recente (364 dias) foi reservado como conjunto de teste, enquanto o restante da série foi utilizado como conjunto de treinamento. A divisão respeita a estrutura temporal dos dados, garantindo que o conjunto de teste se inicie após o término do conjunto de treino, sem aleatoriedade, padrão recomendado para séries temporais.

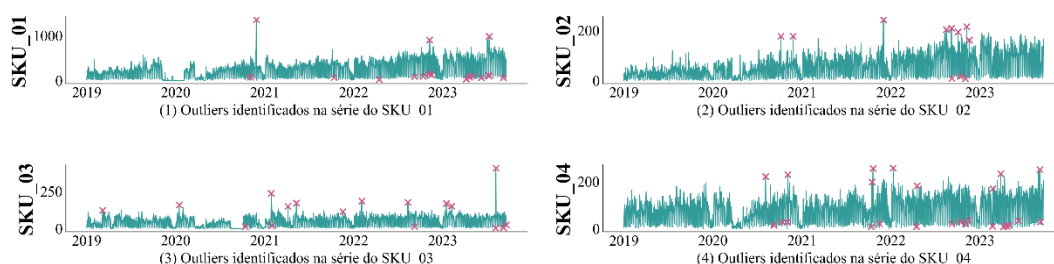
3.3. Feature Engineering

Feature Engineering é a etapa do processo de modelagem onde são usadas técnicas específicas para extração de informações a partir dos dados que podem tornar o processo de aprendizado dos modelos mais fácil e eficiente [Joseph e Tackes 2024]. Para séries temporais, realizar este processo pode contribuir para a captura de padrões sazonais, tendências e anomalias, principalmente no uso de modelos de aprendizado de máquina que não fazem pressuposição de dependência temporal nos dados, como os modelos baseados em árvores utilizados na pesquisa.

Dentre as várias categorias de *features* que podem ser extraídas, as exploradas foram as seguintes:

- **Variáveis de calendário:** Uma característica das séries temporais é que cada observação está associada a uma data – ou data-hora. A partir da data é possível então extrair informações como ano, trimestre, mês, dia, hora etc., limitado à frequência dos dados.
- **Lagged Feature:** Além das variáveis de calendário, é possível utilizar a própria variável objetivo com defasagem como variável de regressão. Por exemplo, pode-se utilizar o valor observado da venda no mesmo dia da semana ou ano anterior como preditor. Neste caso, optou-se por utilizar a defasagem anual, considerando que a série apresenta padrões sazonais recorrentes ao longo dos anos.
- **Feriados e outros eventos:** Feriados trazem efeitos diferentes para cada tipo de negócio, podendo elevar o nível de vendas em alguns segmentos do comércio enquanto diminui em outros. A informação foi adicionada inicialmente em uma única variável, e os feriados considerados foram todos os feriados nacionais brasileiros. Subsequentemente, foram adicionadas 3 novas variáveis que são atrasos e adiantamentos da variável de feriados, uma vez que estas datas podem afetar não somente o dia exato do feriado, mas os dias próximos, representando efeito antecipado ou prolongado dos feriados nas vendas. Também foi adicionada a variável pandemia, compreendendo o período entre 19 de março e 03 de maio de 2025.
- **Outliers:** Detectar observações anômalas é uma etapa essencial em qualquer problema de predição, uma vez que estas anomalias podem influenciar negativamente o desempenho dos modelos, que em muitos casos se traduz em perdas financeiras significativas. Para este fim, foi utilizada uma técnica baseada no modelo Seasonal-ESD, proposto por Hochenbaum et al., que aplica o método Desvio Extremo de Student (ESD) [Rosner 1983] aos resíduos de uma série decomposta sazonalmente [Hochenbaum et al. 2017]. Neste trabalho, foi utilizado o método ESD unilateral tanto para detecção *outliers* nas extremidades inferior e superior separadamente nas séries decompostas usando médias móveis via *seasonal_decompose* do pacote *statsmodels* [Seabold, S., & Perktold, J. 2010], resultado em duas variáveis binárias que indicam quais registros são considerados extremos. Os *outliers* detectados em cada série podem ser vistos na Figura 4.

Figura 4: Outliers detectados via S-ESD



- **Termos de Fourier:** Assim como as variáveis de calendário, termos de Fourier visam modelar a sazonalidade, mas o uso deste recurso é mais vantajoso em séries com sazonalidades de períodos mais longos e mais complexas. Os primeiros termos seno e cosseno são dados por:

$$x_{1,t} = \text{sen}\left(\frac{2\pi t}{s}\right), x_{2,t} = \text{cos}\left(\frac{2\pi t}{s}\right),$$

onde t é a variável de calendário, como mês ou dia da semana, e s é o período sazonal [Hyndman e Athanasopoulos 2021].

A Tabela 2 apresenta todas as *features* que foram extraídas dos dados, bem como os valores que podem assumir em cada observação.

Tabela 2: Features extraídas dos dados

Feature	Tipo	Descrição	Valores
<i>dia</i>	Inteiro	Dia do mês	$x \in \{1, 2, \dots, 31\}$
<i>feriado</i>	Binário	Se o dia é feriado nacional	$x \in \{0, 1\}$
<i>feriado_lag</i>	Binário	Variáveis de feriado com adiantamento/atraso	$x \in \{0, 1\}$
<i>dia_da_semana</i>	Inteiro	Número que representa o dia da semana	$x \in \{1, 2, \dots, 7\}$
<i>semana</i>	Inteiro	Número que representa a semana do ano	$x \in \{1, 2, \dots, 53\}$
<i>mes</i>	Inteiro	Número que representa o mês	$x \in \{1, 2, \dots, 12\}$
<i>outlier_sup</i>	Binário	Se a observação é um outlier superior	$x \in \{0, 1\}$
<i>outlier_inf</i>	Binário	Se a observação é um outlier inferior	$x \in \{0, 1\}$
<i>dia_da_semana_sen</i>	Contínuo	Termo seno de Fourier para ciclo semanal	$x \in [-1, 1]$
<i>dia_da_semana_cos</i>	Contínuo	Termo cosseno de Fourier para ciclo semanal	$x \in [-1, 1]$
<i>mes_sen</i>	Contínuo	Termo seno de Fourier para ciclo mensal	$x \in [-1, 1]$
<i>mes_cos</i>	Contínuo	Termo cosseno de Fourier para ciclo mensal	$x \in [-1, 1]$
<i>y_lag</i>	Discreto	Variável y com defasagem de 1 ano	$x \in \mathbb{Z}_+$
<i>pandemia</i>	Binário	Se a observação ocorreu no período afetado	$x \in \{0, 1\}$

3.4. Seleção de Features

O objetivo da seleção de *features* é reduzir o número de variáveis explicativas usadas para análise e modelagem, mantendo apenas aquelas que mais contribuem para o desempenho dos modelos. Esse processo visa eliminar *features* redundantes ou com baixo poder explicativo, tornando os modelos mais computacionalmente eficientes e interpretáveis.

Neste trabalho utilizamos o método de eliminação recursiva de *features* (RFE) com validação cruzada. O método RFE funciona ajustando inicialmente um modelo com as n *features* e remove do conjunto a *feature* com menor importância para o modelo. Em seguida, um novo modelo é ajustado com o subconjunto das $n - 1$ *features* e uma nova *feature* é retirada. Estes passos são repetidos até que um número mínimo definido de *features* seja atingido. A validação cruzada executa o método RFE em subconjuntos de observações de treino e validação e o conjunto de *features* escolhido é aquele com melhor desempenho médio nos dados de validação. O modelo utilizado no RFE foi o Random Forest, com 4 conjuntos de validação cruzada e mínimo de 5 *features* selecionadas. A Tabela 3 apresenta cada *feature* escolhida para as séries.

Tabela 3: Features selecionadas utilizando o método RFECV

SKU	Features Selecionadas
SKU_01	{dia, feriado, dia_da_semana, semana, mes, outlier_sup, outlier_inf, dia_da_semana (sen), mes (sen, cos), y_lag, pandemia}
SKU_02	{dia, feriado, dia_da_semana, semana, mes, outlier_sup, outlier_inf, feriado_lag (-3, -2, -1, 1, 2, 3), dia_da_semana (sen, cos), mes (sen, cos), y_lag, pandemia}
SKU_03	{dia, feriado, dia_da_semana, semana, outlier_sup, y_lag}
SKU_04	{dia, feriado, dia_da_semana, semana, mes, outlier_sup, outlier_inf, feriado_lag (1, 3), dia_da_semana (sen, cos), mes (sen, cos), y_lag, pandemia}

3.5. Otimização de Hiperparâmetros

Modelos de *machine learning* apresentam diferentes configurações de arquitetura que não podem ser aprendidos com os dados, chamados hiperparâmetros, que são assim chamados para fazer distinção dos parâmetros de um modelo que são estimados a partir dos dados [Chollet 2021] – p. ex., $\hat{\beta}$ de uma regressão linear é um exemplo de parâmetro, enquanto a quantidade de árvores em um modelo Random Forest é um hiperparâmetro.

A escolha da melhor combinação de hiperparâmetros pode se tornar uma tarefa demorada e inviável quando realizada manualmente, especialmente em modelos de maior complexidade ou

com grandes volumes de dados. Para lidar com esse desafio, diversas estratégias de busca têm sido desenvolvidas, como Grid Search, Random Search e Otimização Bayesiana [Ilemobayo et al., 2024]. Neste trabalho, foi adotado o método Grid Search com validação cruzada, que realiza uma busca exaustiva em uma grade pré-definida de combinações de hiperparâmetros. Essa abordagem avalia cada combinação com base na média das perdas obtidas em n partições dos dados, permitindo a seleção do conjunto que minimiza a função de perda. Cada modelo foi treinado com 4 conjuntos de validação cruzada e conjunto de hiperparâmetros apresentados na Tabela 4. Os hiperparâmetros escolhidos para cada modelo e série podem ser vistos na Tabela 5.

Tabela 4: Lista de hiperparâmetros usados no Grid Search

Modelo	Hiperparâmetro	Definição	Valores
Random Forest	<i>n_estimators</i>	Quantidade de árvores na floresta	{100, 200, 300}
	<i>max_depth</i>	Profundidade máxima das árvores	{None, 5, 7, 10}
	<i>min_samples_split</i>	Mínimo de amostras para dividir um nó	{2, 5, 10}
	<i>min_samples_leaf</i>	Mínimo de amostras por folha	{1, 2, 5}
XGBoost	<i>n_estimators</i>	Quantidade de árvores (boosting rounds)	{100, 200, 300}
	<i>max_depth</i>	Profundidade máxima das árvores	{None, 5, 7, 10}
	<i>learning_rate</i>	Taxa de aprendizado de cada nova árvore	{0.01, 0.05, 0.1}
	<i>subsample</i>	% de amostras usadas por árvore	{0.8, 0.9, 1.0}
	<i>colsample_bytree</i>	% de <i>features</i> usadas por árvore	{0.8, 0.9, 1.0}

Tabela 5: Lista de hiperparâmetros escolhidos para cada SKU

Modelo	Hiperparâmetro	SKU_01	SKU_02	SKU_03	SKU_04
Random Forest	<i>n_estimators</i>	200	100	100	300
	<i>max_depth</i>	7	10	7	None
	<i>min_samples_split</i>	10	10	2	2
	<i>min_samples_leaf</i>	1	1	1	1
XGBoost	<i>n_estimators</i>	100	100	200	100
	<i>max_depth</i>	7	7	5	5
	<i>learning_rate</i>	0.1	0.05	0.05	0.1
	<i>subsample</i>	0.9	1.0	1.0	1.0
	<i>colsample_bytree</i>	0.9	0.9	1.0	0.9

3.6. Função de Perda

Nas etapas de seleção de *features* e otimização de hiperparâmetros foi utilizada uma função de perda para comparar as alterações realizadas e escolher a melhor combinação. A métrica adotada neste trabalho foi a Raiz do Erro Quadrático Médio (REQM), definida por:

$$REQM = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2},$$

onde n é o número de pontos previstos, $\hat{y}_i$ e y_i são a previsão e valor real no i -ésimo ponto.

Embora métricas relativas, como o Erro Percentual Absoluto Médio (EPAM), possam ser úteis por expressarem o erro de forma percentual e possibilitar a comparação de resultados entre diferentes séries, sua aplicação não é adequada neste caso devido à presença de valores iguais a zero nos dados reais (y_i), o que resultaria em uma operação indefinida.

4. Resultados

O estudo buscou avaliar a aplicabilidade de modelos baseados em árvores de decisão na previsão de longo prazo de séries temporais de frequência diária. Foram utilizados os modelos

Naive Sazonal, SARIMA e ETS como modelos de base para comparação e avaliação dos resultados obtidos. Os modelos de *machine learning* aplicados foram: Árvore de Decisão, Florestas Aleatórias e XGBoost. Além disso, diversas técnicas de *feature engineering* foram implementadas a fim de tornar os modelos de árvores adequados para o tipo de problema proposto. A Tabela 6 apresenta a comparação entre os modelos utilizando a métrica REQM, com o melhor resultado para cada série destacado.

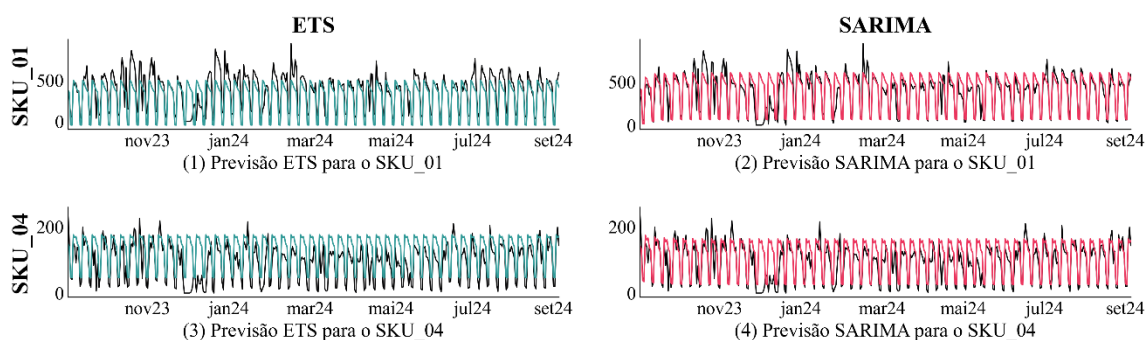
Tabela 6: Comparação dos modelos baseada na métrica REQM

Modelo	SKU_01	SKU_02	SKU_03	SKU_04
Naive Sazonal	215,95	54,20	29,62	63,68
SARIMA	135,29	32,58	23,49	39,88
ETS	134,60	32,46	21,69	38,21
Árvore de Decisão	138,65	35,73	26,12	47,90
Florestas Aleatórias	119,05	26,56	17,60	38,03
XGBoost	112,85	28,16	20,92	38,36

Os métodos de *machine learning* Florestas Aleatórias e XGBoost notadamente foram dominantes na previsão das séries, apresentando resultados que reduziram em média 76,6% do erro comparados ao Naive Sazonal, 15,8% comparados ao SARIMA e 12,1% comparados ao ETS. Dentre eles, o modelo com melhor performance geral foi o Florestas Aleatórias, que obteve melhor resultado em três das quatro séries, apresentando alta capacidade de generalização mesmo sendo um modelo teoricamente mais simples do que o método de *boosting*.

As Figuras 5 e 6 mostram que os modelos estatísticos apresentam desempenho satisfatório nas primeiras semanas do conjunto de teste. No entanto, sua performance diminui significativamente no longo prazo, principalmente pela limitação em captar padrões não lineares presentes nas séries temporais — como sazonalidades mensais e ciclos recorrentes de aumento e queda nas vendas ao longo das semanas. Por outro lado, os modelos baseados em árvores demonstraram maior capacidade de adaptação a esses padrões, especialmente porque foram treinados com covariáveis relevantes para este propósito. Essa vantagem decorre de sua habilidade em modelar relações não lineares entre a variável alvo e as variáveis explicativas. Um exemplo marcante pode ser observado no mês de dezembro, quando houve uma queda acentuada nos volumes de venda que não foi captada pelos modelos clássicos, mas foi parcialmente capturada pelos modelos de árvores. A Figura 7 complementa essa análise ao apresentar a evolução do REQM em janelas móveis de 5 dias: os modelos clássicos exibem um crescimento mais acelerado do erro ao longo do tempo, enquanto os modelos de árvore mantêm um crescimento mais moderado nos SKUs 02 e 03, e uma estabilidade notável nos SKUs 01 e 04 durante todo o horizonte de previsão.

Figura 5: Realizado versus previsto dos modelos Baseline para os SKUs 01 e 04.



Como evidenciado, os modelos de árvores apresentaram bom desempenho na modelagem das oscilações sazonais das séries, o que indica que as etapas de *feature engineering* e seleção de variáveis foram eficazes em fornecer informações relevantes para capturar a sazonalidade. Entre os modelos avaliados, Florestas Aleatórias destacou-se pelo melhor desempenho geral, apresentando também maior estabilidade na evolução do erro ao longo do tempo, conforme ilustrado na Figura 7. Isso sugere que esse modelo possui capacidade para realizar previsões mais confiáveis em horizontes mais longos, especialmente quando há disponibilidade de variáveis explicativas adequadas.

Observou-se ainda uma limitação comum a todos os modelos, que é a dificuldade em prever picos de vendas. Esse comportamento pode estar relacionado à ausência de variáveis exógenas importantes no conjunto de dados, como preços, promoções ou eventos sazonais específicos do negócio, os quais normalmente exercem forte influência nas vendas. A inclusão desses fatores em estudos futuros pode contribuir significativamente para a melhoria das previsões, auxiliando na captura dessa volatilidade.

Figura 6: Realizado versus previsto dos modelos de Árvore para os SKUs 01 e 04.

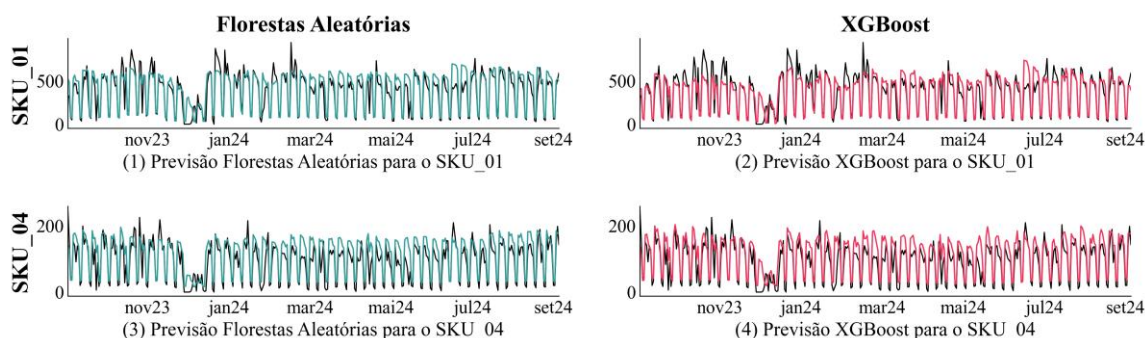
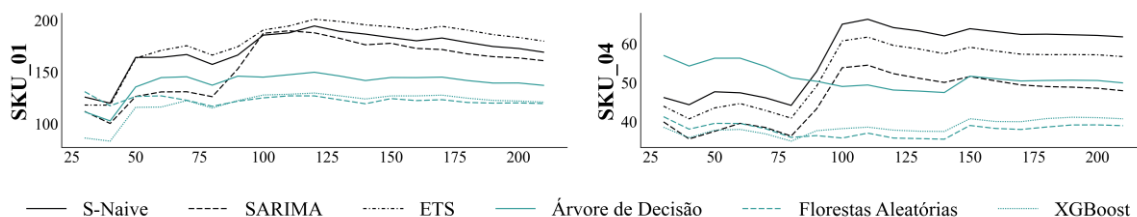


Figura 7: Evolução do REQM através de janelas de 5 dias para os SKUs 01 e 04.



Em síntese, os resultados deste estudo evidenciam que modelos baseados em árvores, especialmente o Florestas Aleatórias, oferecem melhor desempenho e maior robustez na previsão de séries temporais de vendas em horizontes de longo prazo, sobretudo quando lhes é fornecido um conjunto de variáveis explicativas adequadas. A comparação com modelos estatísticos clássicos demonstrou que estes, embora sejam eficientes em curtos prazos, apresentam maiores erros à medida que o horizonte de previsão aumenta, em grande parte pela limitação em captar padrões não lineares e sazonalidades mais complexas. No entanto, tanto os modelos clássicos quanto os de *machine learning* tiveram dificuldades em prever picos de vendas, sugerindo a necessidade de incorporar novas variáveis internas ao negócio. Assim, o estudo reforça a importância do *feature engineering* na modelagem de séries temporais, um processo que embora demande mais esforço em termos de construção, entendimento e identificação das variáveis importantes, pode trazer mais confiabilidade nas previsões de longo prazo.

Referências

- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
- Chen, T., e Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *ArXiv*.
- Chollet, F. (2021) *Deep Learning with Python*, Second Edition, Manning, Shelter Island, NY.
- Hochenbaum, J., Vallis, O., e Kejariwal, A. (2017). Automatic Anomaly Detection in the Cloud Via Statistical Learning. *ArXiv*.
- Hyndman, R.J., e Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. otexts.com/fpp3. Acessado em 01/04/2025.
- Ikughur, A. J., Uba, T., e Ogunmola, A. O. (2015). Application of residual analysis in time series model selection. *Journal of Statistical and Econometric Methods*, 4(4), 41-53.
- Ilemobayo, J., Durodola, O., Alade, O., Awotunde, O., Adewumi, T., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., Odezuligbo, I., Edu, O. e Ifeanyi, Ark. (2024). Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*.
- Izbicki, R. e Santos, T. M. (2020). *Aprendizado de máquina: uma abordagem estatística*. 1ª edição. <https://rafaelizbicki.com/ame/>. Acessado em 01/04/2025.
- Joseph, M. e Tackes, J. (2024). *Modern Time Series Forecasting with Python*. 2ª Edição. Packt Publishing, Birmingham, Inglaterra.
- Lye, J. e Hirschberg, J. (2020). *Time Series Plots: A Compendium of time series plots*.
- Ng, K. W., Mubang, F., Hall, L. O., Skvoretz, J., e Iamnitchi, A. (2023). Experimental evaluation of baselines for forecasting social media timeseries. *EPJ data science*, 12(1), 8.
- Rosner, B. (1983). Percentage Points for a Generalized ESD Many-Outlier Procedure. *Technometrics*, 25(2), 165–172.
- Seabold, S. e Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, 57, 61.
- Spiliotis, E. (2022). Decision Trees for Time-Series Forecasting. *Foresight: The International Journal of Applied Forecasting*.
- Svetunkov, I. (2023). *Forecasting and Analytics with the Augmented Dynamic Adaptive Model (ADAM)* (1st ed.). Chapman and Hall/CRC.
- Wei, W. W. S. (2006). *Time series analysis: Univariate and multivariate methods*. Pearson Addison Wesley.