v7.n11.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

DESENVOLVIMENTO DE UM PIPELINE DE DADOS INTEGRADO À PREVISÃO DE DEMANDA NO VAREJO DE COMÉRCIO ELETRÔNICO

DEVELOPMENT OF A DATA PIPELINE INTEGRATED WITH DEMAND FORECASTING IN E-COMMERCE RETAIL

DESARROLLO DE UN PIPELINE DE DATOS INTEGRADO A LA PREDICCIÓN DE DEMANDA EN EL COMERCIO MINORISTA ELECTRÓNICO

Yasmim Fernandes de Paula[1], Felipe Diniz Dallilo[2], Fabiana Florian[3]

e7119455
https://doi.org/10.47820/recima21.v7i11.9455

PUBLICADO: 11/2026

RESUMO

Este trabalho desenvolve um *pipeline* de dados integrado à previsão de demanda no varejo de comércio eletrônico e avalia se modelos de aprendizado de máquina superam um *baseline* simples. Utilizou-se o Brazilian E-Commerce Public Dataset by Olist, organizado em camadas *bronze*, *silver* e *gold* no PostgreSQL, com a demanda definida como o número de itens vendidos em pedidos entregues, por categoria e mês. Regressão Linear, *Random Forest* e *Gradient Boosting* foram comparados ao *Naive* (demanda do mês anterior) por MAE, RMSE e WAPE, em dois cenários: sem e com agrupamento Pareto das categorias. No teste principal (junho a agosto de 2018), o *Naive* obteve o menor erro, com WAPE de 16,11% sem Pareto e 12,92% com Pareto, contra 22,61% a 25,94% e 19,55% a 20,11% dos demais modelos. Análises complementares (sensibilidade ao conjunto de variáveis, validação *walk-forward* com janela expansível e melhorias de modelagem) mantiveram o resultado: o *Naive* teve WAPE igual ou menor em todos os cenários e janelas avaliados, com empate prático em um único recorte. O comportamento é compatível com mudança de regime da demanda, histórico curto e baixa previsibilidade da variação mensal, hipóteses que não foram testadas formalmente. Conclui-se que, neste conjunto de dados, a maior complexidade dos algoritmos não trouxe ganho de precisão e que *baselines* devem ser referência obrigatória na seleção de modelos preditivos.

PALAVRAS-CHAVE: Engenharia de dados. Previsão de demanda. Aprendizado de máquina. PostgreSQL. Varejo. Olist.

ABSTRACT

This study develops a data pipeline integrated with demand forecasting in e-commerce retail and assesses whether machine learning models outperform a simple baseline. The Brazilian E-Commerce Public Dataset by Olist was organized into bronze, silver and gold layers in PostgreSQL, with demand defined as the number of items sold in delivered orders, by category and month. Linear Regression, Random Forest and Gradient Boosting were compared with a Naive baseline (previous month's demand) using MAE, RMSE and WAPE, under two scenarios: without and with Pareto grouping of categories. In the main test (June to August 2018), the Naive baseline had the lowest error, with WAPE of 16.11% without Pareto and 12.92% with Pareto, against 22.61% to 25.94% and 19.55% to 20.11% for the other models. Complementary analyses (sensitivity to the feature set, expanding-window walk-forward validation and modeling improvements) did not change this result: the Naive baseline had equal or lower WAPE in all

[1] Graduanda em Engenharia de Computação pela Universidade de Araraquara – UNIARA.
[2] Doutor em Engenharia de Software pela USP. Docente na UNIARA.
[3] Doutora em Alimentos e Nutrição. Docente e coordenadora de TCC nos cursos de Sistemas de Informação e Engenharia de Computação na UNIARA.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

scenarios and windows evaluated, with a practical tie in a single case. This behavior is consistent with a demand regime change, a short history and low predictability of monthly variation, hypotheses that were not formally tested. It is concluded that, on this dataset, greater algorithmic complexity brought no gain in accuracy and that baselines should be a mandatory reference when selecting predictive models.

KEYWORDS: *Data engineering. Demand forecasting. Machine learning. PostgreSQL. Retail. Olist.*

RESUMEN

Este trabajo desarrolla un pipeline de datos integrado a la predicción de demanda en el comercio minorista electrónico y evalúa si los modelos de aprendizaje automático superan a un baseline simple. Se utilizó el Brazilian E-Commerce Public Dataset by Olist, organizado en capas bronze, silver y gold en PostgreSQL, con la demanda definida como el número de ítems vendidos en pedidos entregados, por categoría y mes. Regresión Lineal, Random Forest y Gradient Boosting se compararon con el Naive (demanda del mes anterior) mediante MAE, RMSE y WAPE, en dos escenarios: sin y con agrupación Pareto de las categorías. En la prueba principal (junio a agosto de 2018), el Naive obtuvo el menor error, con WAPE de 16,11% sin Pareto y 12,92% con Pareto, frente a 22,61% a 25,94% y 19,55% a 20,11% de los demás modelos. Los análisis complementarios (sensibilidad al conjunto de variables, validación walk-forward con ventana expansiva y mejoras de modelado) no modificaron este resultado: el Naive tuvo WAPE igual o menor en todos los escenarios y ventanas evaluados, con un empate práctico en un solo caso. El comportamiento es compatible con un cambio de régimen de la demanda, un historial corto y la baja previsibilidad de la variación mensual, hipótesis que no se probaron formalmente. Se concluye que, en este conjunto de datos, la mayor complejidad de los algoritmos no aportó mejora de precisión y que los baselines deben ser referencia obligatoria en la selección de modelos predictivos.

PALABRAS CLAVE: *Ingeniería de datos. Predicción de demanda. Aprendizaje automático. PostgreSQL. Comercio minorista. Olist.*

1. INTRODUÇÃO

A disponibilidade de grandes volumes de dados e a ampliação da capacidade de processamento têm favorecido a decisão baseada em dados (PROVOST; FAWCETT, 2013). No varejo, informações de pedidos, produtos e vendas podem ser usadas para compreender o comportamento da demanda e apoiar o planejamento operacional. Uma dessas aplicações é a previsão de demanda, isto é, o uso de dados históricos para estimar o consumo futuro, com impacto em compras, abastecimento e distribuição. Contudo, a aplicação de técnicas preditivas depende de dados organizados, consistentes e disponíveis em uma estrutura adequada para análise.

A escolha do método de previsão costuma ser tratada como uma questão de sofisticação algorítmica: espera-se que modelos de aprendizado de máquina, frequentemente propostos como alternativa aos métodos estatísticos, superem métodos simples. A evidência empírica nem sempre confirma essa expectativa: em um estudo com 1.045 séries mensais, Makridakis, Spiliotis

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

e Assimakopoulos (2018) verificaram que métodos de aprendizado de máquina foram superados em acurácia por métodos estatísticos tradicionais. A literatura de previsão também recomenda comparar qualquer modelo com referências simples antes de adotá-lo (HYNDMAN; ATHANASOPOULOS, 2021). O problema de pesquisa deste trabalho é: como um *pipeline* de dados integrado à previsão de demanda pode organizar informações de vendas e apoiar a avaliação de estratégias preditivas em uma operação de varejo de comércio eletrônico, e se os modelos de aprendizado de máquina superam um *baseline* simples nesse cenário?

O objetivo geral é desenvolver e avaliar um *pipeline* de dados integrado à previsão de demanda, utilizando dados públicos de comércio eletrônico da Olist. Os objetivos específicos são: (i) estruturar os dados em camadas de tratamento (*bronze*, *silver* e *gold*) em um banco de dados relacional; (ii) construir uma série temporal mensal de demanda por categoria de produto; (iii) comparar Regressão Linear, *Random Forest* e *Gradient Boosting* com um *baseline Naive*; e (iv) verificar se o resultado se mantém quando mudam as variáveis, o esquema de validação e a configuração dos modelos.

A justificativa está na necessidade de transformar dados operacionais em informação estruturada para apoiar decisões e de verificar, com evidências, se uma maior complexidade do modelo traz melhores previsões para os dados disponíveis. O projeto foi concebido originalmente para o varejo de materiais de construção; como não havia dados reais desse segmento, adotou-se o conjunto público da Olist (OLIST, 2018), delimitando o contexto ao comércio eletrônico. O ineditismo do trabalho está em combinar, em um fluxo reprodutível, a estruturação dos dados em camadas e uma avaliação de robustez do desempenho de modelos frente a um *baseline* simples (sensibilidade às variáveis, validação *walk-forward* e melhorias de modelagem), aplicadas à demanda mensal por categoria do conjunto da Olist. Trata-se de pesquisa aplicada, quantitativa e experimental.

Dois aspectos orientam o desenho do estudo. O primeiro é de engenharia de dados: dados transacionais brutos, como pedidos e itens vendidos, não estão prontos para modelagem e precisam ser organizados em uma estrutura reprodutível, que preserve a fonte original e torne explícitas as regras de tratamento. O segundo é de avaliação: como a série disponível é curta e o resultado de um único teste pode ser acidental, a comparação entre modelos foi repetida sob diferentes variáveis, esquemas de validação e configurações, em vez de depender de uma única medida de desempenho.

O artigo está organizado em quatro seções, além desta introdução: o referencial teórico, que apresenta previsão de demanda, engenharia de dados e os modelos e métricas utilizados;

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

a metodologia, que descreve o *pipeline*, as variáveis, os cenários e as análises complementares; os resultados e a discussão; e as considerações finais.

2. REFERENCIAL TEÓRICO

2.1. Previsão de demanda e *baselines*

A previsão de demanda é tratada como um problema de série temporal, e os métodos devem ser escolhidos e avaliados conforme tendência, sazonalidade, variabilidade e horizonte de previsão (HYNDMAN; ATHANASOPOULOS, 2021). As abordagens vão de métodos simples, como repetir o último valor observado (*Naive* ou de persistência), a modelos estatísticos e de aprendizado de máquina. O *Naive* é a referência mínima de desempenho e permite verificar se um modelo mais complexo traz ganho real. Séries curtas ou sujeitas a mudanças de comportamento limitam a capacidade de generalização de modelos complexos (HYNDMAN; ATHANASOPOULOS, 2021).

Para avaliar modelos sem usar informações futuras, a validação *walk-forward* (avaliação com origem móvel) treina o modelo apenas com observações anteriores a cada período avaliado e avança a origem da previsão no tempo, o que reduz a dependência de uma única janela de teste (HYNDMAN; ATHANASOPOULOS, 2021).

2.2. Gestão de estoques e agrupamento por volume

A gestão de estoques busca equilibrar a disponibilidade de produtos e os custos de mantê-los, evitando tanto rupturas quanto excessos. Parâmetros como o ponto de pedido, o estoque de segurança e o nível de serviço dependem, em maior ou menor grau, de estimativas da demanda futura. No varejo, essas estimativas apoiam o planejamento de compras, o abastecimento e a distribuição entre pontos de venda ou centros de distribuição; quanto mais precisa a estimativa, menor tende a ser o estoque de segurança necessário para compensar a incerteza, o que justifica investir na qualidade dos modelos preditivos (CHOPRA; MEINDL, 2016).

Em portfólios com itens de volumes muito distintos, a classificação ABC, fundamentada no princípio de Pareto, reconhece que uma pequena parcela dos itens (cerca de 20%) responde pela maior parte do volume movimentado (cerca de 80%). Os itens de maior volume recebem atenção individualizada, enquanto os de menor volume costumam ser tratados de forma agregada, pois o esforço analítico dedicado a cada um isoladamente não se justifica diante de sua contribuição ao total (CHOPRA; MEINDL, 2016). Esse raciocínio fundamenta o cenário Com Pareto adotado neste trabalho.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

2.3. Engenharia de dados e arquitetura em camadas

A engenharia de dados reúne práticas de coleta, armazenamento, transformação e disponibilização de dados de forma confiável para uso analítico, e os *pipelines* organizam o fluxo entre as etapas, reduzindo tratamentos manuais e favorecendo a reprodutibilidade (REIS; HOUSLEY, 2023). Confiabilidade, escalabilidade e manutenibilidade são propriedades centrais de sistemas de dados (KLEPPMANN, 2017). A arquitetura medalhão separa os dados em camadas progressivas: *bronze* (dados brutos), *silver* (dados tratados e padronizados) e *gold* (dados consolidados para análise) (DATABRICKS, 2023). O uso de SQL, *views* e tabelas intermediárias permite executar o tratamento próximo ao armazenamento, deixando à linguagem Python as etapas de análise e modelagem. Para Silva e Cassiano (2024), o encadeamento correto das etapas do fluxo de dados é determinante para transformar registros brutos em suporte à decisão. Trabalhos aplicados mostram a viabilidade de combinar ETL e *dashboards* como apoio à decisão em diferentes domínios, como o de Brandão (2025), que desenvolveu um painel analítico para dados de seguros de automóveis a partir de um *pipeline* de ETL integrado a um banco de dados.

2.4. Modelos e métricas

Foram utilizados três algoritmos supervisionados de regressão (GÉRON, 2022; PEDREGOSA et al., 2011). A Regressão Linear, de fácil interpretação e baixo custo computacional, estima a variável-alvo como combinação linear das variáveis explicativas, mas é limitada para capturar relações não lineares. O *Random Forest* combina árvores de decisão treinadas em subamostras aleatórias e agrega suas previsões por média, o que reduz o risco de sobreajuste (*overfitting*) de uma árvore isolada. O *Gradient Boosting* constrói árvores em sequência, cada uma corrigindo os erros residuais das anteriores, o que costuma dar maior capacidade de ajuste aos dados de treino, ao custo de maior sensibilidade a ruído e de maior tempo de treinamento. Como referência, o *Naive Lag-1* usa a demanda do período anterior como previsão do seguinte, ou seja, prevê variação nula. Como métricas, usaram-se o MAE (erro absoluto médio), o RMSE (raiz do erro quadrático médio) e o WAPE, razão entre a soma dos erros absolutos e a soma dos valores reais (KOLASSA; SCHÜTZ, 2007). O WAPE foi a métrica principal porque, diferentemente do MAPE, permanece definido quando a demanda real é zero, situação frequente nesta base, em que parte das combinações de categoria e mês tem demanda nula. O MAE expressa o erro médio na mesma unidade da demanda (itens), e o RMSE, por elevar os erros ao quadrado, penaliza de forma mais acentuada os desvios de maior magnitude (HYNDMAN; ATHANASOPOULOS, 2021).

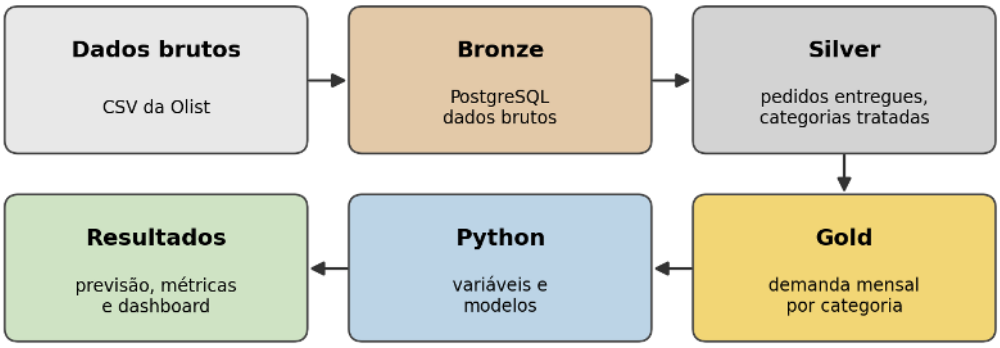
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

3. METODOLOGIA

3.1. Dados e *pipeline*

Utilizaram-se quatro tabelas do Brazilian E-Commerce Public Dataset by Olist (OLIST, 2018): pedidos (99.441 registros), itens de pedido (112.650), produtos (32.951) e tradução de categorias. Os dados foram carregados em um banco PostgreSQL, por script Python com Pandas (MCKINNEY, 2018) e SQLAlchemy (SQLALCHEMY, 2024). O fluxo geral é mostrado na Figura 1: os dados brutos são armazenados na camada *bronze*, tratados e padronizados na camada *silver* e consolidados para uso analítico na camada *gold*; em seguida, o ambiente Python prepara as variáveis e treina os modelos, e as previsões alimentam o processo de apoio à decisão sobre reposição de estoque e planejamento de compras. O PostgreSQL concentra o armazenamento e parte do processamento, e o Python fica responsável por análise e modelagem.

Figura 1. Fluxo do *pipeline* de dados desenvolvido. Fonte: elaborado pelos autores (2026)



Na camada *bronze*, as tabelas foram criadas por instruções DDL, preservando os tipos originais dos arquivos CSV e sem regras de negócio, para que a camada representasse fielmente a fonte. A carga foi automatizada por script Python, sem importação manual, e a contagem de linhas de cada arquivo coincidiu com a das tabelas carregadas: 99.441 pedidos, 112.650 itens de pedido, 32.951 produtos e 71 categorias traduzidas. Na camada *silver*, por consultas SQL, mantiveram-se apenas os pedidos com situação “entregue” (97,02% dos pedidos), pois pedidos cancelados ou não entregues não representam demanda efetivamente atendida, e os produtos sem categoria foram agrupados em “categoria_desconhecida” por meio da função COALESCE, evitando a perda dessas observações na agregação. Na camada *gold*, construiu-se a tabela de

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

demanda mensal por categoria, na qual a demanda é o número de itens vendidos em pedidos entregues, por categoria de produto e mês da compra. Gerou-se um *grid* completo de combinações entre categoria e mês: a sequência de meses, criada com a função generate_series do PostgreSQL, foi cruzada com a lista de categorias e unida à demanda observada por LEFT JOIN, com zero nas combinações sem venda. Sem esse cuidado, os meses sem venda desapareceriam da agregação, e o modelo não teria como aprender que demanda igual a zero é um valor real. O *grid* cobre 74 categorias por 24 meses (setembro de 2016 a agosto de 2018), o que resulta em 1.776 linhas e 110.197 itens. A soma da demanda na camada *gold* coincidiu com a contagem de itens da camada *silver*; a agregação, portanto, não duplicou nem perdeu registros. A Tabela 1 resume o tamanho das bases em cada etapa, desde a camada *gold* até os conjuntos de treino e teste, e o percentual de observações com demanda zero.

Tabela 1. Bases de dados utilizadas, por etapa de preparação. Fonte: elaborado pelos autores (2026)

Base	Observações	Período	Demanda zero (%)
Camada *gold* completa	1.776	set/2016 a ago/2018	28,32
Após remover os meses iniciais	1.480	jan/2017 a ago/2018	16,15
Base de modelagem (com variáveis defasadas)	1.258	abr/2017 a ago/2018	13,04
Treino, cenário Sem Pareto	1.036	abr/2017 a mai/2018	–
Teste, cenário Sem Pareto	222	jun a ago/2018	–
Treino / teste, cenário Com Pareto	322 / 69	abr/2017 a mai/2018 / jun a ago/2018	–

Nota: "–" indica que o percentual não foi calculado separadamente para o subconjunto.

3.2. Variáveis e partição temporal

As variáveis foram: ano (*year*); mês, em codificação cíclica por seno e cosseno, para que dezembro e janeiro não sejam tratados como meses distantes; indicador do mês de novembro, associado à *Black Friday*; demanda do mês anterior (*lag-1*) e de dois meses antes (*lag-2*); e média móvel dos três meses anteriores, calculada apenas com meses passados, o que evita o uso da própria demanda a prever. O Quadro 1 mostra o trecho da função que gera essas variáveis, e a Tabela 2 as descreve. A categoria do produto entrou por *One-Hot Encoding*, ajustado apenas com os dados de treino. Removeram-se os meses de setembro a dezembro de 2016 (fase inicial de operação da plataforma), restando 1.480 observações, e as 222 linhas sem histórico para as variáveis defasadas (os três primeiros meses de cada série), restando 1.258

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

observações, das quais 13,04% têm demanda zero. A partição foi temporal, e não aleatória: treino de abril de 2017 a maio de 2018 (1.036 observações) e teste de junho a agosto de 2018 (222 observações). Embaralhar os registros antes da divisão permitiria ao modelo acessar, no treino, informações posteriores ao período a prever, o que invalidaria a avaliação. O horizonte de três meses foi adotado porque o conjunto cobre um período curto, e um teste maior reduziria excessivamente o treino.

Tabela 2. Variáveis utilizadas pelos modelos

Variável	Descrição	Tipo
year	Ano da observação; funciona como indicador de tendência	Numérica
month_sin, month_cos	Mês do ano em codificação cíclica (seno e cosseno)	Numérica
is_black_friday_month	Indicador do mês de novembro	Binária
demand_lag_1	Demanda da categoria no mês anterior	Numérica
demand_lag_2	Demanda da categoria dois meses antes	Numérica
rolling_mean_3	Média da demanda dos três meses anteriores	Numérica
Categoria do produto	Codificação *One-Hot*, ajustada apenas no treino	Categórica

Fonte: elaborado pelos autores (2026).

Quadro 1. Trecho do código Python de engenharia de atributos (arquivo build_features.py).

```
df["month_sin"] = np.sin(2 * np.pi * df["month"] / 12)
df["month_cos"] = np.cos(2 * np.pi * df["month"] / 12)
df["is_black_friday_month"] = (df["month"] == 11).astype(int)
df["demand_lag_1"] = grouped.shift(1)
df["demand_lag_2"] = grouped.shift(2)
df["rolling_mean_3"] = grouped.transform(lambda s: s.shift(1).rolling(window=3).mean())
```

Fonte: elaborado pelos autores (2026).

3.3. Modelos, cenários e métricas

Compararam-se Regressão Linear, *Random Forest* (200 árvores) e *Gradient Boosting* (100 árvores, taxa de aprendizado 0,1, profundidade 3), com hiperparâmetros padrão da biblioteca scikit-learn (PEDREGOSA et al., 2011) e semente aleatória 42, além do *Naive Lag-1*, que prevê a demanda do mês anterior. As previsões foram limitadas a valores não negativos. Avaliaram-se dois cenários: Sem Pareto, com as 74 categorias; e Com Pareto, em que as categorias de cauda longa foram agrupadas em “outras”, mantendo-se as categorias cujo volume

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

acumulado anterior não ultrapassa 90% no período de janeiro de 2017 a maio de 2018 (22 categorias mais “outras”). Usaram-se MAE, RMSE e WAPE, definidos nas expressões a seguir, em que y_i é a demanda real, $\hat{y}_i$ a prevista e n o número de observações. O erro é definido como real − previsto; valores negativos indicam superestimação.

$$MAE = (1/n) \cdot \Sigma |y_i - \hat{y}_i|$$
$$RMSE = \sqrt{[\ (1/n) \cdot \Sigma (y_i - \hat{y}_i)^2\ ]}$$
$$WAPE = (\ \Sigma |y_i - \hat{y}_i| / \Sigma y_i\ ) \times 100$$
$$\textit{Naive Lag-1}: \hat{y}_t = y_{t-1}$$

Os resultados foram também consolidados em um *dashboard* no Power BI, alimentado pelos arquivos exportados do Python com as previsões e as métricas de cada combinação de modelo e cenário. O painel tem cinco páginas (visão geral, comparação dos modelos, análise de erro, análise por categoria e conclusão) e usa medidas em linguagem DAX para fixar os indicadores do modelo operacional, de modo que filtros aplicados a outras visualizações não os alterem sem intenção. A visualização é a etapa final do *pipeline*: transforma as saídas dos experimentos em informação de leitura direta.

3.4. Análises complementares

Quatro análises verificaram a robustez do resultado: (i) exploração dos dados e correlação entre variáveis, com as linhas de treino; (ii) sensibilidade a cinco configurações de variáveis (original, sem ano, só *lag-1* com ano, só *lag-1* sem ano e *lag-1* puro), no mesmo particionamento; (iii) validação *walk-forward* com janela expansível, de maio de 2017 a agosto de 2018 (16 passos), em que o codificador, o modelo e, no cenário Com Pareto, o corte de 90% são recalculados em cada passo apenas com dados anteriores ao mês-alvo, apresentando-se também os passos com pelo menos seis meses de treino (11 passos); e (iv) quatro melhorias em sequência no *walk-forward* Sem Pareto: remoção das variáveis redundantes e do ano (B), alvo em variação, isto é, demanda menos a do mês anterior (C), mais árvores, com 1.000 no *Random Forest* e 500 no *Gradient Boosting* (D), e hiperparâmetros regularizados (E), definidos antes da avaliação, para não usar os meses avaliados.

Para evitar vazamento de informação do período avaliado para o treino, adotaram-se os seguintes cuidados: a partição é sempre temporal; as variáveis defasadas e a média móvel usam apenas meses anteriores ao mês previsto; o *One-Hot Encoding* é ajustado somente com os dados de treino; no cenário Com Pareto, o corte de 90% é calculado apenas com meses anteriores ao teste (no *walk-forward*, o agrupamento é refeito em cada passo, com 21 a 23 categorias mantidas conforme o passo); e os hiperparâmetros do degrau E não foram ajustados

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

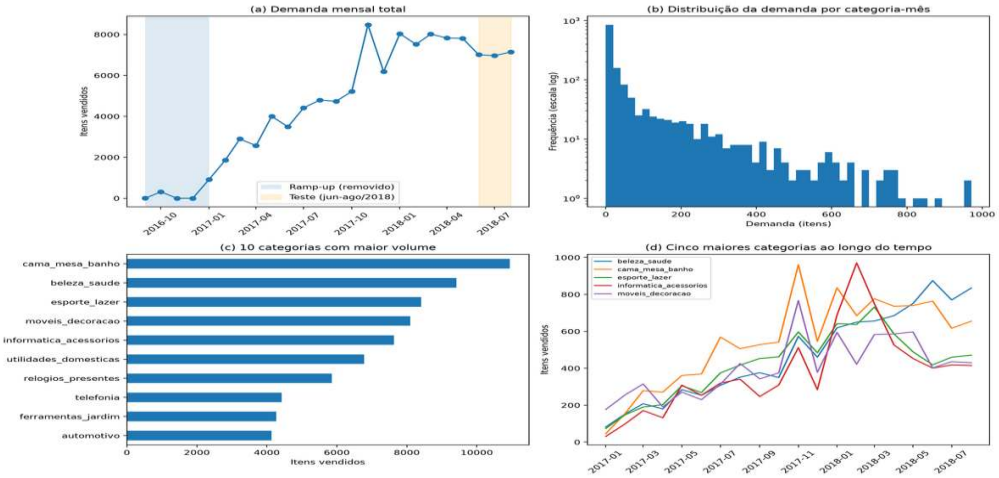
sobre os meses avaliados. Os modelos de árvores usam semente aleatória fixa, e as versões das bibliotecas foram registradas no repositório do projeto, que contém o código completo e está disponível em: https://github.com/rexyasmim/tcc-previsao-demanda.

4. RESULTADOS E DISCUSSÃO

4.1. Exploração dos dados

A demanda mensal total cresceu de 913 itens em janeiro de 2017 para um pico de 8.475 em novembro, caiu para 6.187 em dezembro e se estabilizou entre 7.000 e 8.000 itens em 2018, ficando em torno de 7.000 nos meses de teste, abaixo dos 7.810 de maio de 2018 (Figura 2a). A distribuição por categoria e mês é muito assimétrica (Figura 2b): a mediana é de 13 itens e o máximo, de 971. Os zeros representam 28,32% da camada *gold* completa, 16,15% após a remoção dos meses iniciais e 13,04% da base de modelagem. As dez maiores categorias respondem por 63,67% das vendas (Figura 2c), e são necessárias 22 para atingir 90% do volume até maio de 2018.

Figura 2. Análise exploratória: (a) demanda mensal total; (b) distribuição da demanda por categoria e mês; (c) dez categorias de maior volume; (d) cinco maiores categorias ao longo do tempo. Fonte: elaborado pelos autores (2026).



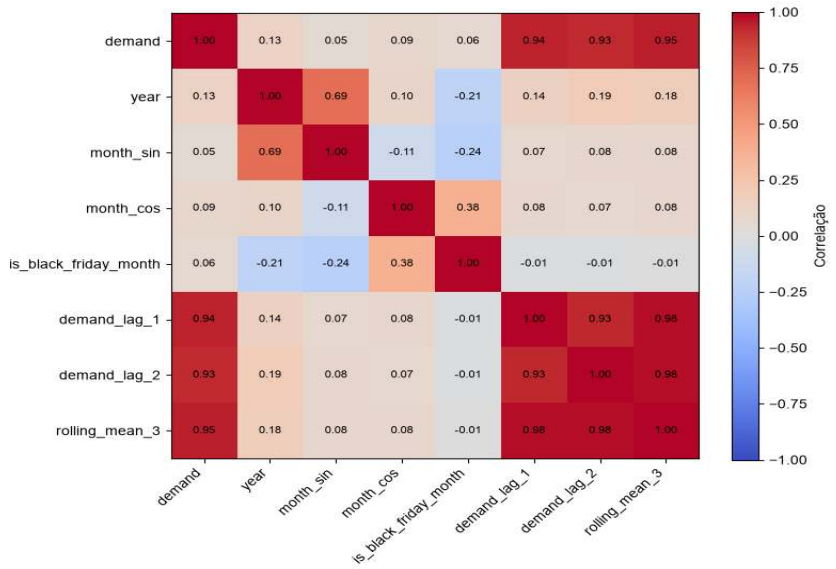
As cinco maiores categorias não se movem em conjunto (Figura 2d). Em 2018, beleza_saude subiu de 619 itens em janeiro para 875 em junho, enquanto informatica_acessorios caiu de 971 em fevereiro para 415 em agosto e esporte_lazer, de 732 em março para 471 em agosto. O platô do total esconde, portanto, trajetórias distintas por

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

categoria. A demanda média por categoria em novembro (114,5 itens) é a maior entre os meses, mas, como novembro aparece uma única vez na série, os dados não permitem separar sazonalidade de crescimento da plataforma.

As três variáveis de histórico têm correlação de 0,94 (*lag-1*), 0,93 (*lag-2*) e 0,95 (média móvel) com a demanda e, entre si, de 0,93 a 0,98, o que indica redundância (Figura 3). O maior valor, 0,98, ocorre entre *lag-2* e a média móvel; o ano correlaciona-se 0,69 com month_sin, efeito da janela de treino, que não cobre os meses do ano de forma uniforme, e as variáveis de calendário têm correlação baixa com a demanda (0,05 a 0,13). A demanda também dá sinais de mudança de regime: a variação média mensal por categoria foi de +4,74 itens no treino e de −3,01 no teste, e o total caiu de 7.810 para 7.010 itens entre maio e junho de 2018 (−10,24%). Um modelo que aprendeu o crescimento do treino tende a superestimar o período seguinte.

Figura 3. Correlação entre as variáveis e a demanda nas linhas de treino. Fonte: elaborado pelos autores (2026)



Esses achados têm duas consequências para a modelagem. Primeiro, a presença de zeros (13,04% da base de modelagem) inviabiliza o uso do MAPE, que ficaria indefinido nessas observações, e justifica o WAPE como métrica principal. Segundo, a forte assimetria e a diferença de volume entre categorias fazem com que o erro absoluto se concentre nas categorias maiores, enquanto o erro relativo é mais alto nas menores, o que exige olhar as duas medidas em conjunto.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.2. Teste principal e análise dos erros

A Tabela 3 mostra que o *Naive* obteve o menor MAE, RMSE e WAPE nos dois cenários. Sem Pareto, o WAPE foi de 16,11% para o *Naive* e de 22,61%, 24,76% e 25,94% para Regressão Linear, *Random Forest* e *Gradient Boosting*. Com Pareto, o *Naive* atingiu 12,92%, contra 19,85%, 19,55% e 20,11%. O WAPE menor no cenário Com Pareto decorre da agregação, que reduz o teste de 222 para 69 observações; por isso, o MAE e o RMSE não são comparáveis entre os cenários. O *Naive* Sem Pareto foi escolhido como modelo operacional, por ter mais observações de teste e evitar a categoria heterogênea "outras", cuja interpretação é menos direta. Os modelos de aprendizado de máquina mantiveram previsões sistematicamente acima da demanda real no período de teste, comportamento compatível com a hipótese de que aprenderam a tendência de crescimento do treino e a extrapolaram para meses em que a demanda se estabilizou. O *Naive*, ao contrário, reajusta-se imediatamente ao nível do mês anterior, o que reduz o erro acumulado em horizontes curtos. A ausência de variáveis exógenas, como preço, promoções ou investimento em marketing, somada ao histórico curto, também pode contribuir para a vantagem do *Naive*, porque nenhuma variável antecedente sinaliza a mudança de comportamento.

Tabela 3. Desempenho no teste principal (junho a agosto de 2018). Fonte: elaborado pelos autores (2026)

Cenário	Modelo	MAE	RMSE	WAPE (%)
Sem Pareto	*Naive*	**15,32**	**30,43**	**16,11**
Sem Pareto	Regressão Linear	21,50	35,52	22,61
Sem Pareto	*Random Forest*	23,55	45,06	24,76
Sem Pareto	*Gradient Boosting*	24,68	50,94	25,94
Com Pareto	*Naive*	**39,54**	**55,27**	**12,92**
Com Pareto	Regressão Linear	60,74	73,58	19,85
Com Pareto	*Random Forest*	59,83	78,13	19,55
Com Pareto	*Gradient Boosting*	61,53	83,47	20,11

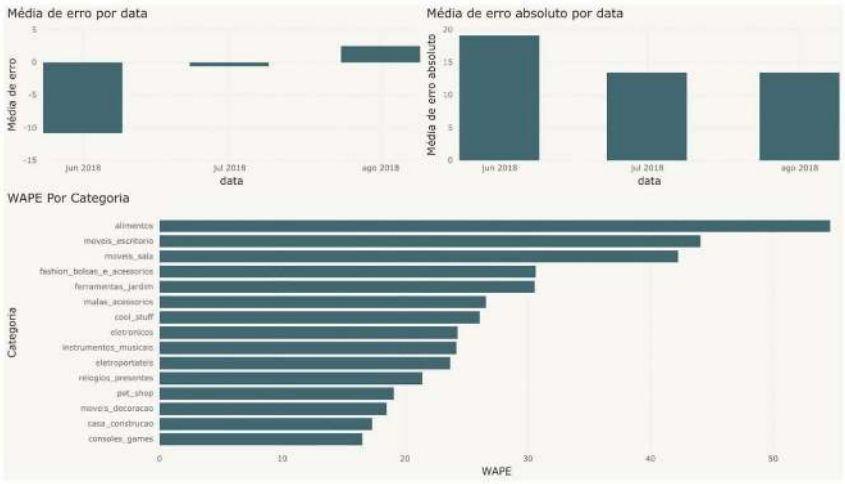
Nota: os valores do *Gradient Boosting* podem variar até cerca de 0,5 ponto percentual com a semente aleatória (utilizada: 42).

O erro médio do *Naive* Sem Pareto foi de −10,81 itens em junho, −0,64 em julho e +2,42 em agosto (Figura 4): superestimou no mês da queda e se ajustou depois. Em junho, a Regressão Linear teve erro médio de −16,78. As categorias de menor volume tiveram WAPE elevado, com alimentos no pior caso (54,68%), enquanto as de maior volume tiveram maior erro absoluto, o que mostra a importância de métricas complementares. Houve ainda 24 observações com demanda real zero no teste, em que o MAE foi de 0,46, contra 17,13 nas observações com demanda positiva. Os maiores erros absolutos médios ficaram em relogios_presentes (100,67),

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

beleza_saude (98,00), moveis_decoracao (78,33), cama_mesa_banho (69,67) e ferramentas_jardim (50,67).

Figura 4. Erro médio e erro absoluto médio do *Naive* por mês de teste e WAPE das 15 categorias com maior WAPE entre as que somam mais de 100 itens no teste (Sem Pareto).



Fonte: elaborado pelos autores (2026).

4.3. Sensibilidade ao conjunto de variáveis

A Tabela 4 compara cinco configurações de variáveis no mesmo teste, o que dá à análise caráter de sensibilidade, e não de seleção de modelo. Retirar o ano reduziu o WAPE da Regressão Linear de 22,61% para 18,30% e seu erro médio de −13,99 para −1,85 itens, o que indica que o ano induzia superestimação ao extrapolar o crescimento de 2017. Com apenas o *lag-1* e a categoria, a Regressão Linear chegou a 16,51%, valor próximo ao do *Naive*, o que é esperado, pois a previsão passa a ser essencialmente o valor do mês anterior ajustado por categoria. As árvores foram melhores nas duas configurações “Só *lag-1*” (19,48% a 20,30%) do que nas que mantêm *lag-2* e média móvel (24,18% a 25,98%). Nenhuma configuração superou o *Naive* (16,11%).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Tabela 4. WAPE (%) por configuração de variáveis no teste de junho a agosto de 2018 (Sem Pareto). Fonte: elaborado pelos autores (2026)

Configuração	Variáveis numéricas	Reg. Linear	*Random Forest*	*Gradient Boosting*
Original	ano, mês, novembro, *lag-1*, *lag-2*, média móvel	22,61	24,76	25,94
Sem ano	mês, novembro, *lag-1*, *lag-2*, média móvel	18,30	24,18	25,98
Só *lag-1*	ano, mês, novembro, *lag-1*	28,31	19,94	19,48
Só *lag-1*, sem ano	mês, novembro, *lag-1*	18,50	20,30	19,53
Lag-1 puro	*lag-1*	16,51	22,37	22,06

Nota: o WAPE do *Naive* no mesmo período é de 16,11%.

4.4. Validação *walk-forward* e melhorias de modelagem

Como o teste único usa apenas três meses de estabilização da demanda, aplicou-se a validação *walk-forward* (Tabela 5). O *Naive* teve o menor WAPE em todos os cenários e janelas. A menor diferença ocorreu no cenário Com Pareto com pelo menos seis meses de treino (21,14% do *Naive* contra 21,35% da Regressão Linear), um empate prático de 0,21 ponto percentual. Nesse recorte e no cenário Sem Pareto com pelo menos seis meses, o RMSE da Regressão Linear (101,82 e 55,33) ficou ligeiramente abaixo do *Naive* (105,40 e 55,73), enquanto o MAE e o WAPE do *Naive* permaneceram menores. Mês a mês, os modelos de aprendizado de máquina tiveram WAPE menor que o *Naive* em 5 a 7 dos 16 meses, sempre de forma isolada, e o *Naive* foi superior em todos os meses de abril a agosto de 2018 no cenário Sem Pareto. Em dezembro de 2017, mês seguinte ao pico, apenas a Regressão Linear superou o *Naive* (31,88% contra 43,03%), e as árvores tiveram erro maior (46,26% e 46,39%).

Tabela 5. WAPE agregado (%) na validação *walk-forward*, com as variáveis originais. Fonte: elaborado pelos autores (2026)

Modelo	Sem Pareto, 16 passos	Sem Pareto, ≥ 6 meses de treino	Com Pareto, 16 passos	Com Pareto, ≥ 6 meses de treino
Naive	**23,92**	**23,72**	**21,36**	**21,14**
Regressão Linear	26,25	24,82	23,97	21,35
Random Forest	26,28	26,26	23,10	23,10
Gradient Boosting	27,10	27,13	25,16	25,62

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

A Tabela 6 mostra que nenhuma das melhorias aproximou os modelos do *Naive* (23,92% com 16 passos e 23,72% com pelo menos seis meses). Retirar o ano e as variáveis redundantes (B) piorou a Regressão Linear, de 26,25% para 28,65%, ao contrário do observado no teste único: o erro médio foi de +0,69 para +11,59 itens, isto é, o modelo passou a subestimar a demanda. Durante o crescimento de 2017, o ano funcionava como indicador de tendência, e a mesma variável que causava superestimação no platô de 2018 ajudava no período de crescimento; escolher variáveis com base em uma única janela de teste pode, portanto, não se sustentar em outra. Mudar o alvo para variação (C) e aumentar o número de árvores (D) não trouxeram melhora, e os hiperparâmetros regularizados (E) reduziram um pouco o erro (28,27% e 28,63%), sem se aproximar do *Naive*. Diferenças de poucos décimos entre C, D e E não devem ser interpretadas, dado o ruído de semente do *Gradient Boosting* (o WAPE agregado do *walk-forward* Sem Pareto oscilou entre 26,47% e 26,96% ao variar a semente). A Figura 5 mostra que o *Naive* fica abaixo ou próximo dos modelos na maior parte dos meses, com diferença maior nos primeiros passos e em dezembro de 2017, quando as árvores chegam a cerca de 70% de WAPE contra 43% do *Naive*.

Tabela 6. WAPE (%) por degrau no *walk-forward* Sem Pareto (16 passos / passos com ≥ 6 meses de treino). Fonte: elaborado pelos autores (2026)

Degrau	Reg. Linear	*Random Forest*	*Gradient Boosting*
A original	26,25 / 24,82	26,28 / 26,26	27,10 / 27,13
B enxuta (sem *lag-2*, média móvel e ano)	28,65 / 26,04	27,52 / 26,61	27,32 / 26,11
C B + alvo em variação	–	28,66 / 27,83	29,18 / 27,67
D C + mais árvores	–	28,90 / 28,07	30,35 / 29,01
E D + hiperparâmetros regularizados	–	28,27 / 27,47	28,63 / 27,46

Nota: a Regressão Linear aparece apenas nos degraus A e B, pois trocar o alvo por variação não altera suas previsões (o *lag-1* já é variável explicativa).

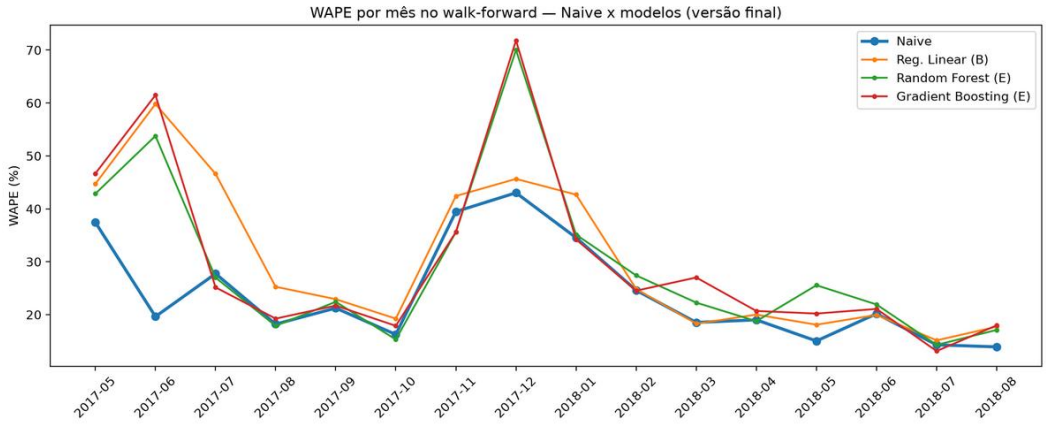
A Tabela 7 detalha o WAPE de cada passo do *walk-forward* para o *Naive* e para as versões finais dos modelos (degraus B e E). Nessas versões, os modelos tiveram WAPE menor que o *Naive* em 2 (Regressão Linear), 6 (*Random Forest*) e 5 (*Gradient Boosting*) dos 16 meses. O erro é alto para todos nos passos iniciais, com um a três meses de treino, e em dezembro de 2017, mês seguinte ao pico de novembro; de fevereiro a agosto de 2018, os valores mensais de WAPE ficam entre 13% e 28% para todos os modelos.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Tabela 7. WAPE mensal (%) no *walk-forward* Sem Pareto. Fonte: elaborado pelos autores (2026)

Mês-alvo	Meses de treino	*Naive*	Reg. Linear (B)	*Random Forest* (E)	*Gradient Boosting* (E)
05/2017	1	37,44	44,72	42,86	46,68
06/2017	2	19,63	59,81	53,78	61,49
07/2017	3	27,74	46,61	27,00	25,15
08/2017	4	18,20	25,27	17,88	19,24
09/2017	5	21,24	22,92	22,42	21,71
10/2017	6	16,24	19,24	15,30	17,88
11/2017	7	39,47	42,42	35,61	35,61
12/2017	8	43,03	45,65	70,03	71,81
01/2018	9	34,54	42,68	35,10	34,21
02/2018	10	24,59	24,86	27,41	24,52
03/2018	11	18,52	18,28	22,27	27,02
04/2018	12	19,01	20,00	18,76	20,69
05/2018	13	15,02	18,09	25,56	20,20
06/2018	14	20,17	19,94	21,92	21,08
07/2018	15	14,29	15,13	14,27	13,11
08/2018	16	13,90	17,72	17,11	17,97

Figura 5. WAPE mensal no *walk-forward* (Sem Pareto): *Naive*, Regressão Linear (degrau B), *Random Forest* e *Gradient Boosting* (degrau E)



Fonte: elaborado pelos autores (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.5. Discussão e limitações

Em todas as análises, o *Naive* teve WAPE menor ou, em um único recorte, equivalente ao dos modelos de aprendizado de máquina. O conjunto de evidências é compatível com quatro explicações, que devem ser tratadas como hipóteses, e não como resultados comprovados: (1) mudança de regime, pois a demanda cresceu no treino e se estabilizou no teste, e os modelos extrapolaram o crescimento; (2) histórico curto, já que 20 meses utilizáveis e uma única *Black Friday* não separam sazonalidade anual de crescimento da plataforma; (3) variação mensal pouco previsível, de modo que prever variação nula (o *Naive*) é difícil de superar e as estimativas de variação feitas pelos modelos acrescentam ruído, o que é coerente com o alvo em variação não ter melhorado os resultados; e (4) heterogeneidade entre categorias, que dificulta o trabalho de um único modelo global.

As limitações são: não foi realizado teste estatístico formal, de modo que diferenças pequenas, como a diferença de 0,21 ponto percentual do empate prático, não são interpretadas como significativas; os resultados do *Gradient Boosting* variam com a semente e com a versão da biblioteca; a busca de hiperparâmetros se limitou a um conjunto definido previamente; as melhorias de modelagem foram avaliadas apenas no cenário Sem Pareto; a demanda considera apenas itens de pedidos entregues; e o histórico é curto (24 meses, dos quais 20 utilizáveis, com uma única *Black Friday* e três meses de teste), de modo que os resultados se restringem ao conjunto de dados da Olist e a superioridade observada do *Naive* não pode ser generalizada a outros conjuntos de dados, segmentos ou horizontes sem nova validação.

4.6. Ameaças à validade

Quanto à validade interna, os cuidados contra vazamento descritos na Seção 3.4 reduzem, mas não eliminam, o risco de conclusões espúrias: as configurações de variáveis da Seção 4.3 foram comparadas nos mesmos meses de teste, o que dá à análise caráter de sensibilidade. Quanto à validade de construto, a demanda foi medida pelos itens de pedidos entregues, por mês de compra, e não pela demanda potencial: períodos de ruptura de estoque ou de cancelamento podem subestimá-la, e o atraso entre compra e entrega não é modelado.

Quanto à validade externa, os resultados se referem a um único conjunto de dados, de 24 meses, de uma plataforma que estava em fase de crescimento acelerado no início da série e se estabilizou ao final. É possível que, em séries mais longas, com várias sazonalidades anuais completas, ou em segmentos com demanda mais regular, os modelos de aprendizado de máquina superem o *Naive*; este estudo não permite afirmar nem negar isso. A contribuição se limita a mostrar que, nas condições avaliadas, a vantagem esperada da complexidade não se materializou, e que o *baseline* deve ser sempre calculado e reportado.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.7. Implicações para o uso operacional

Do ponto de vista do planejamento de estoques, o resultado indica que, para este conjunto de dados, a previsão da demanda do mês seguinte por categoria pode partir do valor observado no mês anterior sem perda frente aos modelos avaliados. Isso não significa que modelos mais elaborados sejam inúteis, mas que só se justificam quando superam, em validação adequada, uma referência simples e barata de manter. O *pipeline* desenvolvido torna essa verificação repetível: com os dados organizados em camadas e as variáveis geradas de forma padronizada, novos modelos, novas variáveis e novos esquemas de validação podem ser avaliados contra o mesmo *baseline*.

A comparação também mostra que a escolha do esquema de avaliação altera conclusões parciais, como o efeito de retirar o ano (que melhorou a Regressão Linear no teste único e a piorou no *walk-forward*). Por isso, recomenda-se que decisões sobre variáveis e hiperparâmetros sejam tomadas com mais de uma janela temporal e, sempre que possível, com a incerteza das estimativas explicitada.

5. CONSIDERAÇÕES FINAIS

O trabalho desenvolveu um *pipeline* de dados em camadas, do armazenamento bruto à visualização, e o integrou à comparação de modelos de previsão de demanda com dados da Olist. O *baseline Naive*, que usa a demanda do mês anterior, teve WAPE igual ou menor que o dos modelos de Regressão Linear, *Random Forest* e *Gradient Boosting* no teste principal, na sensibilidade ao conjunto de variáveis, na validação *walk-forward* e nas melhorias de modelagem, com empate prático em um único recorte. Esse resultado mostra que a complexidade do algoritmo não deve ser o único critério de seleção de uma solução preditiva e que *baselines* simples devem ser referência obrigatória. Esse desempenho superior do *Naive* refere-se ao conjunto de dados analisado, de histórico temporal curto, e não deve ser generalizado para outros conjuntos de dados, segmentos ou horizontes sem nova validação.

A principal contribuição está na integração entre engenharia de dados, análise preditiva e visualização, e na demonstração de que a escolha do modelo deve se basear em evidências obtidas nos próprios dados e em mais de uma janela de validação. Recomenda-se, em trabalhos futuros, incorporar bases empresariais com estoque, compras e prazos de fornecedores, ampliar o histórico disponível, incluir variáveis exógenas, testar modelos específicos de séries temporais e aplicar testes estatísticos de comparação entre modelos.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REFERÊNCIAS

BRANDÃO, Adriano Martins Pataro. **Criação de dashboard analítico para dados de seguros de automóveis utilizando ETL e banco de dados**. 2025. 101 f. Monografia (Graduação em Engenharia de Produção) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2025.

CHOPRA, Sunil; MEINDL, Peter. **Gestão da cadeia de suprimentos: estratégia, planejamento e operação.** 6. ed. São Paulo: Pearson, 2016.

DATABRICKS. Medallion architecture. [S. l.]: Databricks, 2023. Disponível em: https://www.databricks.com/glossary/medallion-architecture. Acesso em: 10 abr. 2026.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow**. 3. ed. Sebastopol: O'Reilly Media, 2022.

HYNDMAN, Rob J.; ATHANASOPOULOS, George. **Forecasting**: Principles and Practice. 3. ed. Melbourne: OTexts, 2021. Disponível em: https://otexts.com/fpp3/. Acesso em: 3 jun. 2026.

KLEPPMANN, Martin. **Designing Data-Intensive Applications**: The Big Ideas Behind Reliable, Scalable and Maintainable Systems. Sebastopol: O'Reilly Media, 2017.

KOLASSA, Stephan; SCHÜTZ, Wolfgang. Advantages of the MAD/Mean ratio over the MAPE. **Foresight: The International Journal of Applied Forecasting**, n. 6, p. 40-43, 2007.

MAKRIDAKIS, Spyros; SPILIOTIS, Evangelos; ASSIMAKOPOULOS, Vassilios. Statistical and Machine Learning forecasting methods: concerns and ways forward. PLOS ONE, v. 13, n. 3, e0194889, 2018. DOI: 10.1371/journal.pone.0194889.

MCKINNEY, Wes. **Python for Data Analysis**: Data Wrangling with Pandas, NumPy, and IPython. 2. ed. Sebastopol: O'Reilly Media, 2018.

OLIST. **Brazilian E-Commerce Public Dataset by Olist**. Kaggle, 2018. Disponível em: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce. Acesso em: 27 jul. 2026.

PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825-2830, 2011.

PROVOST, Foster; FAWCETT, Tom. Data science and its relationship to big data and data-driven decision making. **Big Data**, v. 1, n. 1, p. 51-59, 2013. DOI: 10.1089/big.2013.1508.

REIS, Joe; HOUSLEY, Matt. **Fundamentos de engenharia de dados: projete** e construa sistemas de dados robustos. São Paulo: Novatec, 2023.

SILVA, Isadora S. da; CASSIANO, Mateus B. **Big Data e processamento de dados: uma jornada para a descoberta de insights. 2024. Trabalho de Conclusão de Curso (Graduação) – Universidade Federal de Mato Grosso do Sul, Campo Grande, 2024.**

SQLALCHEMY. **SQLAlchemy Documentation**. 2024. Disponível em: https://docs.sqlalchemy.org/. Acesso em: 15 abr. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.