



Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação



Danilo Henrique Barbosa de Oliveira

Aprimoramento das Previsões de Demanda: Uma Abordagem Integrada com Machine Learning e Tratamento Eficiente de Dados

Monografia de Graduação

Ouro Preto, 2025

Danilo Henrique Barbosa de Oliveira

**Aprimoramento das Previsões de Demanda: Uma
Abordagem Integrada com Machine Learning e
Tratamento Eficiente de Dados**

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Profa. Dra. Adrielle de Carvalho Santana

Coorientador: Prof. Dr. Jadson Castro Gertrudes

Ouro Preto

2025



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Danilo Henrique Barbosa de Oliveira

Aprimoramento das Previsões de Demanda: Uma Abordagem Integrada com Machine Learning e Tratamento Eficiente de Dados

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 08 de Abril de 2025

Membros da banca

Dra. Adrielle de Carvalho Santana - Orientadora - Universidade Federal de Ouro Preto
Dr. Jadson Castro Gertrudes - Coorientador - Universidade Federal de Ouro Preto
Luis Gustavo de Oliveira Miranda - Convidado - Universidade Federal de Ouro Preto
M.Sc. Josemar Coelho Felix - Convidado - Universidade Federal de Ouro Preto

Adrielle de Carvalho Santana, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 09/04/2025



Documento assinado eletronicamente por **Adrielle de Carvalho Santana, PROFESSOR DE MAGISTERIO SUPERIOR**, em 09/04/2025, às 20:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0892119** e o código CRC **CEA54452**.

Agradecimentos

Gostaria de dedicar este espaço para expressar minha sincera gratidão a todos que desempenharam um papel crucial na realização deste trabalho. Agradeço a Deus por me iluminar neste momento crucial da minha formação, reconhecendo que, sem Vossa força, eu não teria alcançado este patamar. Meu agradecimento é profundo e genuíno, estendendo-se aos meus pais, Arilton e Dalva, assim como aos meus irmãos, Arthur e Maria Clara. Sua presença constante ao meu lado, oferecendo apoio incondicional e estimulando meu desenvolvimento acadêmico, é um tesouro que valorizo imensamente.

Expresso meu reconhecimento aos amigos que fizeram sacrifícios significativos para viabilizar a concretização deste momento. Não posso deixar de mencionar a República Molotov, cuja diligência e inspiração foram fundamentais para este empreendimento.

À minha namorada, Carol, dirijo meus agradecimentos pelo seu carinho, paciência, ensinamentos e motivação, elementos que sempre me impulsionaram a dar o meu melhor.

De maneira especial, manifesto minha gratidão aos meus orientadores, Prof. Dra. Adrielle e Prof. Dr. Jadson, pelo apoio inabalável e pela orientação que me guiaram durante este período de crescimento acadêmico, pessoal e profissional. Ao professor Danny, expresso meu reconhecimento pela paciência e didática exemplares que contribuíram de maneira fundamental para a concretização deste trabalho.

*Até a menor pessoa pode mudar
o curso do futuro.*

— J.R.R. Tolkien.

Resumo

Na era globalizada atual, a crescente competitividade e a necessidade de redução de custos impulsionam a adoção de métodos precisos para previsão de demanda. Neste trabalho, foram aplicadas técnicas avançadas de *machine learning* a dados transacionais do varejo online, visando aprimorar a acurácia das previsões e auxiliar na gestão de estoques. Para tanto, foram avaliados modelos baseados em árvores de decisão – *Decision Tree*, *Random Forest* e *XGBoost* – bem como um modelo de rede neural do tipo *Long Short-Term Memory* (LSTM). O pré-processamento envolveu a normalização dos dados, tratamento de valores faltantes e identificação de *outliers* por meio dos métodos *z-score* e *IQR*. A validação dos modelos foi realizada utilizando a técnica *TimeSeriesSplit*, que preserva a ordem temporal dos dados. Os resultados demonstraram que, utilizando configurações padrão, o *XGBoost* apresentou melhores métricas (MAE = 4.24, RMSE = 5.88, MAPE = 2.26%), enquanto a otimização dos hiperparâmetros reduziu os erros tanto no *Random Forest* quanto no *XGBoost* (MAE \approx 4.21, RMSE \approx 5.83, MAPE \approx 2.18%). Os modelos baseados em árvores demandaram tempo de treinamento entre 14 e 17 minutos, sendo significativamente mais eficientes que o modelo LSTM, que, apesar do tempo de treinamento elevado (aproximadamente 44 minutos), evidenciou consistência na captura de padrões temporais (MAE = 4.37, RMSE = 5.98, MAPE = 1.57%). Conclui-se que a combinação de um pré-processamento rigoroso com modelos avançados de *machine learning* resulta em ganhos expressivos na acurácia das previsões, proporcionando suporte robusto para decisões estratégicas e melhoria na eficiência operacional das empresas.

Palavras-chaves: previsão de demanda, machine learning, redes neurais, Random Forest, XGBoost, Decision Tree, LSTM.

Abstract

In the current globalized era, increasing competitiveness and the need to reduce costs drive the adoption of precise demand forecasting methods. In this work, advanced machine learning techniques were applied to online retail transactional data to enhance forecasting accuracy and support inventory management. Models based on decision trees – including Decision Tree, Random Forest, and XGBoost – as well as a Long Short-Term Memory (LSTM) neural network were evaluated. Data preprocessing involved normalization, handling missing values, and outlier detection using the z-score and IQR methods. Model validation was performed with TimeSeriesSplit to preserve the temporal order of the data. Results showed that, with default settings, XGBoost achieved superior performance (MAE = 4.24, RMSE = 5.88, MAPE = 2.26%), while hyperparameter tuning reduced errors for both Random Forest and XGBoost (MAE \approx 4.21, RMSE \approx 5.83, MAPE \approx 2.18%). Tree-based models required training times between 14 and 17 minutes, proving to be significantly more efficient than the LSTM model, which, despite its longer training time (approximately 44 minutes), consistently captured complex temporal patterns (MAE = 4.37, RMSE = 5.98, MAPE = 1.57%). In conclusion, integrating rigorous data preprocessing with advanced machine learning models leads to significant improvements in forecasting accuracy, supporting strategic decision-making and enhancing operational efficiency.

Key-words: demand forecasting, machine learning, neural networks, Random Forest, XGBoost, Decision Tree, LSTM.

Lista de ilustrações

Figura 1 – Regressão Linear	20
Figura 2 – Estrutura de uma Árvore de Decisão	21
Figura 3 – Estrutura <i>Random Forest</i>	22
Figura 4 – Rede Neural Artificial	28
Figura 5 – Validação Cruzada Utilizando <i>K-Fold</i>	31
Figura 6 – Divisão de séries temporais utilizando <i>Time Series Split</i> . Fonte: Elaborado pelo autor.	33
Figura 7 – Recorte do banco de dados	34
Figura 8 – Distribuição de transações por país	39
Figura 9 – Distribuição geográfica das transações comerciais realizadas Fonte: Elaborado pelo autor.	40
Figura 10 – Top 10 itens por valor total de vendas	41
Figura 11 – Volume de vendas mensal no Reino Unido	42
Figura 12 – Curva ABC	43
Figura 13 – Distribuição dos itens por classe	43
Figura 14 – Transações por país	44
Figura 15 – Vendas Semanais - Reino Unido	45
Figura 16 – Vendas Mensais - Reino Unido	45
Figura 17 – Matriz de Correlação	47
Figura 18 – Distribuição comparativa de <i>outliers</i> por método. Fonte: Elaborado pelo autor.	49
Figura 19 – Matriz de Correlação dos Resíduos. Fonte: Elaborado pelo autor.	52
Figura 20 – Distribuição de <i>y_train</i> . Fonte: Elaborado pelo autor.	55

Lista de tabelas

Tabela 1 – Distribuição de vendas por país em Libras Esterlinas (£) e percentual de vendas. Fonte: Elaborado pelo autor.	38
Tabela 2 – Comparação do Coeficiente de Variação antes e após o tratamento. . .	47
Tabela 3 – Resumo comparativo dos métodos de identificação de <i>outliers</i>	48
Tabela 4 – Top 5 <i>outliers</i> mais extremos por <i>z-score</i>	49
Tabela 5 – Análise das Variâncias dos Componentes de Séries Temporais	51
Tabela 6 – Proporções de Variabilidade Retirada e Restante	51
Tabela 7 – Tamanhos dos <i>Splits</i> na Divisão Temporal	54
Tabela 8 – Resultados da avaliação dos modelos sem otimização de hiperparâmetros.	60
Tabela 9 – Resultados da avaliação dos modelos com otimização de hiperparâmetros.	61
Tabela 10 – Resultados da avaliação do modelo LSTM.	61

Sumário

1	INTRODUÇÃO	11
1.1	Contextualização	11
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
1.3	Organização e estrutura	13
2	REVISÃO DE LITERATURA	14
2.1	Previsão de Vendas	14
2.2	Aplicação de Técnicas de Aprendizado de Máquina em Previsão de Demanda	15
2.2.1	Ensaio sobre Previsão de Vendas no Varejo da Moda	15
2.2.2	Estudos sobre Redes Neurais e Modelos Híbridos no Varejo	16
2.3	Séries temporais	17
2.4	Aprendizado de Máquina	17
2.5	Overfitting e Underfitting	18
2.6	Regressão	19
2.6.1	<i>Decision Tree</i>	20
2.6.2	<i>Random Forest</i>	21
2.6.2.1	<i>N Estimators</i>	23
2.6.2.2	<i>Max Depth</i>	23
2.6.2.3	<i>Min Samples Split</i>	23
2.6.2.4	<i>Min Samples Leaf</i>	23
2.6.2.5	<i>Max Features</i>	23
2.6.3	XGBoost	23
2.7	Métricas de Avaliação em Modelos de Regressão	25
2.8	Redes Neurais Artificiais	27
2.8.1	Rede Neural de Propagação Direta	28
2.8.2	Backpropagation	29
2.8.3	Treinamento da Rede Neural	30
2.8.3.1	Batch Size	30
2.8.3.2	Taxa de Aprendizado	30
2.8.3.3	Função de Ativação	30
2.8.3.4	Camadas	31
2.8.3.5	Algoritmo de Otimização	31

2.9	Validação Cruzada	31
2.9.1	<i>Time Series Split</i>	32
3	DESENVOLVIMENTO	34
3.1	Base de Dados	34
3.1.1	Explorando as variáveis do banco de dados	34
3.1.2	Importação e Manipulação do Banco de Dados	36
3.1.3	Motivo da Exclusão	37
3.1.4	Análise Exploratória dos Dados	37
3.1.4.1	Países presentes no conjunto de dados	37
3.2	Metodologia de Análise dos Dados	40
3.2.1	Filtragem e Seleção de Dados	40
3.2.2	Análise Descritiva dos Dados	41
3.2.3	Análise da Curva ABC	41
3.2.4	Análise Temporal Detalhada	44
3.3	Análise de Preços e Outliers	46
3.3.1	Metodologia de Normalização e Detecção de <i>Outliers</i>	46
3.3.2	Identificação de <i>Outliers</i>	48
3.3.3	Decomposição de Séries Temporais	50
3.3.4	Análise de Variância	50
3.3.5	<i>Lags e Rolling Windows</i>	51
3.3.5.1	Importância das <i>Lags</i> e <i>Rolling Windows</i>	51
3.3.6	Matriz de Correlação	52
3.4	Divisão do Dataset	53
3.4.1	Divisão Temporal dos Dados	54
3.4.2	Distribuição de <i>y_train</i>	54
3.5	Treinamento dos Modelos de Machine Learning	55
3.5.1	Descrição Geral da Implementação	55
3.5.2	Divisão dos Dados e Estratégia de Validação	56
3.5.3	Pré-Processamento dos Dados	56
3.5.4	Modelos Base e Seus Hiperparâmetros	56
3.5.5	Modelo LSTM	58
4	RESULTADOS	60
4.1	Análise dos Resultados Obtidos com Decision Tree, Random Forest e XGBoost	60
4.1.1	Parâmetros padrão do pacote <i>sklearn</i>	60
4.1.2	Resultados obtidos com otimização dos hiperparâmetros	61
4.2	Análise dos Resultados Obtidos com LSTM	61
4.3	Discussão Geral dos Resultados	62

5	CONCLUSÃO	64
5.1	Limitações	64
5.2	Sugestões para Trabalhos Futuros	65
	Referências	67

1 Introdução

1.1 Contextualização

Na presente era globalizada, o crescente impulso pela competitividade e a necessidade de reduzir custos têm incentivado as indústrias a implementar em métodos de previsão de demanda mais precisos. De acordo com [Passari \(2003\)](#), a gestão eficaz de qualquer empresa, seja ela no âmbito público ou privado, industrial ou varejista, requer um processo cuidadoso de planejamento. Para que esse planejamento seja verdadeiramente efetivo, é essencial ter uma compreensão precisa das condições futuras nas quais a corporação irá operar, bem como entender a interação dos elementos que condicionam essas expectativas.

As decisões estratégicas tomadas pelas corporações levam em consideração os aspectos e elementos do setor nas quais elas estão inseridas. Para apoiar tais decisões, as empresas desenvolvem sistemas e procedimentos para explorar cenários com base em informações quantitativas e/ou qualitativas. Com o avanço tecnológico, modelos e técnicas quantitativas de previsão complementam as análises qualitativas, proporcionando maior precisão.

De acordo com o estudo de [Dias \(1999\)](#), o processo de previsão pode ser definido como o conjunto de todas as atividades envolvidas para que seja possível a geração dos números finais da previsão, o que inclui desde a etapa de coleta de dados até a última revisão da previsão.

As previsões desempenham um papel crucial na tomada de decisões na cadeia de suprimentos, em um cenário onde os clientes buscam cada vez mais disponibilidade de produtos e agilidade na entrega. A gestão eficaz de recursos escassos é vital para garantir a satisfação do cliente, evitar perdas de vendas e ruptura.¹ Segundo [Ali et al. \(2009\)](#) um modelo de previsão de demanda adequado pode contribuir para o aumento do lucro e rentabilidade de uma empresa, entre outros fatores, por facilitar a gestão dos estoques e reduzir o nível de segurança dos estoques consideravelmente. Ainda no mesmo trabalho, [Ali et al. \(2009\)](#) indicam que para períodos promocionais os métodos de *machine learning* apresentam resultados até cerca de 65% melhores do que métodos tradicionais, por meio do uso de algoritmos de árvore.

Dentro do campo de estudo da previsão de demanda utilizando *machine learning*, alguns erros de previsão podem ocorrer, contaminando as decisões para gestão eficiente. No

¹ A ruptura de estoque é a falta de um determinado produto no momento de compra do consumidor, afetando tanto o nível de serviço prestado como o nível de satisfação do consumidor junto a uma determinada marca ([BESSA, 2021](#), p. 17)

trabalho de Passari (2003), pode ser observado como os erros verificados pelo modelo de Redes Neurais podem ser altos e como os desvios podem ser ainda maiores para horizontes de tempo mais amplos. Ainda de acordo com o mesmo trabalho de Passari (2003), vários fatores podem explicar o alto erro nas previsões obtidas com a metodologia adotada, dentre elas o conjunto de dados com valores faltantes *missing*, dados inconsistentes e baixo número de registros por produtos, levam a esta alta na taxa de erro.

Os autores Gea et al. (2021) conceituam em seu trabalho os *outliers* como pontos extremos existentes em uma série temporal. Os *outliers* representam observações que se desviam acentuadamente do comportamento usual das demais observações em um conjunto de dados. Ainda no mesmo trabalho, os autores Gea et al. (2021) definem os *outliers* como uma anomalia existente nas séries temporais, que podem levar a especificação incorreta de modelos, estimativas viesadas dos parâmetros e avaliação incorreta das previsões, pois interfere na relação entre as observações passadas e futuras.

Assim, é interessante que se busque aprimorar as previsões de demanda, focando especialmente no tratamento eficiente de dados para obter resultados mais precisos. Nesse contexto, a implementação de métodos avançados de *machine learning* é considerada uma abordagem promissora. Além disso, é válido explorar outras técnicas e estratégias que visam mitigar os desafios identificados na literatura em relação aos erros de previsão, tais como *outliers* e dados faltantes.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é aprimorar as previsões de demanda, com foco no tratamento eficiente de dados, mediante a implementação de métodos avançados de *machine learning*. A pesquisa busca contribuir para a eficácia da gestão de recursos e a redução de erros de previsão em ambientes empresariais.

1.2.2 Objetivos Específicos

1. Identificar os principais desafios relacionados à previsão de demanda, com ênfase na influência de *outliers* e dados faltantes nos resultados.
2. Explorar diferentes técnicas de pré-processamento de dados para lidar eficazmente com *outliers* e valores faltantes.
3. Desenvolver e avaliar modelos de *machine learning*, utilizando conjuntos de dados específicos do domínio, com o objetivo de prever a demanda de produtos em diferentes horizontes de tempo.

4. Comparar os resultados obtidos pelos modelos propostos, analisando o impacto do pré processamento realizado.
5. Fornecer recomendações e melhores práticas para aprimorar a eficácia das previsões de demanda em contextos empresariais, considerando a interpretabilidade dos modelos.

1.3 Organização e estrutura

O presente trabalho está organizado da seguinte forma: O [Capítulo 1](#) apresenta a justificativa e relevância do assunto, a metodologia e os objetivos do trabalho; No [Capítulo 2](#) apresentamos a fundamentação teórica do conteúdo abordado; No [Capítulo 3](#) apresentamos a organização para desenvolvimento e execução dos experimentos; o [Capítulo 4](#) apresenta os resultados obtidos e a discussões desses resultados. Por fim, o [Capítulo 5](#) apresenta as conclusões, limitações e sugestões para a continuação desse trabalho.

2 Revisão de literatura

2.1 Previsão de Vendas

Para fortalecer a vantagem competitiva em um ambiente dinâmico, os líderes organizacionais precisam tomar decisões precisas no momento adequado, utilizando informações disponíveis ([PASSARI, 2003](#)). Uma previsão precisa desempenha um papel crucial, permitindo compreender o comportamento do consumidor diante das mudanças nos fatores controláveis do varejo. Ao analisar o composto de marketing, os gestores podem antecipar os resultados de suas ações, obtendo dados diagnósticos para embasar decisões futuras.

No trabalho de [Makridakis, Wheelwright e McGee \(1983\)](#), os autores ressaltam os fatores que contribuíram para o aumento da importância da previsão:

- A crescente complexidade das organizações, evidenciada pelo aumento do número de nichos de mercado e de produtos oferecidos, juntamente com as mudanças tecnológicas e na estrutura da demanda. Essa complexidade torna desafiador considerar todos os fatores relacionados ao desenvolvimento futuro da organização.
- A adoção por parte das organizações de procedimentos decisórios mais sistemáticos, que demandam justificativas explícitas para cada ação tomada. A formalização de previsões de vendas emerge como uma maneira de apoiar esses procedimentos.
- O constante avanço das técnicas de previsão e suas aplicações, possibilitando não apenas a compreensão e utilização dessas técnicas por analistas especializados, mas também por gerentes e outros tomadores de decisão.

De acordo com [Rob J. Hyndman e Athanasopoulos \(2021\)](#), a previsibilidade de certas variáveis difere, sendo influenciada por diversos fatores. Alguns elementos impactam a capacidade de prever uma série temporal, incluindo:

- O nível de compreensão dos fatores correlacionados à variável ou evento em estudo;
- A disponibilidade de dados relevantes;
- O potencial impacto das ações decorrentes das previsões nos valores futuros.

Prever a demanda pode ser encarado como um desafio complexo, especialmente diante da escassez e/ou ausência de dados, ou ainda na presença de valores atípicos. Os registros de vendas, considerados em séries temporais, são abordados por diversos modelos já desenvolvidos, como *Holt-Winters*, ARIMA, SARIMA, SARIMAX, GARCH, entre outros (PAVLYSHENKO, 2019). A aplicação prática revela que, em comparação com as séries temporais, a utilização de regressão pode oferecer resultados superiores, pois os algoritmos de *machine learning* conseguem identificar padrões nos modelos de Séries Temporais (SHERSTINSKY, 2020). É possível descobrir até mesmo padrões complexos nos históricos de previsão de vendas por meio da aplicação de modelos supervisionados de *machine learning*, sendo os algoritmos *Random Forest* e *Tree Based Machine Learning* os mais populares para esse propósito (JAMES et al., 2013).

2.2 Aplicação de Técnicas de Aprendizado de Máquina em Previsão de Demanda

2.2.1 Ensaios sobre Previsão de Vendas no Varejo da Moda

O emprego de técnicas de aprendizado de máquina tem sido amplamente utilizado em diversas áreas, sobretudo por meio da regressão, para a realização de previsões. No trabalho “Ensaios sobre Previsão de Vendas no Varejo da Moda”, da autora Bessa (2021), foi realizado um estudo para tornar as previsões de vendas mais precisas em termos de quantidade para setor da moda, com foco nas perspectivas: aplicação de tratamentos sobre dados, especificamente o tratamento da sazonalidade; seleção de variáveis; e consideração de novos métodos de previsão.

A autora propôs a utilização dos modelos XGBoost com tratamento sazonal por ajuste de curva e com tratamento de seleção de variáveis RFA, *LightGBM* (LGBM), *Random Forest* e Redes Neurais (CNN, MLP e LSTM). A métrica principal do trabalho foi o RMSLE, porém também foram calculadas outras métricas como: MAE, RMSE e MAPE, além de uma adaptação nas métricas RMSE e MAPE, sendo criadas RMSEP e MAEP. No contexto exclusivo da métrica de acurácia, o algoritmo mais eficaz, aliado aos tratamentos de sazonalidade e seleção de variáveis mais promissores, considerando o critério de frequência entre as seis métricas de precisão analisadas para cada uma das categorias de produtos, foram os seguintes:

- **Camiseta de homem:** LGBM sem tratamento sazonal e com tratamento de seleção de variáveis RFA;
- **Blusa de mulher:** XGBOOST com tratamento sazonal por ajuste de curva e com tratamento de seleção de variáveis RFA;

- **Camiseta de menino:** MLP com tratamento sazonal por diferença e com tratamento de seleção de variáveis RFA;
- **Blusa de menina:** XGBOOST com tratamento sazonal por ajuste de curva e com tratamento de seleção de variáveis RFA.

A escolha dessas abordagens levou em consideração a análise criteriosa do desempenho em acurácia, destacando-se como as mais adequadas para cada categoria de produto. Diante do exposto, fica evidente que os modelos com a aplicação de algoritmos de redes neurais e o uso do tratamento de seleção de variáveis por RFA afetam significativamente a velocidade de tratamento dos algoritmos considerados. Ademais, o uso do tratamento sazonal teve um impacto bem menor quando comparado com o uso do tratamento de seleção de variáveis (BESSA, 2021).

2.2.2 Estudos sobre Redes Neurais e Modelos Híbridos no Varejo

A utilização de redes neurais do tipo *Long Short-Term Memory (LSTM)* para previsão de demanda tem recebido destaque na literatura, especialmente devido à sua capacidade de capturar dependências temporais e padrões não lineares em séries temporais.

No trabalho de Fukai, Cavalieri e Castro (2024), a autora investigou o desempenho de três variações de *LSTM*: *Vanilla LSTM*, *Conv1D-LSTM* e *Multiscale CNN-LSTM*, aplicadas à previsão de vendas no varejo. Os resultados indicaram que, após otimização bayesiana dos hiperparâmetros, a arquitetura *Conv1D-LSTM* apresentou o melhor desempenho, superando as outras variantes em métricas como *MAE*, *RMSE* e *MASE*. Apesar disso, métodos baseados em árvores de decisão, como *LightGBM* e *XGBoost*, demonstraram desempenho comparável, com custos computacionais menores, destacando-se como alternativas viáveis em cenários com restrições de recursos computacionais.

Em outro estudo, Mitra et al. (2022) realizaram uma análise comparativa de modelos de previsão de demanda para uma empresa de varejo multicanal nos Estados Unidos. Foram avaliados algoritmos como *Random Forest*, *Extreme Gradient Boosting (XGBoost)*, *Gradient Boosting*, *Adaptive Boosting (AdaBoost)* e *Artificial Neural Networks (ANN)*, além de um modelo híbrido que combina *Random Forest*, *XGBoost* e Regressão Linear. Os dados de vendas semanais de diversas lojas foram utilizados, considerando fatores como temperatura e tamanho das lojas. O modelo híbrido (*RF-XGBoost-LR*) apresentou o melhor desempenho, superando os modelos individuais em métricas como *MSE*, R^2 e *MAE*, reforçando o potencial de abordagens híbridas para melhorar a precisão das previsões.

Esses estudos destacam a relevância de diferentes abordagens para previsão de demanda no varejo. Enquanto redes neurais, como *LSTM*, se mostram eficazes em capturar

padrões temporais complexos, métodos híbridos e baseados em árvores de decisão oferecem soluções robustas e computacionalmente eficientes. Dessa forma, a escolha do modelo ideal depende não apenas da precisão, mas também de fatores como custos computacionais e a complexidade dos dados disponíveis.

2.3 Séries temporais

Uma série temporal é qualquer conjunto de observações ordenadas no tempo, cuja ordenação cronológica gera efeitos de autocorrelação (MORETTIN; TOLOI, 2006). Os dados podem ser coletados em intervalos regulares — diariamente, semanalmente, mensalmente, trimestralmente, semestralmente ou anualmente — ou em intervalos extremamente curtos, quase contínuos, como no caso das cotações das ações.

Ao analisar uma série temporal, Morettin e Tolo (2006) destacam que os principais objetivos incluem:

- Investigar o mecanismo gerador da série.
- Criar previsões de valores futuros a curto ou longo prazo.
- Descrever graficamente o comportamento da série (tendências, ciclos, variações sazonais, histogramas e diagramas de dispersão).
- Verificar periodicidades por meio de análise espectral e avaliar a adequação do modelo.

As séries temporais podem ser univariadas (baseadas em uma única variável histórica) ou multivariadas (com mais de uma série) e podem ser classificadas como contínuas ou discretas. A decomposição da série nos seus componentes — tendência, sazonalidade e resíduos — é essencial para compreender os padrões e possibilitar a geração de previsões precisas. Esses fundamentos teóricos servem de base para a aplicação de técnicas de aprendizado de máquina, que se beneficiam de um pré-processamento adequado dos dados temporais.

2.4 Aprendizado de Máquina

Aprendizado de máquina, ou *machine learning*, é uma área de Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (MONARD et al., 2003, p.39). Segundo Mitchell (1997), diz-se que um programa de computador aprende pela experiência, com respeito a algum tipo de tarefa e desempenho, se sua desempenho nas tarefas, na forma medida, melhoram com a experiência.

Conforme destacado por Makridakis et al. (2018), as aplicações primárias da inteligência artificial incluem os sistemas de previsão, cujo sucesso reside principalmente em algoritmos capazes de aprimoramento contínuo por meio de processos de tentativa e erro, diferenciando-se das abordagens tradicionais baseadas exclusivamente em lógica de programação. Os métodos de aprendizado de máquina, especialmente as Redes Neurais, revelam-se altamente eficazes na previsão de séries temporais. Embora haja uma profusão de novos métodos propostos que indicam melhorias na eficácia das previsões, a validação dessa superioridade é limitada, com poucos estudos corroborando tal vantagem. Estes estudos, em geral, restringem-se à análise de um número limitado de séries temporais, frequentemente com horizontes de previsão reduzidos, e oferecem escassa variedade de métodos alternativos para comparação (MAKRIDAKIS et al., 2018).

Os modelos de aprendizado de máquina explorados neste estudo incluem *Decision Tree*, *Random Forest*, *XGBoost* e Rede Neural. Esses modelos foram categorizados como causais, seguindo a classificação proposta por Ali et al. (2009). A seleção destes algoritmos específicos é fundamentada pelas evidências empíricas apresentadas nos estudos recentes de Fukai, Cavalieri e Castro (2024) e Mitra et al. (2022). O estudo de Fukai, Cavalieri e Castro (2024) demonstrou que, apesar da eficácia de arquiteturas neurais complexas como *Conv1D-LSTM* na captura de padrões temporais, algoritmos baseados em árvores de decisão como *LightGBM* e *XGBoost* alcançaram desempenho comparável com menor custo computacional, justificando assim a inclusão do *Decision Tree*, *Random Forest* e *XGBoost* em nossa abordagem. Complementarmente, a pesquisa de Mitra et al. (2022) reforça esta escolha ao evidenciar o potencial de modelos híbridos que integram *Random Forest* e *XGBoost* para superar modelos individuais em métricas relevantes como *MSE* e *MAE*. A inclusão de uma Rede Neural, por sua vez, é respaldada pela capacidade demonstrada das arquiteturas neurais em capturar dependências temporais complexas e padrões não lineares frequentemente presentes em dados de varejo, conforme evidenciado em ambos os estudos citados.

2.5 *Overfitting* e *Underfitting*

Entre os desafios enfrentados por modelos de previsão do tipo causal, como as redes neurais (ALI et al., 2009), está a questão do *overfitting* e *underfitting*. Esses problemas representam especificações excessivamente altas e baixas, respectivamente, durante o treinamento de um modelo de previsão (ZHANG, 2007).

Em seu trabalho, Zhang (2007) explica que o problema de *underfitting* ocorre quando o modelo é elaborado e treinado com uma baixa especificação da importância dos parâmetros, resultando em previsões com grande margem de erro, inclusive para o grupo de teste. Embora seja um problema perceptível durante a implementação do modelo, é

considerado menos crítico em comparação com o *overfitting*.

Já o *overfitting* é mais crítico, e diversos estudos e discussões têm abordado esse tema. Esse problema ocorre quando um modelo de previsão identifica padrões específicos para o conjunto de teste utilizado, ajustando-se demasiadamente aos dados observados. Embora o modelo apresente baixo erro no ambiente de teste, inclusive durante as etapas de validação cruzada, indica que o modelo foi especificado demais para o conjunto de teste, sendo, portanto, ineficaz para prever novos resultados.

Em seu trabalho, os autores [Ayotte e al. \(2021\)](#) abordam outro fator crítico que pode levar ao *overfitting* de um modelo, que exige atenção durante a etapa de validação cruzada: o vazamento de dados, também conhecido como *ata leaked*. Esse é considerado um dos principais erros na análise de dados, caracterizado pela inclusão no treino de teste de valores ou informações pertencentes ao grupo de teste, os quais não deveriam estar disponíveis.

No contexto de séries temporais, [Rob J Hyndman e Athanasopoulos \(2018\)](#) destacam que ao usar a validação cruzada em séries temporais, é crucial garantir que as divisões entre conjuntos de treinamento e teste ocorram de maneira sequencial no tempo. Isso significa que, ao realizar a validação cruzada, os dados de teste para cada iteração estão situados após os dados de treinamento. Essa abordagem reflete mais realisticamente a aplicação do modelo em cenários futuros, onde a previsão é feita com base em informações disponíveis até um determinado ponto no tempo.

2.6 Regressão

Regressão é uma técnica estatística utilizada em análise de dados para modelar e investigar a relação entre uma variável dependente e uma ou mais variáveis independentes. No estudo conduzido por [Goodfellow, Bengio e Courville \(2016\)](#), os pesquisadores destacam a regressão como uma categoria de tarefa na qual um programa de computador é designado para prever um valor numérico com base em uma entrada dada. Este método, de natureza paramétrica, visa estabelecer uma função matemática que capture de forma otimizada a relação entre as variáveis, fundamentando-se em um conjunto de dados específico.

Nesta seção, são introduzidos os algoritmos empregados no decorrer desta pesquisa. Dentre eles, incluem-se: Regressão Linear, *Decision Tree*, *Random Forest* e XGBoost. A Figura 1 ilustra o gráfico de uma regressão linear, em que cada ponto representa um dado de entrada e o seu valor correspondente de saída. O erro é determinado pela distância desses pontos em relação à reta de regressão, que modela a relação $y_i = f(x_i)$.

A regressão, como ferramenta fundamental na análise estatística, desempenha um papel crucial em diversas áreas, proporcionando percepções valiosas e possibilitando

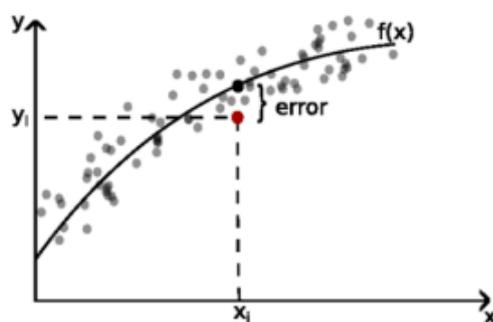


Figura 1 – Regressão Linear.

Fonte: [Santana \(2020\)](#).

previsões significativas. No entanto, é essencial reconhecer que a aplicação da regressão não está isenta de desafios. A seleção inadequada de variáveis ou a presença de *outliers* nos dados podem impactar significativamente a qualidade das previsões. A interpretação correta dos resultados também é crucial, pois uma relação estatística não implica necessariamente causalidade.

2.6.1 *Decision Tree*

Decision Trees, ou Árvores de Decisão, é um método de aprendizado supervisionado utilizado para classificação e regressão. O objetivo é criar um modelo que preveja o valor de uma variável-alvo ao aprender regras de decisão simples inferidas das características dos dados ([BREIMAN et al., 1984](#)).

A Figura 2 ilustra a estrutura de uma Árvore de Decisão, que é composta por diferentes partes: o Nó Raiz, os Nós de Decisão e as Folhas. O Nó Raiz é o ponto inicial, onde os dados começam a ser analisados e as decisões são tomadas. Os Nós de Decisão representam pontos intermediários onde as escolhas são feitas com base em critérios específicos, dividindo os dados em caminhos diferentes. Por fim, as Folhas representam os resultados finais ou previsões do modelo, onde a análise termina. De forma simples, as Árvores de Decisão funcionam ao “quebrar” o problema em partes menores, criando regras simples e intuitivas que facilitam a tomada de decisão.

Em seu trabalho, [Dobra \(2002\)](#) argumenta que as *Decision Trees* são amplamente utilizadas devido às suas três principais vantagens, proporcionando resultados comparáveis a outros métodos. A primeira vantagem é a representação intuitiva da árvore, facilitando a compreensão. A segunda é a não parametricidade das árvores, tornando-as aplicáveis a diversas situações. A terceira vantagem é a facilidade de escalabilidade dos algoritmos de árvores.

No contexto de problemas de regressão, as árvores de decisão são escolhas frequentes,

uma vez que oferecem modelos intuitivos com resultados comparáveis a outros métodos (DOBRA, 2002). Algoritmos fundamentados em árvores de decisão simples demandam pouco esforço na preparação dos dados, conferindo versatilidade para diversas aplicações. Além disso, são computacionalmente eficientes, exigindo baixo poder de processamento, embora sua eficácia na previsão de conjuntos de dados extensos seja limitada (DAWOOD, 2019).

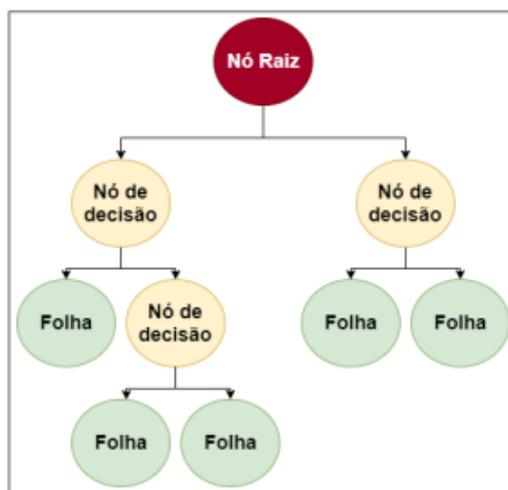


Figura 2 – Estrutura de uma Árvore de Decisão.
Fonte: Delgado Filho (2020).

As Árvores de Decisão apresentam uma solução eficaz para problemas de regressão, construindo-se com base na premissa de que os relacionamentos entre os atributos e o valor a ser previsto podem ser lineares ou não lineares. No contexto da regressão, as árvores se tornam variantes de um classificador utilizado para aproximar funções de valor real. A construção de uma Árvore de Regressão adota o princípio do particionamento binário, onde a estrutura inicial da árvore é definida utilizando toda a amostra dos dados, e as divisões são selecionadas de forma a minimizar a soma dos desvios quadrados da média nas partes segregadas.

2.6.2 *Random Forest*

Random Forest é um algoritmo de aprendizado de máquina, baseado em uma combinação de árvores de decisão (*Decision Tree*), normalmente utilizado para realizar tarefas de regressão e classificação. A técnica que combina múltiplos classificadores e agrega seus resultados é conhecida como “*Ensemble Learning*”.

Random Forest também soluciona problemas de regressão, seguindo a mesma ideia de se obter um conjunto de diferentes Árvores de Regressão. Nesse contexto cada árvore encarrega-se de realizar um mapeamento não linear, dividindo o problema original em problemas menores, solucionável por um conjunto de modelos simples (PHAN et al., 2014).

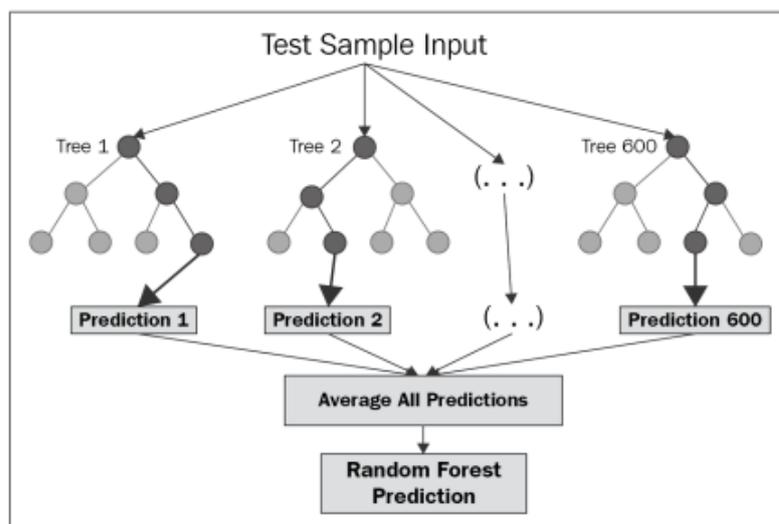


Figura 3 – Estrutura *Random Forest*.
Fonte: Science (2019).

A Figura 3 ilustra de forma esquemática o funcionamento do *Random Forest*. A entrada do modelo é um conjunto de dados de teste, que é processado por múltiplas árvores de decisão independentes (Tree 1, Tree 2, ..., Tree 600). Cada árvore gera uma previsão parcial, como indicado pelos blocos *Prediction 1*, *Prediction 2*, até *Prediction 600*. O resultado final é obtido por meio de da média ou votação das previsões individuais, dependendo se o problema é de regressão ou classificação. Esse processo reduz o risco de *overfitting*, aumentando a capacidade de generalização do modelo.

Para alcançar tal propósito, pressupõe-se que o conjunto de treinamento seja modelado de forma independente em relação aos vetores aleatórios utilizados como entrada do modelo. Adicionalmente, as florestas de regressão requerem uma baixa correlação entre os resíduos e árvores com uma taxa reduzida de erro. Em outras palavras, a randomização aplicada deve visar uma correlação mínima. Diante dessas características, é possível se concluir que o *Random Forest* é uma excelente ferramenta para solucionar problemas de regressão e classificação, já que, por contar com uma combinação de árvores os modelos, não se ajustam demais, causando um problema de *overfitting*, tendo assim um grande poder de generalização (BREIMAN, 2001).

Conforme destacado por Chollet (2021), a *Random Forest* representa uma abordagem robusta e prática para a aplicação de árvores de decisão, mostrando-se, na maioria das situações, capaz de produzir resultados satisfatórios em diversas tarefas de aprendizado de máquina. O treinamento de uma *Random Forest* é uma etapa crucial no desenvolvimento do modelo, onde o sistema aprende com os dados de treinamento e ajusta seus parâmetros para otimizar a eficácia na previsão de novos dados. Os principais hiperparâmetros que podem ser ajustados na *Random Forest* para aprimorar o desempenho do modelo são explicados a seguir.

2.6.2.1 *N Estimators*

O hiperparâmetro *N Estimators* define o número de árvores a serem criadas no modelo. Ao aumentar este valor, a robustez do modelo pode ser ampliada, tornando-o menos suscetível a variações aleatórias nos dados de treinamento. Entretanto, é importante considerar que um número excessivo de árvores pode aumentar o tempo de treinamento e levar o modelo ao sobreajuste.

2.6.2.2 *Max Depth*

O hiperparâmetro *Max Depth* estabelece a profundidade máxima da árvore de decisão, afetando significativamente o desempenho do modelo. Valores muito pequenos resultam em árvores subajustadas, incapazes de capturar a complexidade dos relacionamentos entre as variáveis. Por outro lado, valores muito altos podem levar ao sobreajuste, prejudicando a generalização do modelo.

2.6.2.3 *Min Samples Split*

O hiperparâmetro *Min Samples Split* determina o número mínimo de amostras necessárias para dividir um nó interno durante a construção da árvore. A utilização adequada desse parâmetro pode evitar a superajustagem, impedindo que a árvore se torne excessivamente complexa ou profunda.

2.6.2.4 *Min Samples Leaf*

O hiperparâmetro *Min Samples Leaf* estabelece a quantidade mínima de amostras que devem ser alocadas em um nó folha da árvore de decisão. A definição apropriada desse parâmetro auxilia na regularização da árvore, evitando o sobreajuste ao limitar a complexidade da estrutura final da árvore.

2.6.2.5 *Max Features*

O hiperparâmetro *Max Features* regula o número máximo de recursos considerados em cada divisão de nó durante a construção do modelo. A escolha apropriada desse valor é dependente das características dos dados e do problema específico. Assim como em outros hiperparâmetros, valores extremamente altos podem resultar em sobreajuste, enquanto valores muito baixos podem conduzir a uma árvore subajustada. O ajuste adequado do *Max Features* é crucial para equilibrar a capacidade de ajuste e generalização do modelo.

2.6.3 XGBoost

XGBoost é uma implementação escalável e eficiente da estrutura *gradient boosting* de Friedman (2001), sendo a abreviação do pacote *eXtreme Gradient Boosting*. Essa técnica

se destaca por sua capacidade de realizar otimizações avançadas e aprimoramentos na abordagem de *gradient boosting*, contribuindo para um desempenho robusto em tarefas de aprendizado de máquina. Pelo seu desempenho superior, o modelo figura entre os modelos mais utilizados por vencedores de competições no site *Kaggle*.

No contexto do XGBoost, os modelos empregados para a combinação são representados por árvores de decisão de pequeno porte (CHEN; GUESTRIN, 2016), o que substancialmente diminui a probabilidade de o modelo sofrer *overfitting* (DAWOOD, 2019). Segundo Chen e Guestrin (2016), o atributo mais significativo para a prevalência do modelo XGBoost reside em sua escalabilidade e aplicabilidade em diversas situações, graças a inúmeras otimizações de algoritmos implementadas durante o seu desenvolvimento. Esses atributos conferem ao XGBoost uma posição proeminente entre os modelos mais empregados em cenários práticos. Além das características mencionadas, Chen, He et al. (2015) e Dhaliwal, Nahid e Abbas (2018) ressaltam diversas vantagens associadas ao uso do XGBoost, as quais incluem:

- Capacidade para Processamento Paralelo: O algoritmo demonstra habilidade para realizar processamento paralelo, explorando todos os núcleos disponíveis na máquina em que é executado, tornando-se altamente eficaz para o pré-processamento de dados em larga escala.
- Compatibilidade com Diversos Tipos de Dados: O XGBoost aceita uma variedade de tipos de dados como entrada, podendo ser uma matriz densa ou esparsa, além de oferecer suporte para arquivos locais.
- Customização do Algoritmo: Sua natureza customizável permite a definição de funções de objetivo e avaliação personalizadas.
- Execução em Diversas Plataformas: Pode ser executado em diferentes plataformas, incluindo AWS, Azure, Alibaba, entre outras.
- Suporte para Múltiplas Linguagens de Programação: Oferece suporte para diversas linguagens de programação.
- Transformação de Aprendiz Fraco em Forte: Possui a capacidade de converter um aprendiz fraco em forte por meio da otimização para cada árvore implementada no sistema.
- Validação Cruzada Interna: Incorpora a validação cruzada como um recurso interno, facilitando a avaliação do desempenho do modelo.
- Regularização contra *Overfitting*: Apresenta mecanismos de regularização que auxiliam na prevenção de problemas de *overfitting*.

- Eficiência no Tratamento de Valores Ausentes: Está bem preparado para identificar e lidar eficientemente com valores ausentes ou dados faltantes, contribuindo para a robustez do modelo.

Em sua tese, [Bessa \(2021\)](#), apresenta à Escola de Economia da Fundação Getúlio Vargas (FGV EESP), uma pesquisa com o objetivo de aprimorar a precisão das previsões de vendas em termos de quantidade no setor da moda. Após explorar distintos métodos de tratamento de sazonalidade e seleção de variáveis, identificou-se que, para a categoria específica de produto "Blusa de mulher", a abordagem mais eficaz foi o uso do XGBoost. Este modelo, quando combinado com tratamento sazonal por ajuste de curva e seleção de variáveis por RFA, destacou-se como a escolha mais acertada, avaliada por meio de seis métricas de precisão.

2.7 Métricas de Avaliação em Modelos de Regressão

As métricas de avaliação em modelos de regressão desempenham um papel crucial na avaliação da eficácia e qualidade desses modelos. Elas proporcionam uma medida quantitativa de quão bem um modelo consegue prever as saídas corretas para um determinado conjunto de dados de entrada. Dentre as métricas mais comuns, incluem-se o erro médio absoluto (MAE), a raiz do erro quadrático médio (RMSE) e o erro percentual médio absoluto (MAPE).

A avaliação por meio dessas métricas é fundamental para a seleção da melhor solução entre diversas alternativas e para otimizar o algoritmo escolhido, ajustando seus parâmetros de acordo com o conjunto de dados em questão. Em resumo, as métricas de avaliação de modelos de regressão possibilitam a avaliação, comparação e refinamento de diferentes modelos, aprimorando sua eficácia e qualidade na previsão de resultados em novos conjuntos de dados.

No contexto deste trabalho, destacam-se as métricas de erro médio absoluto, erro quadrático médio e erro percentual absoluto, utilizadas para analisar e avaliar os modelos de aprendizado de máquina desenvolvidos. Essas métricas são essenciais para uma compreensão aprofundada do desempenho dos modelos e para orientar ajustes que visam aperfeiçoar suas capacidades preditivas.

- **MAE - Erro Médio Absoluto**

O erro médio absoluto (MAE) destaca-se como uma métrica amplamente empregada na avaliação de modelos de regressão, conforme abordado no livro "Forecasting: Principles and Practice" de [Rob J Hyndman e Athanasopoulos \(2018\)](#). Essa métrica realiza o cálculo da média das diferenças entre as previsões geradas pelo modelo e os dados reais. Seu valor

é obtido por meio de da soma das diferenças absolutas entre cada previsão do modelo (\hat{y}_i) e a saída correta (y_i), dividindo-se então pelo número de amostras no conjunto de dados de teste (n), como expresso na seguinte equação:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.1)$$

Onde:

- n é o número de amostras;
- y_i representa o valor real do i -ésimo exemplo;
- \hat{y}_i denota a previsão do modelo para o i -ésimo exemplo.

O MAE é uma medida direta da eficácia do modelo, sendo que valores menores indicam uma maior eficácia. Além disso, é uma métrica de fácil interpretação, uma vez que está na mesma unidade da variável de saída. Em comparação com outras formas de avaliação, como o erro médio quadrático, o MAE é menos sensível a valores discrepantes (*outliers*).

• RMSE - Raiz do Erro Quadrático Médio

A Raiz do Erro Quadrático Médio (RMSE) é uma métrica amplamente utilizada na avaliação de modelos de regressão, fornecendo uma medida da dispersão média entre as previsões do modelo e os valores reais observados. A métrica apresenta como uma de suas principais vantagens o fato de que os valores são relativos aos dados utilizados para o cálculo, tornando o método de fácil aplicação para diversos casos (CHAI; DRAXLER, 2014).

A fórmula para o cálculo da RMSE é expressa por:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.2)$$

Onde:

- n é o número de amostras;
- y_i representa o valor real do i -ésimo exemplo;
- \hat{y}_i denota a previsão do modelo para o i -ésimo exemplo.

Uma das técnicas mais utilizadas para reduzir o erro de uma previsão é a otimização de parâmetros, conforme discutido por [Silveira \(2019\)](#). [Bergstra e al. \(2011\)](#) demonstram diversas metodologias que podem ser empregadas para otimizar os parâmetros de um modelo, incluindo a perspectiva utilizada no desenvolvimento deste trabalho. Essa abordagem envolve determinar um espaço amostral para os parâmetros do modelo com base em estudos divulgados e otimizar cada parâmetro dentro desse espaço, uma perspectiva simples, mas eficaz. Vale ressaltar que a otimização requer atenção especial para evitar problemas de overfitting ([YING, 2018](#)).

• MAPE - Erro Percentual Médio Absoluto

O Erro Médio Percentual Absoluto (MAPE) é uma técnica empregada na avaliação de modelos de regressão, medindo o erro médio absoluto em termos percentuais. Essa métrica calcula a diferença absoluta entre cada valor previsto e o valor observado correspondente. Posteriormente, essas diferenças absolutas são normalizadas pelos valores observados e multiplicadas por 100 para obter a porcentagem de erro absoluto para cada observação ([HYNDMAN, Rob J; ATHANASOPOULOS, 2018](#)).

A fórmula do MAPE é expressa por:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (2.3)$$

Onde:

- n é o número de amostras;
- y_i representa o valor real do i -ésimo exemplo;
- \hat{y}_i denota a predição do modelo para o i -ésimo exemplo.

Assim, o valor final do MAPE é obtido calculando a média das porcentagens de erro absoluto para todas as observações. Quanto menor o valor do MAPE, melhor a eficácia do modelo de regressão, indicando que as previsões estão mais próximas dos valores reais.

2.8 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) representam um conjunto de modelos computacionais constituídos por unidades de processamento interconectadas, conhecidas como neurônios artificiais. Estes neurônios são capazes de aprender a partir de um conjunto de dados, ajustando seus pesos sinápticos para aprimorar a eficácia do modelo.

A estrutura dessas redes é composta por camadas de neurônios, começando pela camada de entrada, que recebe os dados a serem processados. As camadas intermediárias, chamadas de ocultas, realizam o processamento dos dados para gerar uma saída. A camada final da rede, denominada de saída, é responsável por fornecer a resposta final do modelo. Segundo [Goodfellow, Bengio e Courville \(2016\)](#), cada camada oculta da rede neural é composta por vetores de valores, sendo que a dimensão dessas camadas determina a largura do modelo. Cada elemento desses vetores pode ser entendido como tendo uma função semelhante à de um neurônio. A organização dessas camadas é ilustrada na Figura 4.

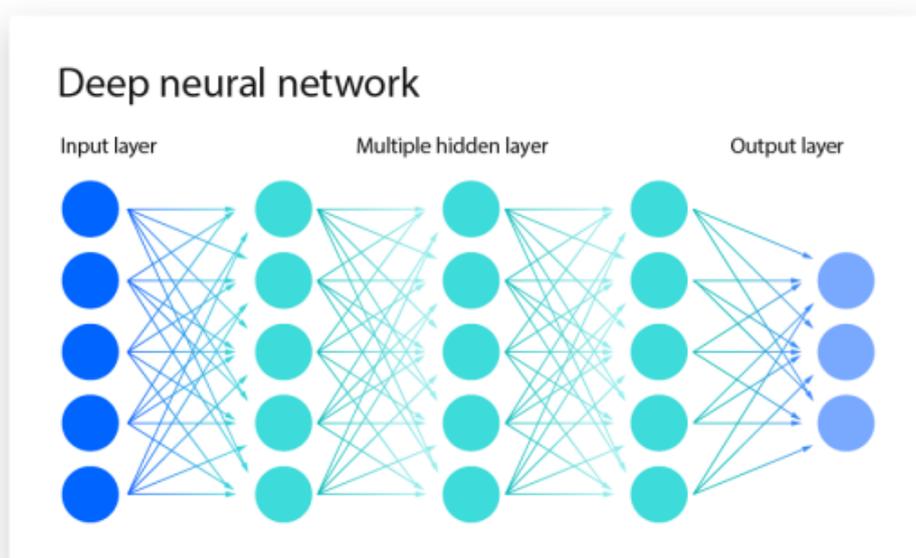


Figura 4 – Rede Neural Artificial. Fonte: [Ultralytics \(2024\)](#).

Na dissertação de doutorado intitulada “Ensaio sobre Previsão de Vendas no Varejo de Moda” conduzida por [Bessa \(2021\)](#), com base nos resultados obtidos, conclui-se que a incorporação de tratamento sazonal não é aconselhável para modelos de árvores de decisão. Este procedimento não proporcionou melhorias significativas em termos de precisão. No entanto, destaca-se a recomendação de aplicar o tratamento sazonal exclusivamente nos modelos fundamentados em redes neurais, mais especificamente nos algoritmos *Convolutional Neural Network* (CNN) e *Long Short-Term Memory* (LSTM).

2.8.1 Rede Neural de Propagação Direta

As Redes Neurais de Propagação Direta (também conhecidas como *feedforward neural networks*) são um tipo de arquitetura de rede neural onde a informação se move em uma única direção, da camada de entrada para a camada de saída ([FLECK et al., 2016](#)). Esse tipo de rede é fundamental no campo de aprendizado de máquina e processamento de dados.

A Figura 4 ilustra a estrutura típica de uma Rede Neural de Propagação Direta, composta por camadas de entrada, camadas ocultas e uma camada de saída. Cada seta representa uma conexão ponderada entre as unidades das camadas adjacentes, e os nós representam neurônios que processam e transferem a informação.

2.8.2 Backpropagation

O algoritmo de Backpropagation (retropropagação do erro) é um método fundamental no treinamento de redes neurais de propagação direta, responsável por ajustar os pesos das conexões sinápticas de forma a minimizar o erro da rede durante o processo de aprendizagem.

Principais características do Backpropagation:

- **Princípio de Funcionamento:** O algoritmo opera por meio de um processo iterativo de dois passos principais - a propagação forward (para frente) e a propagação backward (para trás). Na etapa forward, os dados de entrada são processados pela rede, gerando uma saída. Na etapa backward, o erro dessa saída é calculado e propagado de volta por meio de da rede, ajustando os pesos das conexões.
- **Cálculo do Gradiente do Erro:** O backpropagation utiliza a regra da cadeia do cálculo diferencial para calcular o gradiente do erro em relação a cada peso da rede. Esse gradiente indica a direção e a magnitude da variação necessária para reduzir o erro da rede.
- **Função de Custo:** Geralmente, usa-se uma função de custo como o Erro Quadrático Médio (MSE - Mean Squared Error) para quantificar a diferença entre a saída prevista e a saída esperada. O objetivo do backpropagation é minimizar esta função de custo.
- **Atualização de Pesos:** Os pesos são atualizados proporcionalmente ao gradiente do erro, multiplicado por uma taxa de aprendizado (learning rate). Essa taxa determina o tamanho dos passos de ajuste dos pesos durante o treinamento.
- **Desafios e Limitações:**
 - Problema do Desvanecimento do Gradiente: Em redes profundas, o gradiente pode se tornar muito pequeno, dificultando o aprendizado nas camadas mais próximas da entrada.
 - Sensibilidade à Inicialização dos Pesos: Os resultados do treinamento podem variar significativamente dependendo dos pesos iniciais.

Exemplo matemático simplificado da atualização de pesos:

$$\Delta w = -\eta \cdot \frac{\partial E}{\partial w} \quad (2.4)$$

Onde:

- Δw representa a variação do peso
- η (eta) é a taxa de aprendizado
- $\frac{\partial E}{\partial w}$ é o gradiente do erro em relação ao peso

2.8.3 Treinamento da Rede Neural

O treinamento de uma Rede Neural de Propagação Direta envolve a apresentação iterativa dos dados de treinamento à rede, ajustando os pesos das conexões para minimizar uma função de perda. Este processo é geralmente realizado usando um algoritmo de otimização. É importante conhecer e definir alguns parâmetros de treinamento antes de executá-lo para se ter um bom resultado e os principais são discutidos a seguir.

2.8.3.1 Batch Size

O *batch size* refere-se ao número de exemplos de treinamento utilizados em cada iteração do algoritmo de treinamento. Um *batch* é uma subdivisão do conjunto de dados total. O tamanho do *batch* pode afetar a velocidade e a estabilidade do treinamento, sendo escolhido com base na disponibilidade de recursos computacionais.

2.8.3.2 Taxa de Aprendizado

A taxa de aprendizado (*learning rate*) é um hiperparâmetro crucial que determina o tamanho dos passos dados na direção oposta ao gradiente durante o treinamento. Uma taxa de aprendizado adequada é vital para garantir a convergência do modelo. A escolha desse valor deve ser cuidadosa, pois taxas muito altas podem resultar em oscilações e taxas muito baixas podem levar a um treinamento lento ou estagnado.

2.8.3.3 Função de Ativação

As funções de ativação são utilizadas nos neurônios para introduzir não-linearidade nas redes neurais. Exemplos comuns incluem a função sigmoide, a tangente hiperbólica (tanh) e a unidade linear retificada (ReLU). A escolha da função de ativação depende da natureza do problema e da arquitetura da rede.

2.8.3.4 Camadas

As redes neurais consistem em camadas, sendo a camada de entrada responsável por receber os dados, as camadas ocultas realizam transformações intermediárias, e a camada de saída fornece a resposta da rede. A arquitetura, ou seja, o número e a disposição das camadas, é um aspecto crucial do projeto da rede neural.

2.8.3.5 Algoritmo de Otimização

O algoritmo de otimização é responsável por ajustar os pesos das conexões da rede de maneira a minimizar a função de perda. Gradiente Descendente, Adam e RMSprop são exemplos de algoritmos populares. A escolha do algoritmo pode impactar a velocidade de convergência e a qualidade dos resultados alcançados.

2.9 Validação Cruzada

A validação cruzada (*cross validation*) é uma estratégia utilizada para avaliar o desempenho de um modelo preditivo, garantindo que o modelo seja capaz de generalizar bem para novos dados não utilizados no treinamento. A ideia central por trás da validação cruzada é dividir o conjunto de dados em partes distintas, treinar o modelo em algumas dessas partes e avaliá-lo em outras. A técnica se baseia na ideia de repetir o cálculo de treinamento e teste em diferentes subconjuntos ou divisões aleatórias do conjunto de dados original. A validação desse tipo mais utilizada é a *k-fold*, em que o conjunto de dados é dividido em k subconjuntos exclusivos, cada um com o mesmo número de amostras. Em seguida, o modelo é treinado em $k - 1$ partições e testado na partição restante. Esse processo é repetido k vezes, onde cada partição é usada exatamente uma vez como conjunto de teste, conforme visto na Figura 5.

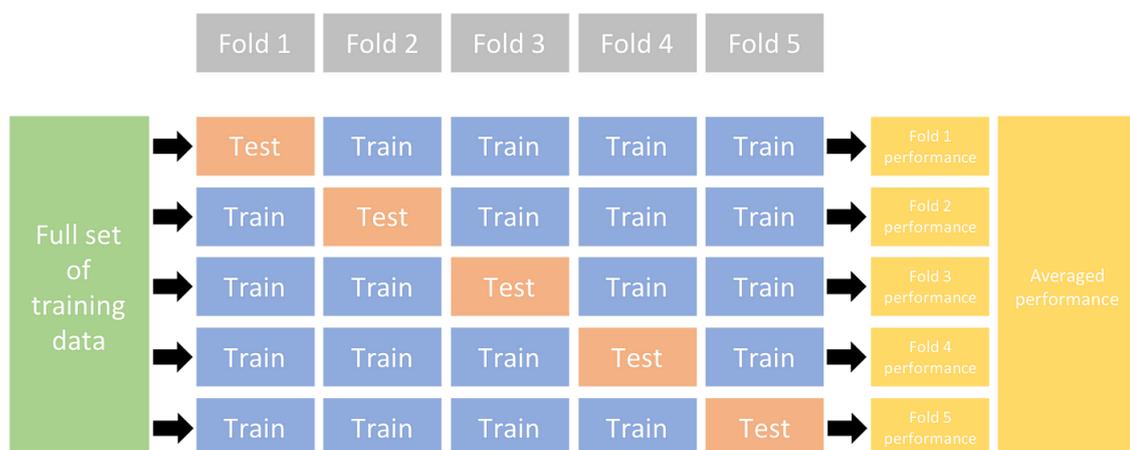


Figura 5 – Validação Cruzada Utilizando *K-Fold*.

Fonte: (ULTRALYTICS, 2024).

A métrica de desempenho derivada da validação cruzada *k-fold* é obtida por meio da média dos valores calculados durante o processo iterativo. Embora este método seja mais computacionalmente exigente, ele evita a perda de dados de treinamento para o conjunto de teste, constituindo uma vantagem significativa, especialmente em cenários com conjuntos de dados de pequeno porte.

2.9.1 *Time Series Split*

A técnica de *Time Series Split* ou Divisão de Séries Temporais representa uma solução robusta para a validação de modelos preditivos aplicados a dados temporais. [Simionato \(2022\)](#) destaca que a aplicação de métodos tradicionais de validação cruzada, como o *K-Fold Cross-Validation*, pode levar a vazamento de dados e comprometimento da integridade temporal do modelo. Diferentemente desses métodos, o *Time Series Split* preserva a ordem cronológica, garantindo que informações futuras não contaminem o conjunto de treinamento.

Para tanto, o *Time Series Split* apresenta as seguintes características fundamentais:

- Os dados de treino sempre precedem os dados de teste.
- A ordem cronológica é rigorosamente preservada.
- Cada divisão simula uma previsão em tempo real, replicando as condições práticas de um modelo em produção.

No processo de *Time Series Split*, o conjunto de dados é dividido progressivamente. Especificamente:

- A primeira divisão utiliza uma janela inicial de treinamento e uma pequena porção de dados para teste.
- Em cada iteração subsequente, a janela de treinamento é expandida, incorporando novas observações.
- Simultaneamente, a janela de teste é deslocada para frente no tempo, mantendo sua independência em relação aos dados de treinamento.

Matematicamente, para um conjunto de dados com n observações e k divisões, o *Time Series Split* pode ser representado da seguinte forma:

$$\text{Tamanho da janela de treinamento} = \left\lfloor \frac{n}{k} \right\rfloor + i \cdot \text{incremento}, \quad (2.5)$$

onde i representa o índice da iteração atual e incremento é o tamanho do passo entre as divisões.

A Figura 6 ilustra visualmente o processo de Time Series Split, destacando o crescimento progressivo da janela de treinamento e o deslizamento da janela de teste para o futuro.

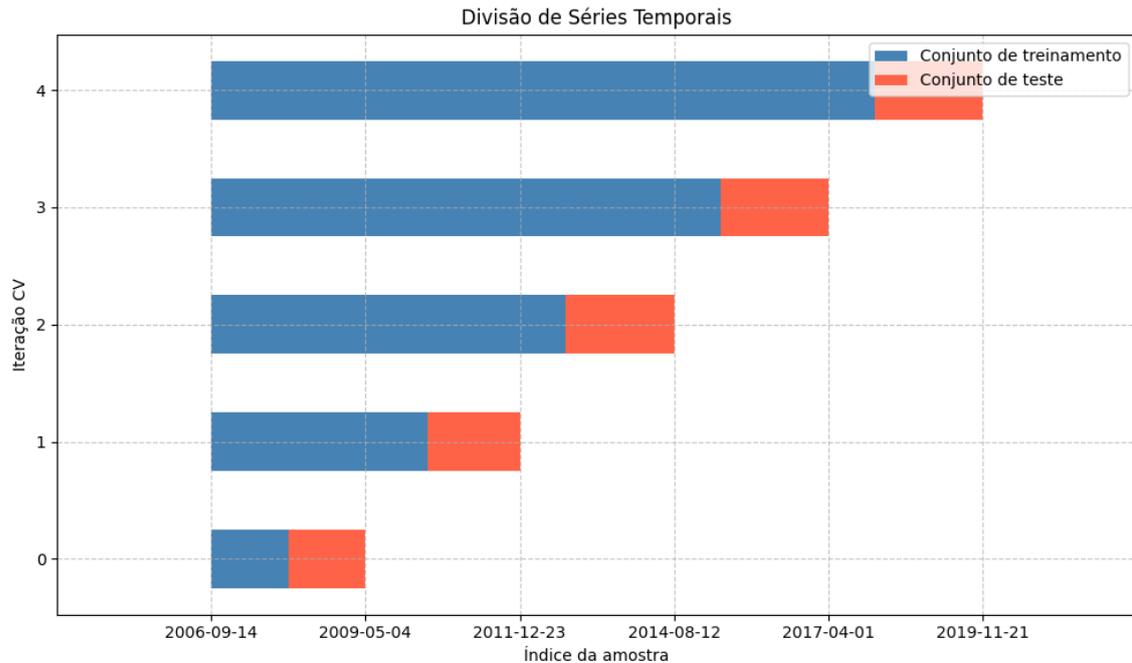


Figura 6 – Divisão de séries temporais utilizando *Time Series Split*.

Fonte: Elaborado pelo autor.

A implementação prática desta abordagem pode ser realizada por meio da classe *Time Series Split* disponível no pacote *scikit-learn* (PEDREGOSA, 2021). Esta ferramenta permite particionar os dados de maneira incremental e respeitando a temporalidade. Dessa forma, o *Time Series Split* torna-se particularmente valioso em tarefas de previsão de séries temporais, onde a dependência temporal e a estrutura cronológica são cruciais para a modelagem e avaliação precisas dos modelos.

3 Desenvolvimento

3.1 Base de Dados

Geralmente, *datasets* de *e-commerce* são proprietários e, conseqüentemente, difíceis de encontrar em dados públicos. No entanto, o *UCI Machine Learning Repository* disponibilizou um *dataset* contendo transações reais dos anos de 2010 e 2011. Esse *dataset* é mantido no site deles e pode ser encontrado pelo título “*Online Retail*”.

Este é um conjunto de dados transacionais que contém todas as transações ocorridas entre 01/12/2010 e 09/12/2011 para um varejo online baseado no Reino Unido. A empresa vende principalmente presentes únicos para diversas ocasiões. Muitos clientes da empresa são atacadistas. De acordo com o UCI Machine Learning Repository, esses dados foram disponibilizados pelo Dr. Daqing Chen, Diretor do grupo Public Analytics.¹

3.1.1 Explorando as variáveis do banco de dados

Na Figura 7, é apresentado um recorte do banco de dados utilizado neste estudo. A base de dados é composta por registros transacionais extraídos de uma plataforma de varejo online, contendo informações como data da compra, quantidade de itens vendidos, preço unitário, identificação do produto e do cliente, entre outros atributos relevantes para a modelagem preditiva. Esse conjunto de dados serviu como base para as etapas de pré-processamento e aplicação dos modelos de *machine learning*, permitindo a análise de padrões de consumo ao longo do tempo e a geração de previsões de demanda mais precisas.

A	B	C	D	E	F	G	H
InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 08:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 08:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 08:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 08:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 08:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 08:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 08:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 08:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 08:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 08:34	1.69	13047	United Kingdom

Figura 7 – Recorte do banco de dados utilizado no estudo, com atributos relevantes para a análise preditiva. Fonte: Elaborado pelo autor.

A seguir, serão detalhadas as principais variáveis do banco de dados:

- **Coluna A – InvoiceNo**

¹ Contato: chend@lsbu.ac.uk, Escola de Engenharia, London South Bank University, Londres SE1 0AA, Reino Unido.

- É o número da fatura ou remessa. No processo de compras, um conjunto de itens adquiridos é tratado como uma remessa ou fatura. Nesta fatura, podem haver um ou mais itens comprados. Cada número representa um pedido/compra realizados, similar a um número correspondente a um carrinho de compras em um supermercado. Este campo é crucial para a identificação de transações únicas e para a análise de padrões de compra ao longo do tempo.

- **Coluna B – StockCode**

- Trata-se da identificação numérica para cada um dos itens. O StockCode geralmente consiste em um número de 5 dígitos, podendo incluir letras para itens da mesma família, permitindo a identificação única. Esta variável é essencial para a análise do desempenho de produtos específicos e para a categorização de itens.

- **Coluna C – Description**

- É a descrição completa com o nome dos itens e suas respectivas características, como cor, tamanho, capacidade, entre outras dimensões. Esta coluna é útil para compreender a variedade de produtos oferecidos e pode ser utilizada para análise de texto e mineração de dados para identificar tendências em descrições de produtos.

- **Coluna D – Quantity**

- Quantidade comprada daquele item específico. Esta variável é fundamental para calcular o volume de vendas e entender a demanda por diferentes produtos. Pode ser usada para identificar itens mais populares e sazonalidades nas vendas.

- **Coluna E – InvoiceDate**

- Data e horário da realização da transação InvoiceNo. Esta informação é crítica para análises temporais, como a identificação de picos de vendas, tendências sazonais, e comportamento de compra ao longo do tempo.

- **Coluna F – UnitPrice**

- Valor unitário do item. Esta variável permite a análise de preços, margens de lucro e pode ser utilizada para identificar estratégias de precificação eficazes.

- **Coluna G – CustomerID**

- Trata-se de um representa o identificador único de cada cliente. Ele serve para vincular transações ou compras a clientes específicos.

- **Coluna H – Country**

- País onde a transação ocorreu. Cada linha representa uma transação, permitindo a análise geográfica das vendas. Esta informação é valiosa para entender a distribuição geográfica dos clientes e pode ser utilizada para estratégias de marketing segmentadas.

3.1.2 Importação e Manipulação do Banco de Dados

A preparação do conjunto de dados envolveu múltiplas etapas de limpeza e processamento. O objetivo dessas ações foi garantir a qualidade e a integridade das informações, elementos essenciais para análises confiáveis e interpretações precisas. Para realizar tal ação, utilizou-se um conjunto robusto de bibliotecas *Python* no ambiente *Google Colaboratory*.

Para manipulação e análise dos dados, foram utilizadas as bibliotecas fundamentais *Pandas* e *NumPy*. O *Pandas* foi essencial para importar o conjunto de dados no formato CSV, além de fornecer estruturas de dados eficientes como o *DataFrame*, enquanto o *NumPy* ofereceu suporte para operações matemáticas e manipulação de arrays. A visualização dos dados foi realizada por meio de um conjunto complementar de bibliotecas: *Matplotlib*, *Seaborn* e *Plotly*. O *Matplotlib* serviu como base para criação de gráficos, o *Seaborn* proporcionou visualizações estatísticas mais sofisticadas, e o *Plotly*, por meio de dos módulos *graph_objects* e *make_subplots*, permitiu a criação de gráficos interativos e complexos.

Inicialmente, realizou-se uma verificação detalhada para identificar valores ausentes nas colunas do conjunto de dados. Essa etapa preliminar forneceu uma visão geral da integridade do banco, permitindo traçar um plano para tratar inconsistências. Em seguida, a coluna *InvoiceDate* foi convertida para o formato *datetime*, facilitando análises temporais. Além disso, essa coluna foi usada como índice do *DataFrame*, possibilitando uma organização mais eficiente dos dados e análises temporais precisas.

Durante o processo, identificou-se que a coluna *CustomerID* não possuía relevância direta para os objetivos da análise. Assim, optou-se pela sua remoção, simplificando o banco e reduzindo possíveis distrações na modelagem futura.

Outro aspecto crítico foi a definição de critérios rigorosos para a seleção de registros válidos. Transações com valores de *UnitPrice* menores ou iguais a zero, bem como aquelas com *Quantity* abaixo de um, foram eliminadas. Tais valores representam inconsistências que poderiam comprometer a análise estatística e preditiva.

As descrições dos produtos também receberam atenção especial. Descrições ausentes foram substituídas pelo termo genérico “UNKNOWN ITEM”. Além disso, itens classificados como não relacionados a vendas, tais como “DOTCOM POSTAGE” e “AMAZON FEE”, foram removidos. Essa exclusão garantiu que o foco permanecesse nas transações efetivas de produtos.

3.1.3 Motivo da Exclusão

Os itens removidos, identificados pelos códigos DOT, M, POST, AMAZONFEE, B e C2, foram excluídos por sua natureza não comercial. Esses registros correspondem a custos operacionais, ajustes financeiros e despesas administrativas, como detalhado abaixo:

- **DOTCOM POSTAGE/POSTAGE**: Custos relacionados a serviços postais.
- **Manual (M)**: Documentação técnica ou suporte administrativo.
- **AMAZON FEE**: Taxas cobradas pela plataforma Amazon.
- **Adjust bad debt (B)**: Ajustes financeiros de débitos.
- **CARRIAGE (C2)**: Custos de transporte.

A exclusão desses itens justifica-se por dois motivos principais. Primeiramente, por sua natureza não comercial, esses registros não representam transações de vendas reais e, portanto, não contribuem para a análise do comportamento de compra dos consumidores. Em segundo lugar, a inclusão desses registros poderia comprometer a qualidade da análise exploratória e dos modelos preditivos subsequentes. Por exemplo, valores inflacionados artificialmente distorceriam as métricas de desempenho comercial, dificultando a identificação de tendências reais.

Portanto, garantir a limpeza e a consistência do banco de dados foi uma etapa crucial para assegurar que os *insights* gerados sejam precisos e aplicáveis.

3.1.4 Análise Exploratória dos Dados

3.1.4.1 Países presentes no conjunto de dados

No banco de dados, tem-se a presença de 38 nações, sendo elas: Austrália, Áustria, Bahrain, Bélgica, Brasil, Canadá, Ilhas do Canal, Chipre, República Checa, Dinamarca, Irlanda, Comunidade Europeia, Finlândia, França, Alemanha, Grécia, Hong Kong, Islândia, Israel, Itália, Japão, Líbano, Lituânia, Malta, Países Baixos, Noruega, Polônia, Portugal, República da África do Sul, Arábia Saudita, Singapura, Suécia, Espanha, Suíça, Emirados Árabes Unidos, Reino Unido, Estados Unidos e conjunto de países não especificados (*Unspecified*). Na Tabela 1 pode-se analisar como a distribuição de vendas ocorreu entre os 38 países, destacando o valor total de vendas.

Tabela 1 – Distribuição de vendas por país em Libras Esterlinas (£) e percentual de vendas.
Fonte: Elaborado pelo autor.

País	VdaTotal (£)	Percentual de Vendas
United Kingdom	8.748.475,52	85,1653
Netherlands	283.889,34	2,7636
EIRE	271.164,30	2,6398
Germany	205.569,89	2,0012
France	184.582,74	1,7969
Australia	138.171,31	1,3451
Spain	55.725,11	0,5425
Switzerland	53.087,90	0,5168
Japan	37.416,37	0,3642
Belgium	36.927,34	0,3595
Sweden	36.839,33	0,3586
Norway	32.454,64	0,3159
Portugal	27.015,16	0,2630
Channel Islands	20.007,44	0,1948
Finland	18.344,88	0,1786
Denmark	18.211,34	0,1773
Italy	15.820,24	0,1540
Cyprus	13.254,69	0,1290
Hong Kong	9952,99	0,0969
Singapore	9120,39	0,0888
Austria	8742,68	0,0851
Israel	8135,26	0,0792
Poland	6974,65	0,0679
Unspecified	4749,79	0,0462
Greece	4425,52	0,0431
Iceland	4310,00	0,0420
USA	3580,39	0,0349
Canada	3115,44	0,0303
Malta	2070,59	0,0202
United Arab Emirates	1864,78	0,0182
Lebanon	1693,88	0,0165
Lithuania	1661,06	0,0162
European Community	1159,25	0,0113
Brazil	1143,60	0,0111
RSA	1002,31	0,0098
Czech Republic	786,74	0,0077
Bahrain	754,14	0,0073
Saudi Arabia	145,92	0,0014

Entre dezembro de 2010 e dezembro de 2011, foi registrada uma movimentação financeira total de £10.272.346,92. A Figura 9 apresenta a distribuição das transações por país. Neste contexto, uma transação é definida como cada ocasião de venda registrada no conjunto de dados, independentemente da quantidade de itens incluídos em uma única

fatura (*InvoiceNo*). Por exemplo, se uma fatura contém 5 produtos diferentes, isso é contabilizado como 5 transações distintas, mesmo que pertençam ao mesmo número de fatura.

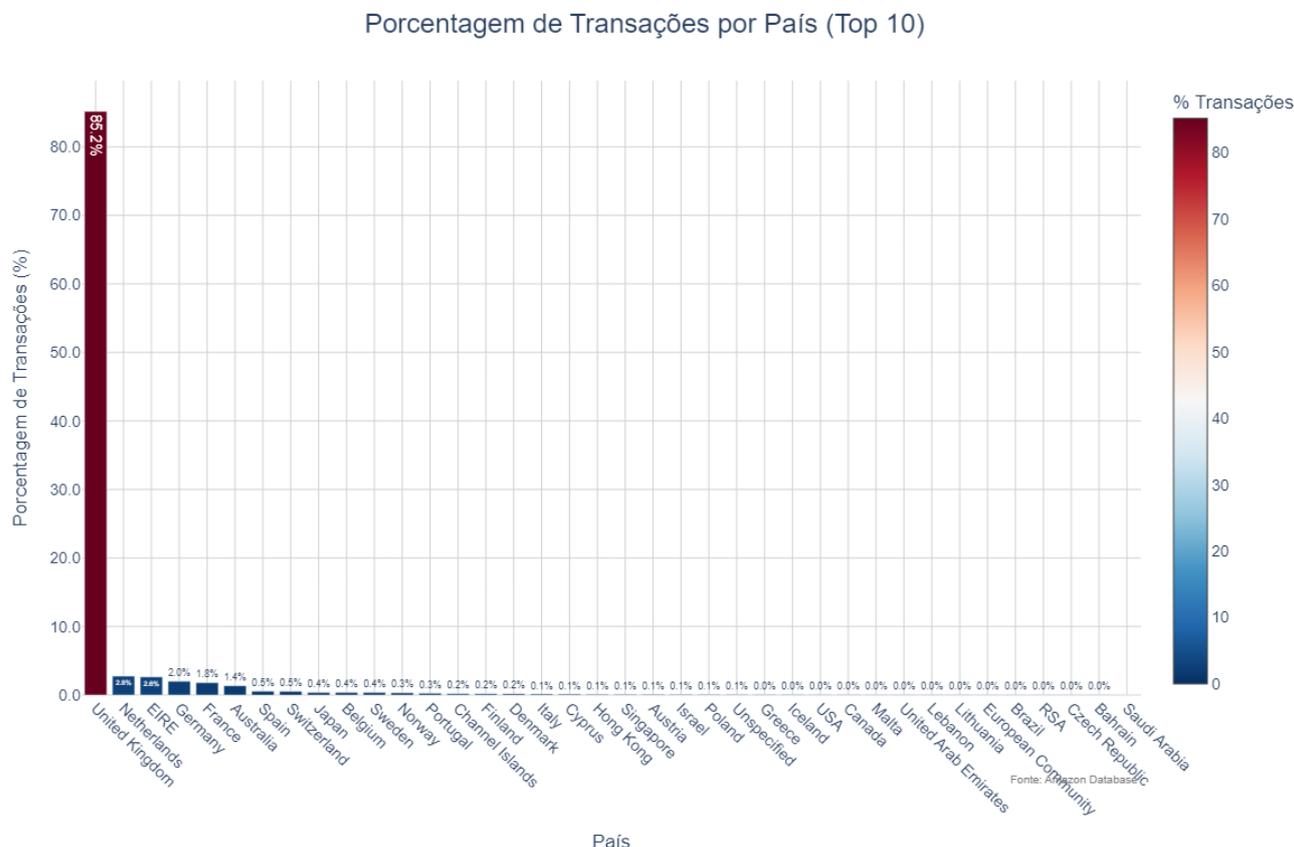


Figura 8 – Distribuição geográfica das transações comerciais por país no período analisado
 Fonte: Elaborado pelo autor.

Analisando a Tabela 1, observa-se que o Reino Unido concentra 85,17% do valor total das vendas, evidenciando sua centralidade nas transações realizadas. Essa dominância justifica a escolha por direcionar a análise detalhada exclusivamente ao Reino Unido.

Em contrapartida, os demais países apresentam contribuições significativamente inferiores: os Países Baixos com 2,76%, Irlanda com 2,64%, Alemanha com 2,00% e França com 1,80%. Esses dados indicam que a inclusão de pilares referentes a esses países tem impacto limitado para os objetivos deste estudo.

Adicionalmente, a análise priorizou o montante total de vendas, sem considerar diretamente a quantidade de peças vendidas. Essa escolha fundamenta-se na expectativa de que a quantidade de peças não agregaria *insights* relevantes para os resultados desejados.

Por fim, a alta densidade de dados do Reino Unido favorece a aplicação de modelos preditivos e análises robustas. Em contrapartida, a menor densidade nos demais países

umenta a probabilidade de ruído nos algoritmos, o que poderia comprometer a eficiência das análises realizadas.

3.2 Metodologia de Análise dos Dados

3.2.1 Filtragem e Seleção de Dados

O conjunto de dados utilizado neste estudo contém 3.806 itens distintos, categorizados em diferentes grupos de produtos, e registra aproximadamente 5.577.150 transações. Conforme a Figura 9, observa-se a distribuição de transações por país, evidenciando a dominância do Reino Unido. A análise foi delimitada ao mercado do **Reino Unido**, dada sua representatividade estatística no conjunto de dados.

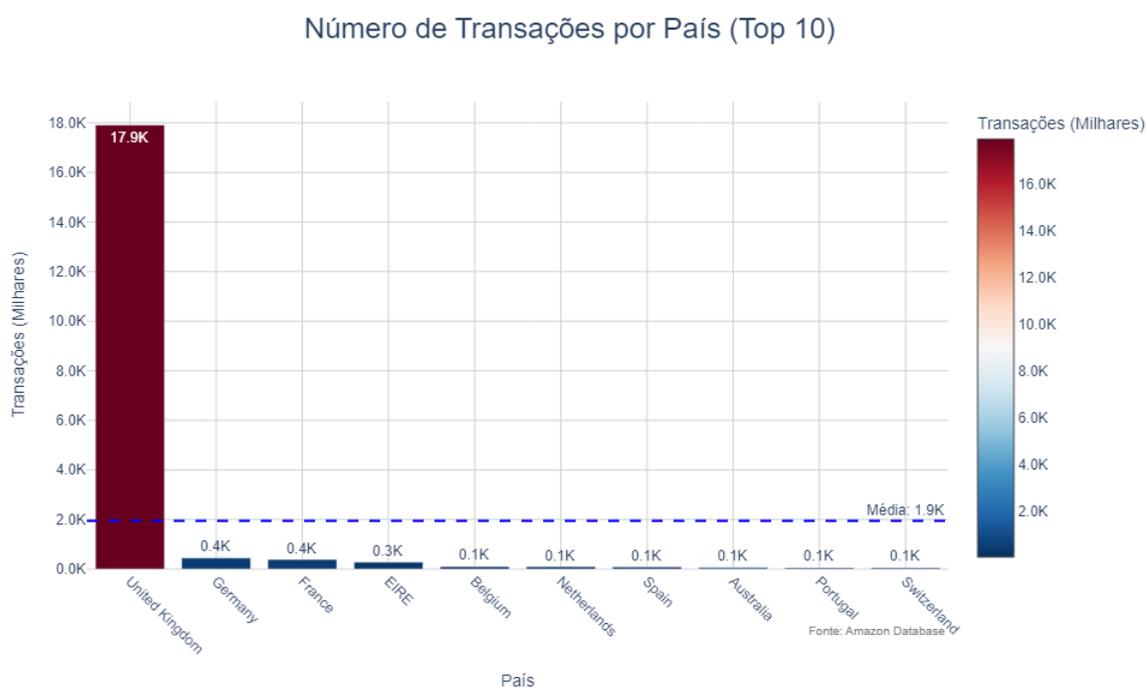


Figura 9 – Distribuição geográfica das transações comerciais realizadas
Fonte: Elaborado pelo autor.

Para alcançar maior consistência nos resultados, transações de outras localidades foram excluídas. Essa decisão baseou-se em três pilares: a robustez dos dados do Reino Unido, que assegura maior confiabilidade às conclusões; a mitigação de possíveis vieses causados por variações climáticas e econômicas entre países; e a simplificação do modelo analítico ao reduzir variáveis externas.

3.2.2 Análise Descritiva dos Dados

A exploração inicial dos dados revelou uma significativa concentração de vendas em poucos produtos. De acordo com os dados apresentados na Figura 10, “PAPER CRAFT, LITTLE BIRDIE” gerou £168.469,60 em vendas, ocupando a liderança no *ranking* de itens comercializados. Em seguida, “REGENCY CAKESTAND 3 TIER” e “WHITE HANGING HEART T-LIGHT HOLDER” destacaram-se com £142.273,29 e £100.497,72, respectivamente.

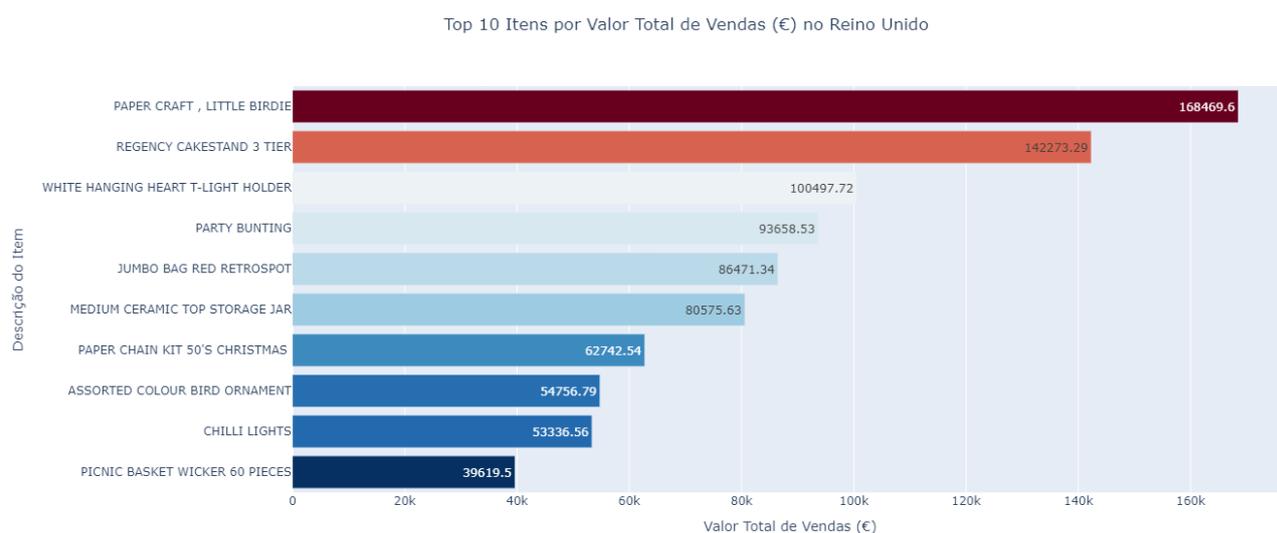


Figura 10 – Ranking dos 10 itens mais vendidos por valor total (£). Fonte: Elaborado pelo autor.

A análise temporal mensal, conforme pode-se analisar na Figura 11, mostrou um padrão de crescimento com picos em outubro e novembro de 2011, períodos em que o volume de transações superou £1.000.000,00. Esses dados indicam tendências sazonais claras, que são essenciais para o desenvolvimento de estratégias comerciais.

3.2.3 Análise da Curva ABC

O princípio de Pareto, idealizado por Vilfredo Pareto em 1897, sugere que 20% dos itens são responsáveis por 80% dos resultados (VIANA, 2000). Mediante essa teoria, a *General Electric* realizou uma adaptação do princípio de Pareto à administração de materiais, que foi denominada curva ABC. Esta representa um instrumento que permite identificar itens que justificam atenção e tratamento adequados em seu gerenciamento. A classificação ABC foi aplicada para identificar a relevância financeira de diferentes produtos.



Figura 11 – Padrão de vendas mensal no Reino Unido, evidenciando picos sazonais. Fonte: Elaborado pelo autor.

Conforme a Figura 12, observa-se a curva de Pareto e a distribuição cumulativa de vendas por categoria. Esta análise aponta que cerca de 20% dos itens (Classe A) representam 70% das vendas totais, enquanto os itens da Classe B contribuem com 20% do faturamento. Já os produtos da Classe C, que somam 50% dos itens, representam apenas 10% das vendas. Na curva, as distinções são representadas visualmente pelas áreas sombreadas:

- **Classe A (Área Rosa):** A área rosa na parte inferior da curva representa os itens da Classe A. Estes são os itens de maior valor, que juntos representam 70% das vendas totais. Eles estão localizados na parte mais íngreme da curva, mostrando que um pequeno número de itens contribui significativamente para as vendas.
- **Classe B (Área Bege):** A área bege no meio da curva representa os itens da Classe B. Estes itens têm um impacto moderado nas vendas, contribuindo com 20% do faturamento total. Eles estão localizados na parte intermediária da curva, onde a inclinação é menos acentuada.
- **Classe C (Área Azul Clara):** A área azul clara na parte superior da curva representa os itens da Classe C. Estes são os itens de menor valor, que juntos representam apenas 10% das vendas totais. Eles estão localizados na parte mais plana da curva, mostrando que um grande número de itens contribui pouco para as vendas.

Conforme a Figura 13, nota-se a distribuição percentual dos itens por classe. A Classe C é composta por 2646 itens, representando 66,2% do total, enquanto a Classe B contém 814 itens (20,3%) e a Classe A, 541 itens (13,5%). Essa distribuição reflete a

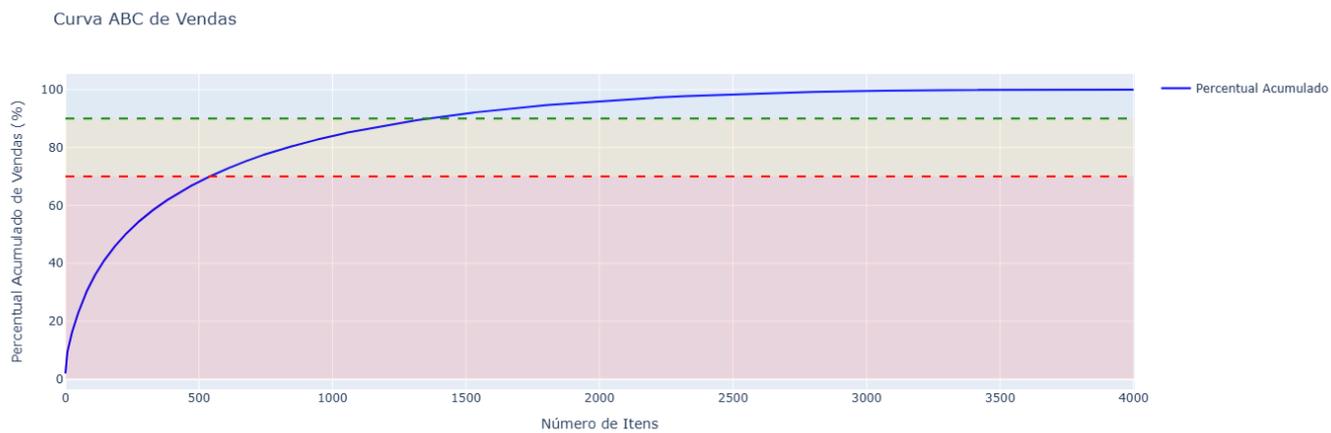


Figura 12 – Distribuição da curva ABC e relevância financeira das classes. Fonte: Elaborado pelo autor.

aplicação do método de Pareto, no qual a maior parte dos itens pertence à Classe C, mas com menor relevância financeira, enquanto a Classe A concentra os itens de maior impacto econômico.

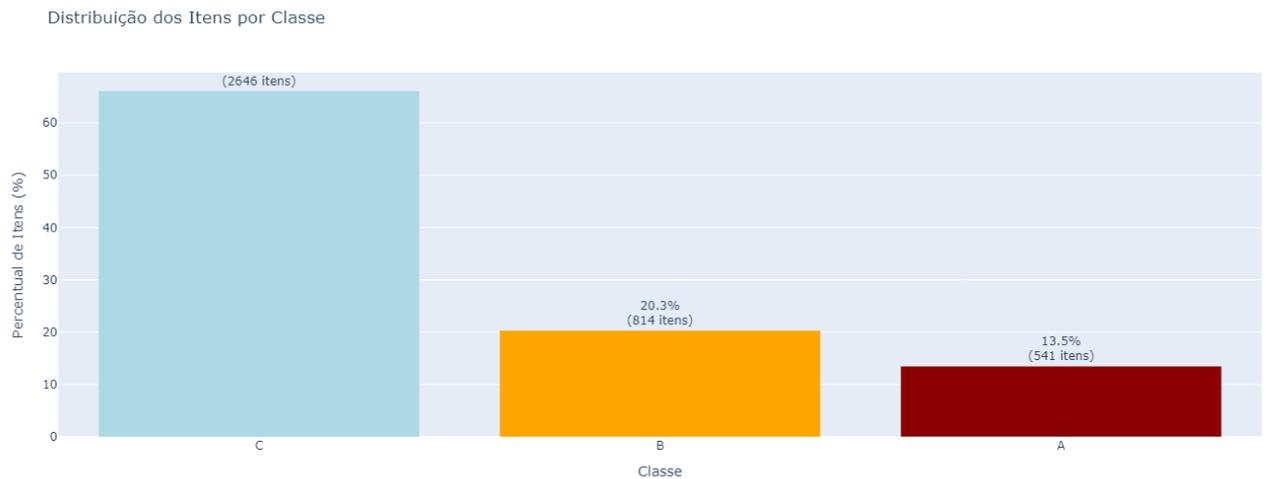


Figura 13 – Distribuição percentual e numérica dos itens classificados em Classe A, B e C, conforme o método ABC. Fonte: Elaborado pelo autor.

Essa segmentação permite priorizar itens críticos para o planejamento de estoques e direcionar estratégias comerciais para produtos com maior impacto financeiro. O foco nos itens da Classe A e, em menor grau, na Classe B, pode otimizar o uso de recursos e aumentar a eficiência das operações logísticas e comerciais.

3.2.4 Análise Temporal Detalhada

O comportamento das vendas ao longo do período de janeiro de 2010 a dezembro de 2011 revelou variações expressivas na demanda. Conforme a Figura 14, visualiza-se a evolução temporal das vendas para os principais itens, “PAPER CRAFT, LITTLE BIRDIE” atingiu £168.469,60 em outubro de 2011, enquanto “MEDIUM CERAMIC TOP STORAGE JAR” alcançou £77.183,60 em janeiro do mesmo ano. Esses picos refletem padrões de consumo relacionados a eventos sazonais.

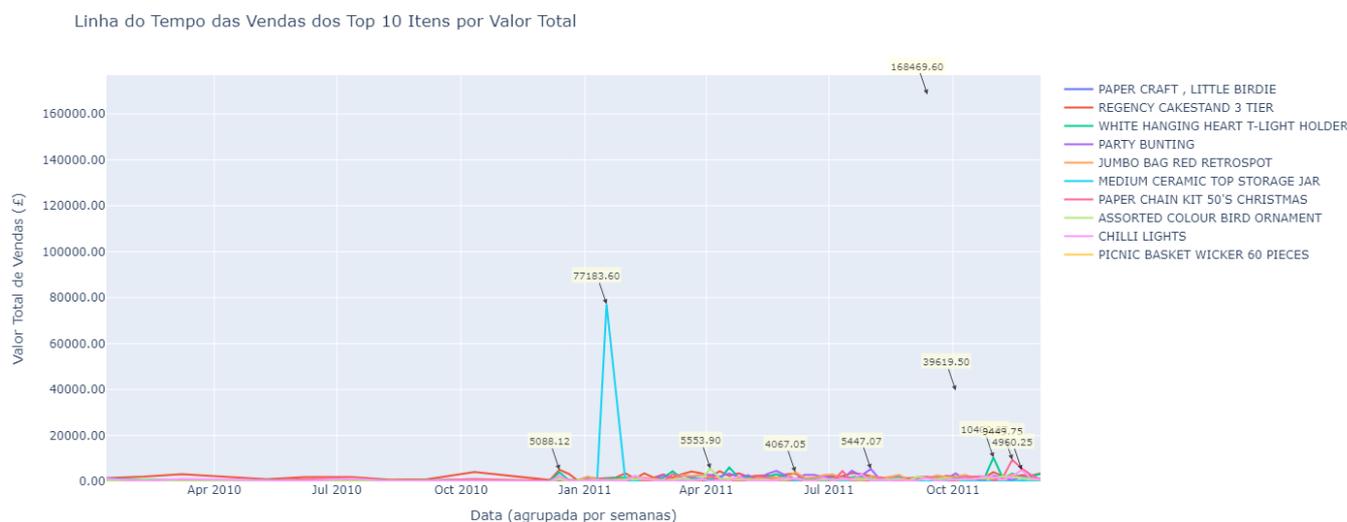


Figura 14 – Distribuição geográfica das transações comerciais realizadas no período analisado. Fonte: Elaborado pelo autor.

A análise semanal, conforme ilustrado na Figura 15, revelou padrões distintos de comportamento ao longo dos anos. Observa-se que o primeiro semestre de 2010 foi caracterizado por estabilidade nas vendas, com valores relativamente baixos e consistentes. Em contraste, o segundo semestre de 2011 apresentou maior volatilidade, evidenciada por picos de vendas expressivos em semanas específicas. Outubro de 2011 destacou-se como o período de maior movimentação, refletindo um aumento acentuado nas transações comerciais. Esses resultados sugerem sazonalidade e possíveis eventos comerciais que impulsionaram as vendas nesse período.

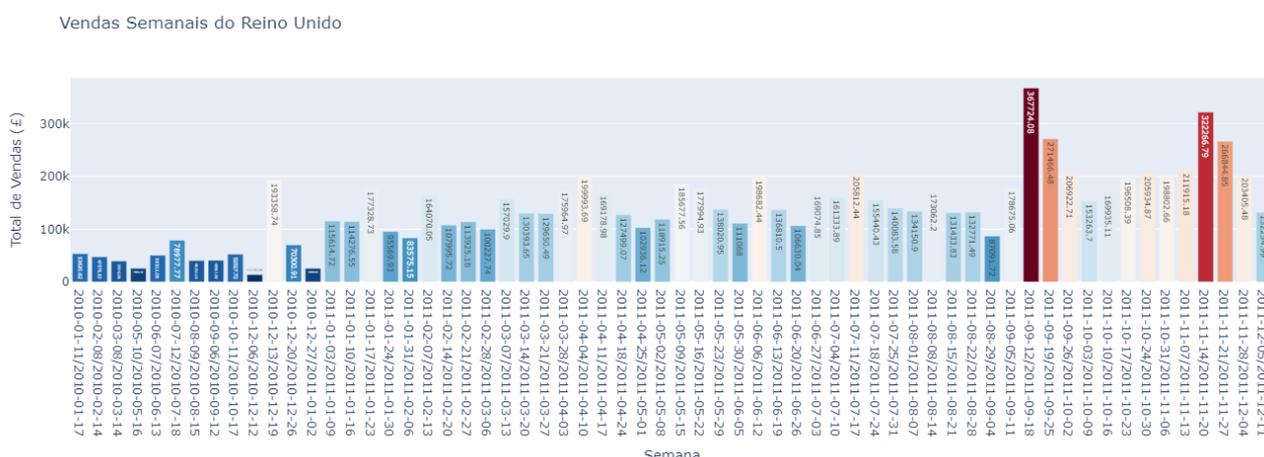


Figura 15 – Total de vendas semanais no Reino Unido (2010-2011). Fonte: Elaborado pelo autor.

A análise mensal, apresentada na Figura 16, complementa os achados semanais, permitindo uma visão consolidada das vendas ao longo do período. Verifica-se que há uma tendência de crescimento nas vendas entre janeiro de 2010 e outubro de 2011, com valores iniciais em torno de £40.000 alcançando picos superiores a £1.000.000 nos últimos meses do período. Destaca-se um aumento significativo no volume de vendas a partir de janeiro de 2011, com particular intensidade nos meses de setembro e outubro de 2011, onde foram registrados os maiores valores do período.

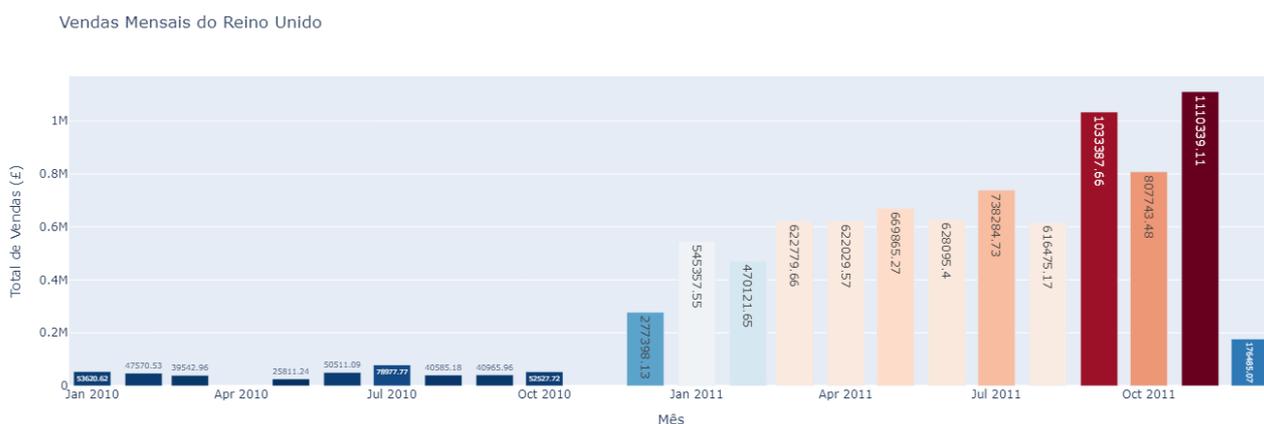


Figura 16 – Total de vendas mensais no Reino Unido (2010-2011). Fonte: Elaborado pelo autor.

3.3 Análise de Preços e *Outliers*

A análise de preços e a identificação de *outliers* desempenham um papel central em estudos de dados transacionais, garantindo a consistência, a comparabilidade e a integridade dos dados. Os preços dos produtos variam devido a fatores como sazonalidade, promoções e flutuações na demanda. Para lidar com esses desafios, foi implementada uma metodologia abrangente que integra o tratamento de preços unitários e a detecção de *outliers*.

3.3.1 Metodologia de Normalização e Detecção de *Outliers*

O processo de tratamento de preços envolveu múltiplas etapas, iniciando com a aplicação do *z-score*, uma técnica estatística que mensura a distância, em desvios padrão, de uma observação em relação à média. Observações com *z-score* superior a 3 foram classificadas como *outliers* e tratadas de forma a reduzir discrepâncias extremas, sem comprometer a distribuição geral dos dados.

Adicionalmente, a escolha do limiar $z = 3$ fundamenta-se na regra empírica dos três sigma, a qual estabelece que, para uma distribuição normal, aproximadamente 99,7% dos dados se encontram no intervalo $(\mu - 3\sigma, \mu + 3\sigma)^2$. Formalmente, o **z-score** é definido pela equação:

$$z = \frac{x - \mu}{\sigma}, \quad (3.1)$$

onde x representa a observação, μ a média e σ o desvio padrão da amostra. Dessa forma, observações que apresentam $|z| > 3$ possuem uma probabilidade extremamente baixa de ocorrência, sendo consideradas discrepantes do comportamento esperado dos dados. Conforme discutido por Curtis et al. (2016), essa abordagem estatisticamente robusta permite identificar criteriosamente apenas os valores que não podem ser atribuídos à variabilidade natural do processo, minimizando a exclusão indevida de dados relevantes.

Além do tratamento de *outliers*, foi implementado um sistema de detecção de promoções baseado em um limiar de 90% do preço base, calculado a partir de uma média móvel de 90 dias. Essa estratégia possibilitou capturar tendências sazonais e identificar períodos promocionais que poderiam impactar significativamente os dados.

A eficácia da metodologia foi avaliada por meio do coeficiente de variação (CV%), que mensura a variabilidade relativa dos dados. A Tabela 2 apresenta os valores de CV% antes e após o tratamento, evidenciando uma redução substancial na variabilidade de diversos itens. Por exemplo, o produto “4 PURPLE FLOCK DINNER CANDLES” apresentou um CV% original de 33,63, reduzindo para apenas 1,09 após o tratamento.

² <https://blog.proffernandamaciel.com.br/regra-empirica/>

Tabela 2 – Comparação do Coeficiente de Variação antes e após o tratamento.

Produto	CV% Original	CV% Tratado
4 PURPLE FLOCK DINNER CANDLES	33,63	1,09
50'S CHRISTMAS GIFT BAG LARGE	30,90	30,78
DOLLY GIRL BEAKER	33,93	33,93
I LOVE LONDON MINI BACKPACK	30,05	30,05
NINE DRAWER OFFICE TIDY	27,20	26,30
OVAL WALL MIRROR DIAMANTE	38,67	33,95
RED SPOT GIFT BAG LARGE	27,70	27,56
SET 2 TEA TOWELS I LOVE LONDON	30,24	29,30
SPACEBOY BABY GIFT SET	20,81	11,37
TOADSTOOL BEDSIDE LIGHT	0,00	0,00

Fonte: Elaborado pelo autor.

Adicionalmente, a análise de correlação demonstrou que o tratamento de preços preservou a estrutura fundamental dos dados. Conforme ilustrado na Figura 17, a correlação entre os preços originais e tratados atingiu 0,98, enquanto a correlação entre preços e quantidade permaneceu praticamente nula (-0,02). Esse resultado indica que as decisões de compra não foram significativamente influenciadas por variações de preço no período analisado, podendo ser atribuídas a fatores como inelasticidade da demanda, fidelidade à marca ou necessidades específicas dos produtos.

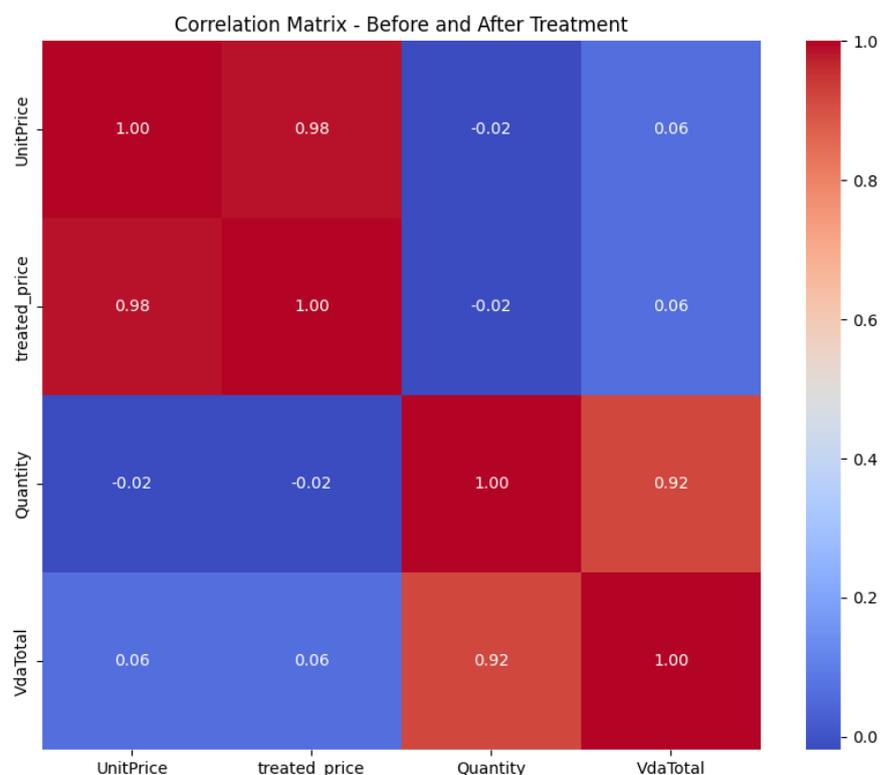


Figura 17 – Matriz de correlação entre preços originais, preços tratados, quantidade e valor total de vendas. Fonte: Elaborado pelo autor.

A análise da Figura 17 evidencia a baixa correlação entre as variáveis *treated_price* e *quantity*. Essa baixa correlação pode ser atribuída a fatores externos, como promoções sazonais ou flutuações de demanda por itens específicos, reforçando a importância de uma normalização eficaz dos preços para evitar distorções nos dados.

3.3.2 Identificação de *Outliers*

A identificação de *outliers* foi realizada utilizando duas técnicas estatísticas estabelecidas: o *z-score* (paramétrico) e o *Interquartile Range* (IQR - não paramétrico). O *z-score* identifica valores que se desviam significativamente da média, calculados como:

$$Z = \frac{X - \mu}{\sigma}, \quad (3.2)$$

onde μ é a média e σ o desvio padrão. Este método pressupõe normalidade dos dados e é sensível a distribuições assimétricas, pois valores extremos podem distorcer tanto μ quanto σ (WAN et al., 2014). Por outro lado, o método IQR, que utiliza os quartis $Q1$ e $Q3$, define *outliers* como valores fora do intervalo:

$$[Q1 - 1.5 \cdot IQR, Q3 + 1.5 \cdot IQR], \quad (3.3)$$

sendo $IQR = Q3 - Q1$. Por ser uma medida não paramétrica, o IQR é mais robusto frente a distribuições não normais e à presença de assimetrias (WAN et al., 2014).

Tabela 3 – Resumo comparativo dos métodos de identificação de *outliers*

Método	Outliers em <i>Quantity</i>	Outliers em <i>VdaTotal_normalized</i>
<i>Z-Score</i>	385	480
<i>IQR</i>	46.045	37.776

Fonte: Elaborado pelo autor.

A discrepância quantitativa observada na Tabela 3 reflete as diferenças metodológicas intrínsecas entre as técnicas. O IQR tende a detectar um número maior de *outliers* em variáveis com caudas pesadas, enquanto o *z-score* é mais restrito, uma vez que sua eficácia depende da suposição de normalidade. Por exemplo, na variável *Quantity*, a dispersão natural dos dados pode exceder os limites definidos pelo modelo paramétrico, reduzindo a eficácia do *z-score*.

Tabela 4 – Top 5 *outliers* mais extremos por *z-score*

Descrição	Quantidade	<i>VdaTotal_normalized</i> (£)
PAPER CRAFT, LITTLE BIRDIE	80,995	168.469,59
MEDIUM CERAMIC TOP STORAGE JAR	74,215	92.768,75
WORLD WAR 2 GLIDERS ASSTD DESIGNS	4,800	38.970,00
SMALL POPCORN HOLDER	4,300	15.445,44
EMPIRE DESIGN ROSETTE	3,906	7.940,70

Fonte: Elaborado pelo autor.

Como demonstrado na Tabela 4, *outliers* como “PAPER CRAFT, LITTLE BIRDIE” apresentam desvios acentuados da média, o que pode distorcer significativamente as métricas de análise. Em contrapartida, o IQR oferece maior estabilidade frente a valores isolados, preservando a integridade estatística, especialmente em distribuições assimétricas (WAN et al., 2014).

A Figura 18 ilustra a distribuição dos *outliers* detectados pelos dois métodos. Note-se que os intervalos interquartílicos, mais conservadores, refletem a robustez do IQR, enquanto os limites expandidos do *z-score* evidenciam a influência de uma variância inflada.



Figura 18 – Distribuição comparativa de *outliers* por método. Fonte: Elaborado pelo autor.

Os resultados indicam que a normalização dos preços reduziu efetivamente a variabilidade e preservou as relações fundamentais dos dados. A integração do tratamento

de preços com a análise de *outliers* permitiu uma compreensão detalhada dos padrões e anomalias presentes.

Ao comparar com os trabalhos de Bessa (2021) e Maramarque et al. (2024), observa-se que as abordagens anteriores geralmente se restringiram ao uso isolado de técnicas estatísticas para a detecção de *outliers* ou à seleção de variáveis, sem explorar uma normalização aprofundada dos dados. Em contraste, o presente trabalho foca em integrar uma estratégia híbrida que combina o *z-score* – com limiar fixado em $z = 3$, fundamentado na regra dos três sigma (CURTIS et al., 2016) – com o método do IQR. Essa abordagem não só garante a identificação criteriosa dos valores extremos, mas também preserva a estrutura correlacional dos dados, o que é essencial para o desempenho de algoritmos de *Machine Learning*, proporcionando modelos preditivos com maior acurácia e robustez.

3.3.3 Decomposição de Séries Temporais

A decomposição de séries temporais consiste em dividir uma série temporal em três componentes principais:

- **Tendência:** Refere-se ao comportamento de longo prazo da série, ou seja, a direção geral dos dados ao longo do tempo. Por exemplo, uma tendência crescente pode refletir o aumento das vendas de um produto ao longo dos anos.
- **Sazonalidade:** Representa padrões repetitivos que ocorrem em intervalos regulares de tempo, como flutuações nas vendas relacionadas a estações do ano ou feriados.
- **Resíduos:** São os elementos da série que não podem ser explicados pela tendência ou sazonalidade. Os resíduos representam as variações imprevisíveis ou ruído da série temporal.

A decomposição pode ser realizada de maneira aditiva ou multiplicativa. Neste trabalho, utiliza-se o modelo aditivo, representado por:

$$y_t = T_t + S_t + R_t, \quad (3.4)$$

onde y_t é o valor observado na série no tempo t , T_t é a componente de tendência, S_t é a componente sazonal, e R_t representa os resíduos.

3.3.4 Análise de Variância

A variância é uma medida estatística que quantifica a dispersão dos dados em relação à sua média. No contexto de séries temporais, calcular a variância de cada componente ajuda a entender como a variabilidade da série é distribuída entre tendência, sazonalidade

e resíduos. No presente trabalho, os resultados da decomposição estão detalhados nas Tabelas 5 e 6.

Tabela 5 – Análise das Variâncias dos Componentes de Séries Temporais

Variável	Original	Tendência	Sazonalidade	Resíduos
<i>Quantity</i>	16,7592	6,0304	0,0009	9,9739
<i>VdaTotal</i>	61,1391	21,1560	0,0036	36,9734

Tabela 6 – Proporções de Variabilidade Retirada e Restante

Variável	Retirada (%)	Restante (%)
<i>Quantity</i>	35,99	59,51
<i>VdaTotal</i>	34,61	60,47

Os resíduos, que concentram a maior parte da variabilidade restante, são de interesse particular, pois capturam as flutuações não explicadas pela tendência ou sazonalidade.

3.3.5 *Lags* e *Rolling Windows*

Lags (defasagens) representam o valor de uma variável em um período anterior. Por exemplo, o lag-1 de uma série y_t corresponde a y_{t-1} . Incorporar *lags* no modelo permite capturar dependências temporais nos dados, como o impacto do desempenho de um período no próximo. Neste trabalho, foram calculados os *lags* de 1, 3, 6 e 12 períodos para os resíduos das séries *Quantity* e *VdaTotal*.

Já as *Rolling Windows* (janelas móveis) calculam estatísticas, como a média ou desvio padrão, em intervalos temporais específicos. Por exemplo, a média móvel de 3 períodos para um dado tempo t é:

$$\text{Média Móvel}(t) = \frac{y_{t-2} + y_{t-1} + y_t}{3}. \quad (3.5)$$

As janelas móveis ajudam a suavizar flutuações de curto prazo, destacando padrões mais estáveis nos dados.

3.3.5.1 Importância das *Lags* e *Rolling Windows*

Tanto os *lags* quanto as *rolling windows* são fundamentais para capturar relações temporais nos dados. Os *lags* permitem modelar a autocorrelação entre períodos, enquanto as janelas móveis fornecem uma visão mais clara de tendências locais e ajudam a lidar com a volatilidade. Esses elementos são cruciais na construção de modelos preditivos que considerem a dependência temporal, especialmente ao trabalhar com resíduos, que representam as flutuações imprevisíveis de curto prazo.

3.3.6 Matriz de Correlação

Com base nas *features* geradas a partir dos resíduos, foi construída uma matriz de correlação, representada na Figura 19. Esta matriz ajuda a identificar relações lineares entre as variáveis derivadas.

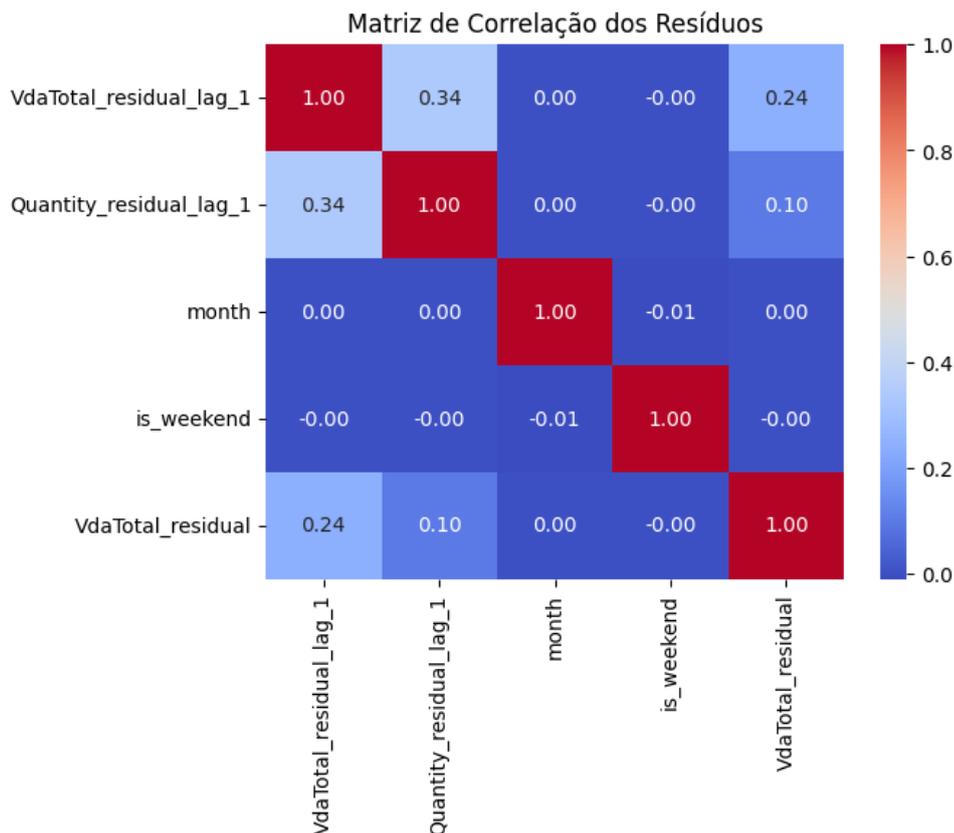


Figura 19 – Matriz de Correlação dos Resíduos. Fonte: Elaborado pelo autor.

A análise da matriz de correlação dos resíduos (Figura 19) revela padrões importantes nas relações temporais das variáveis estudadas. A correlação moderada positiva (0,34) entre *VdaTotal_residual_lag_1* e *Quantity_residual_lag_1* sugere uma dependência temporal significativa entre as vendas e quantidades defasadas em um período.

Observa-se também que *VdaTotal_residual* apresenta uma correlação fraca positiva (0,24) com sua própria defasagem (*VdaTotal_residual_lag_1*), indicando uma leve persistência temporal nas vendas. De maneira similar, *Quantity_residual_lag_1* mantém uma correlação fraca positiva (0,10) com *VdaTotal_residual*.

As variáveis temporais *month* e *is_weekend* demonstram correlações próximas a zero com as demais variáveis, sugerindo que os padrões mensais e de fim de semana têm pouca influência nos resíduos do modelo. Isso indica que a decomposição temporal foi eficaz em capturar e remover os efeitos sazonais relacionados a estes períodos.

Os resultados demonstram que a maior parte da variância das séries temporais

é explicada pela tendência e resíduos, enquanto a sazonalidade teve impacto mínimo. A análise de resíduos, complementada por *lags* e *rolling windows*, revelou padrões importantes para a modelagem preditiva.

Os valores defasados (*lags*) mostraram correlações moderadas entre si, como evidenciado pela correlação de 0,34 entre *VdaTotal_residual_lag_1* e *Quantity_residual_lag_1*. Essas informações são úteis no presente trabalho, que consideram dependências temporais.

Ao se visitar a literatura, trabalhos como [Maramarque et al. \(2024\)](#) e [Bessa \(2021\)](#), mostram que a adoção do modelo de decomposição do tipo aditiva foi coerente, visto que a amplitude da sazonalidade não variou de forma proporcional ao nível da série. Esse fato explica a baixa variância atribuída ao componente sazonal (inferior a 1% em ambos os casos). Em cenários onde a sazonalidade se intensifica ou se atenua à medida que os valores crescem, o modelo multiplicativo poderia ser mais apropriado.

No que diz respeito ao uso de *lags* e *rolling windows*, nossa abordagem segue as recomendações encontradas em estudos de [Fukai, Cavalieri e Castro \(2024\)](#) e [Mitra et al. \(2022\)](#), que enfatizam a importância de incorporar informações de períodos anteriores e médias móveis para capturar efeitos de memória de curto e médio prazo. A presença de correlações moderadas entre *lags* confirma a relevância de incluir tais variáveis nos modelos preditivos.

Por fim, a análise de variância e a consequente remoção de efeitos de tendência e sazonalidade permitem concentrar esforços na explicação das flutuações residuais, frequentemente associadas a promoções, mudanças de comportamento do consumidor ou fatores externos. Essa etapa de decomposição, portanto, enriquece a modelagem preditiva e facilita a identificação de padrões de autocorrelação nos resíduos, crucial para algoritmos de *machine learning* que serão aplicados nas etapas subsequentes.

3.4 Divisão do *Dataset*

Neste trabalho, foi conduzida uma análise de resíduos no Google Colaboratory utilizando bibliotecas do Python como *pandas*, *numpy*, *matplotlib*, *seaborn* e *sklearn*. O objetivo foi preparar o conjunto de dados para análise e modelagem, explorando os padrões nos resíduos e suas correlações. O *dataset* foi dividido em variáveis preditoras (X) e a variável-alvo (y). Neste caso, os resíduos foram utilizados como y (variável dependente) para análise mais precisa das dinâmicas internas. As variáveis preditoras foram:

- *VdaTotal_residual_lag_1* - Resíduo defasado de uma unidade temporal.
- *Quantity_residual_lag_1* - Quantidade residual defasada de uma unidade temporal.
- *month* - Mês do ano.

- *is_weekend* - Indicador de fim de semana.

A variável y foi definida como *VdaTotal_residual*, representando os resíduos calculados previamente.

3.4.1 Divisão Temporal dos Dados

Foi realizada uma divisão temporal utilizando a técnica *TimeSeriesSplit*, que garante a separação dos dados em blocos cronológicos, preservando a ordem temporal. Os tamanhos de treino e teste em cada divisão são apresentados na Tabela 7.

Tabela 7 – Tamanhos dos *Splits* na Divisão Temporal

Split	Tamanho Treino	Tamanho Teste
1	70162	70160
2	140322	70160
3	210482	70160
4	280642	70160
5	350802	70160

3.4.2 Distribuição de y_{train}

A Figura 20 apresenta a distribuição da variável y_{train} , que representa os resíduos utilizados para treinamento. Nota-se uma forma aproximadamente simétrica, indicando que os resíduos seguem um padrão próximo à distribuição normal, o que é desejável para análises estatísticas.

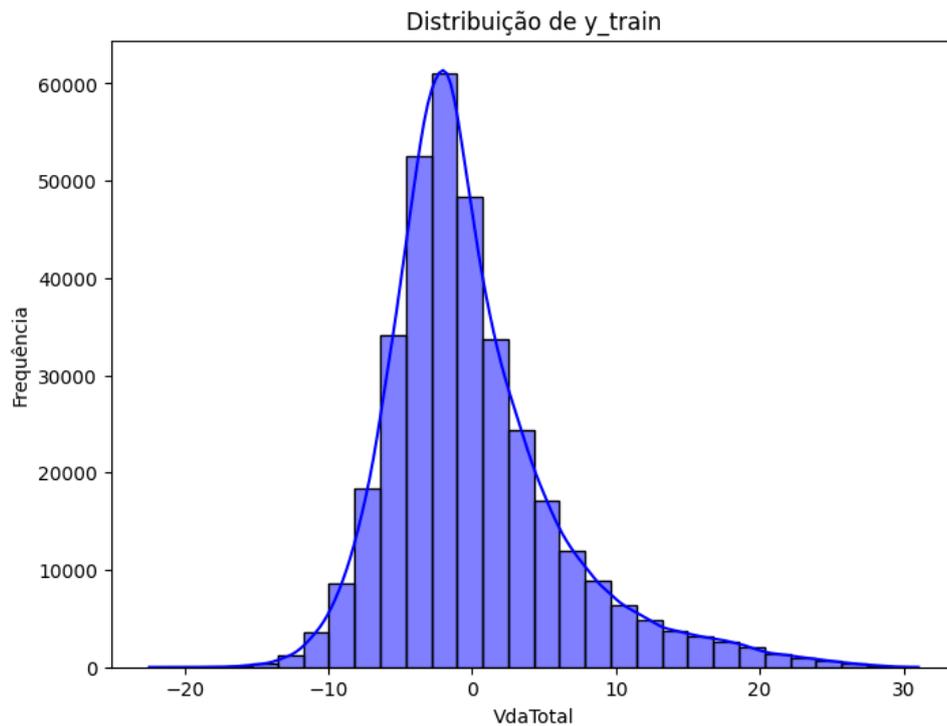


Figura 20 – Distribuição de y_{train} . Fonte: Elaborado pelo autor.

Esta distribuição reforça a adequação do uso de modelos estatísticos na análise, dado o comportamento dos resíduos. Além disso, ao se correlacionar esta figura com a matriz de correlação previamente apresentada na Figura 19, observa-se que a normalidade dos resíduos contribui para a confiabilidade das estimativas de parâmetros. Os passos realizados no *Google Colaboratory* incluíram a preparação do *dataset*, a divisão temporal e a análise de resíduos. A visualização da distribuição de y_{train} indica um comportamento adequado para aplicação de modelos preditivos, destacando-se a importância de trabalhar com resíduos normalizados.

3.5 Treinamento dos Modelos de *Machine Learning*

3.5.1 Descrição Geral da Implementação

A implementação consiste em comparar o desempenho de diferentes modelos de regressão aplicados a uma série temporal, com o objetivo de prever o atributo alvo. O atributo alvo (*target*) foi escolhido com base na relevância para o problema em questão, representando uma variável contínua cuja previsão impacta diretamente na tomada de decisão.

Foram utilizados três tipos principais de modelos:

- *Decision Tree*;

- *Random Forest*;
- XGBoost.

Além disso, foi implementado um modelo baseado em Redes Neurais Recorrentes utilizando LSTM (*Long Short-Term Memory*) para capturar padrões temporais complexos.

3.5.2 Divisão dos Dados e Estratégia de Validação

Foi utilizada a estratégia de validação cruzada específica para séries temporais, *TimeSeriesSplit*, com cinco *folds*. Esse método preserva a ordem temporal dos dados, dividindo-os em subconjuntos de treino e teste, garantindo que os dados de teste nunca estejam no passado dos dados de treino. Essa abordagem evita vazamento de informações temporais, crucial em problemas de séries temporais.

Cada *fold* foi avaliado utilizando as seguintes métricas:

- **Erro Médio Absoluto (MAE):** Mede a diferença absoluta média entre os valores previstos e os valores reais.
- **Raiz do Erro Quadrático Médio (RMSE):** Avalia a magnitude do erro com maior penalização para grandes discrepâncias.
- **Erro Percentual Absoluto Médio (MAPE):** Indica o erro percentual médio entre as previsões e os valores reais.

3.5.3 Pré-Processamento dos Dados

Antes do treinamento, os dados preditores foram normalizados utilizando a técnica de *StandardScaler*, que ajusta os valores para uma distribuição com média zero e desvio padrão igual a um. A normalização foi realizada separadamente para os conjuntos de treino e teste, garantindo que as informações do conjunto de teste não influenciassem o modelo durante o treino.

Para o modelo LSTM, os dados foram transformados em sequências temporais de comprimento fixo (*timesteps*), para permitir a modelagem da dependência temporal.

3.5.4 Modelos Base e Seus Hiperparâmetros

Inicialmente, os modelos base foram treinados sem ajustes personalizados nos hiperparâmetros:

- **Decision Tree:** Configuração padrão do *sklearn*.

- **Random Forest:** Configuração padrão do *sklearn*.
- **XGBoost:** Configuração padrão do *xgboost*.

Posteriormente, os hiperparâmetros dos modelos foram otimizados para melhorar o desempenho: Para o *Random Forest*, os hiperparâmetros foram otimizados com o intuito de reduzir a variância e evitar o sobreajuste, sem sacrificar a capacidade de captura de informações relevantes:

- ***n_estimators*:** Foi definido como 1000, aumentando o número de árvores do ensemble para reduzir a variância das previsões. Essa prática é consistente com a literatura, onde *ensembles* maiores tendem a fornecer estimativas mais robustas [Breiman \(2001\)](#).
- ***max_depth*:** Limitado a 8 para controlar a complexidade das árvores, evitando que elas se ajustem excessivamente aos ruídos dos dados.
- ***min_samples_split* e *min_samples_leaf*:** Definidos como 5 e 3, respectivamente, esses parâmetros garantem que os nós não sejam divididos com base em poucas amostras, o que contribui para reduzir o risco de sobreajuste.
- ***max_features*:** A escolha por *sqrt* (raiz quadrada do número de features) é uma prática comum que equilibra a redução de correlação entre as árvores e a diversidade do ensemble.
- ***bootstrap*:** Mantido em *True* para que cada árvore seja treinada com uma amostra com reposição, reforçando a variabilidade e robustez do modelo.
- ***max_samples*:** Estabelecido em 85%, esse parâmetro limita o tamanho da amostra utilizada em cada árvore, introduzindo mais aleatoriedade e contribuindo para a generalização.

No XGBoost, os parâmetros foram ajustados para mitigar o risco de sobreajuste e garantir um aprendizado gradual, conforme evidenciado em estudos recentes [Chen e Guestrin \(2016\)](#):

- ***n_estimators*:** Com 1000 estimadores, o modelo forma um *ensemble* robusto, aumentando a precisão da previsão.
- ***max_depth*:** Definido como 5, para evitar a complexidade excessiva das árvores e reduzir o risco de *overfitting*.
- ***learning_rate*:** Estabelecido em 0.01, promove atualizações mais graduais dos pesos, o que geralmente resulta em melhor generalização do modelo.

- ***subsample* e *colsample_bytree***: Ambos configurados em 85%, limitam, respectivamente, a proporção de dados e de *features* utilizadas em cada iteração, o que aumenta a diversidade entre as árvores e reduz o sobreajuste.
- ***gamma***: Com valor de 0.2, impõe uma regularização adicional, exigindo uma redução mínima na perda para que uma divisão seja realizada.
- ***min_child_weight***: Definido como 5, assegura que os nós só sejam divididos se a soma dos pesos das amostras exceder esse limiar, contribuindo para a estabilidade do modelo.
- ***reg_alpha* e *reg_lambda***: As regularizações L1 (0.5) e L2 (1.2) são incorporadas para penalizar pesos excessivamente grandes, promovendo modelos mais parsimoniosos e com melhor generalização.

3.5.5 Modelo LSTM

Para o modelo LSTM, a configuração da arquitetura e os hiperparâmetros foram escolhidos visando capturar as dependências temporais complexas presentes nas séries:

- **Camada LSTM com 64 unidades e ativação *tanh***: A escolha de 64 unidades representa um bom equilíbrio entre capacidade de modelagem e risco de sobreajuste, permitindo que o modelo capture relações temporais sem se tornar excessivamente complexo.
- **Camada *Dropout* com taxa de 30%**: Essa técnica é empregada para prevenir o sobreajuste, promovendo a robustez do modelo ao descartar aleatoriamente 30% dos neurônios durante o treinamento.
- **Camada densa com 32 neurônios e ativação *ReLU***: Introduce não linearidade após a camada LSTM e ajuda na transformação das representações extraídas para a tarefa de regressão.
- **Camada de saída com 1 neurônio**: Responsável por gerar a previsão final, adequada para problemas de regressão.
- **Otimizador Adam com *learning_rate* de 0.001 e função de perda *mse***: O otimizador Adam é amplamente utilizado por sua capacidade de adaptar a taxa de aprendizado para cada parâmetro individualmente, e a função MSE é a escolha padrão para tarefas de regressão.
- **Early Stopping**: Implementado para interromper o treinamento quando a *val_loss* não apresentar melhorias por 10 épocas consecutivas, prevenindo o sobreajuste e otimizando o tempo de treinamento.

O modelo foi compilado com o otimizador Adam ($learning_rate = 0.001$) e a função de perda mse . O treinamento foi realizado com *Early Stopping* para interromper o processo caso a val_loss não melhorasse após 10 épocas consecutivas.

Essas escolhas de hiperparâmetros foram embasadas em práticas consagradas na literatura e em experimentos preliminares, que indicaram melhorias significativas na desempenho dos modelos. A otimização sistemática desses parâmetros contribuiu para a robustez e acurácia dos modelos preditivos, permitindo uma comparação justa e a obtenção de resultados consistentes, conforme evidenciado nos experimentos.

4 Resultados

4.1 Análise dos Resultados Obtidos com *Decision Tree*, *Random Forest* e *XGBoost*

4.1.1 Parâmetros padrão do pacote *sklearn*

Nesta etapa, os modelos *Decision Tree*, *Random Forest* e *XGBoost* foram avaliados utilizando suas configurações padrão, sem ajustes nos hiperparâmetros. Os resultados obtidos, apresentados na Tabela 8, mostram que o *XGBoost* apresentou o melhor desempenho, destacando-se nas métricas *MAE*, *RMSE* e *MAPE*. Isso sugere que o *XGBoost* possui uma capacidade intrínseca de generalização superior às outras abordagens avaliadas, mesmo sem otimizações.

Tabela 8 – Resultados da avaliação dos modelos sem otimização de hiperparâmetros.

Modelo	MAE	RMSE	MAPE (%)
Decision Tree	6.2077	8.6028	7.1841
Random Forest	4.4941	6.1350	3.1081
XGBoost	4.2425	5.8782	2.2577

O *Decision Tree*, por outro lado, teve um desempenho inferior, o que pode ser atribuído à sua natureza mais simples e à tendência de *overfitting* em conjuntos de dados complexos. Já o *Random Forest* demonstrou resultados intermediários, beneficiando-se da agregação de várias árvores de decisão para reduzir o viés e a variância do modelo.

O *XGBoost* obteve os melhores resultados entre os modelos avaliados, apresentando os menores valores de *MAE*, *RMSE* e *MAPE*. Esse desempenho superior pode ser atribuído à sua capacidade de construir árvores de decisão de forma iterativa, corrigindo os erros das previsões anteriores, característica dos algoritmos de *boosting*. Além disso, o *XGBoost* incorpora técnicas avançadas de regularização, como penalizações L1 e L2, que ajudam a reduzir o *overfitting* e melhorar a generalização do modelo.

Diferentemente do *Random Forest*, que constrói árvores de decisão de forma independente e depois realiza uma média dos resultados, o *XGBoost* constrói suas árvores sequencialmente, otimizando a minimização da função de perda a cada iteração. Isso permite que o modelo aprenda padrões complexos e tenha um desempenho superior em problemas de previsão baseados em séries temporais.

Os resultados da Tabela 8 evidenciam essa vantagem, pois o *XGBoost* apresentou os menores erros em todas as métricas avaliadas, mesmo sem ajustes finos nos hiperparâmetros.

4.1.2 Resultados obtidos com otimização dos hiperparâmetros

A fim de explorar o potencial máximo dos dois modelos com melhor resultado na avaliação anterior, foram realizados ajustes sistemáticos nos hiperparâmetros. Para o *Random Forest*, o número de estimadores foi aumentado para 1000, e parâmetros como *min_samples_split*, *min_samples_leaf* e *max_samples* foram ajustados. Essa abordagem permitiu um controle mais refinado do *bootstrap*, enquanto a paralelização com *n_jobs=-1* acelerou o treinamento. De forma similar, para o *XGBoost*, o número de estimadores foi ajustado para 1000, a taxa de aprendizado foi reduzida para 0.01, e mecanismos de regularização L1 e L2 foram introduzidos para evitar *overfitting*. O tempo total de processamento foi de aproximadamente 17 minutos.

Os resultados dessa otimização são apresentados na Tabela 9, onde ambos os modelos apresentaram desempenhos aprimorados, com o *MAPE* reduzido significativamente. Curiosamente, os desempenhos do *Random Forest* e do *XGBoost* ficaram muito próximos após a otimização, o que pode ser explicado pela robustez dos dois algoritmos em cenários bem ajustados. Esse empate técnico sugere que a escolha entre os dois pode depender de fatores como o tempo de treinamento ou a facilidade de interpretação, dependendo do contexto do problema.

Tabela 9 – Resultados da avaliação dos modelos com otimização de hiperparâmetros.

Modelo	MAE	RMSE	MAPE (%)
Random Forest	4.2117	5.8266	2.1781
XGBoost	4.2130	5.8298	2.1790

4.2 Análise dos Resultados Obtidos com LSTM

O modelo *LSTM* foi avaliado utilizando os dados transformados em sequências temporais. O tempo total de processamento foi de aproximadamente 44 minutos. Os resultados dos cinco *folds* de validação cruzada são apresentados na Tabela 10. Observa-se que o *LSTM* apresentou desempenho consistente em todos os *folds*, com um *MAPE* médio de 1.5741%. Essa regularidade reforça sua capacidade de capturar padrões temporais complexos nos dados.

Tabela 10 – Resultados da avaliação do modelo LSTM.

Split	MAE	RMSE	MAPE (%)
Split 1	4.3768	5.9672	1.6321
Split 2	4.4065	5.9999	1.6440
Split 3	4.4278	6.0553	1.5663
Split 4	4.2499	5.8504	1.6371
Split 5	4.3736	6.0320	1.3909
Média	4.3669	5.9810	1.5741

Os resultados indicam que, apesar de o *LSTM* ter uma implementação mais complexa e um tempo de treinamento superior aos modelos baseados em árvores, ele oferece uma alternativa viável para tarefas que envolvem dependências temporais marcantes. A menor variabilidade nos resultados dos *folds* também sugere que o *LSTM* pode ser mais robusto em relação a diferentes divisões do conjunto de dados. Entretanto, uma análise mais aprofundada seria necessária para determinar a escalabilidade desse modelo em contextos de produção, considerando custos computacionais e a necessidade de ajustes finos nos hiperparâmetros.

Além disso, os resultados do *LSTM* podem ser interpretados à luz da natureza dos dados temporais. A hipótese é que as dependências de longo prazo capturadas pelas células de memória do *LSTM* sejam particularmente úteis em cenários onde os padrões temporais desempenham um papel crucial. No entanto, em dados com menor sazonalidade ou padrões menos marcados, a vantagem do *LSTM* pode ser reduzida.

4.3 Discussão Geral dos Resultados

Ao comparar os desempenhos dos modelos, alguns pontos se destacam. Primeiramente, a superioridade do *XGBoost* em sua configuração padrão indica que ele é uma escolha robusta para muitos problemas, especialmente quando o ajuste de hiperparâmetros não é viável. Os resultados mostram que o *XGBoost* apresentou os menores valores de erro entre os modelos testados inicialmente, destacando-se na capacidade de generalização. Esse desempenho superior pode ser atribuído à sua abordagem baseada em *boosting*, que corrige iterativamente os erros das previsões anteriores, além da incorporação de regularizações L1 e L2, que ajudam a reduzir o *overfitting* e melhorar a estabilidade do modelo.

Após a otimização dos hiperparâmetros, tanto o *Random Forest* quanto o *XGBoost* atingiram níveis de desempenho muito próximos, sugerindo que ambos os algoritmos possuem alto potencial quando bem ajustados. A redução dos erros após a otimização dos parâmetros evidencia a importância de um ajuste fino dos modelos para maximizar sua eficácia.

Já o *LSTM*, com seu *MAPE* médio de 1.5741%, demonstrou ser uma solução promissora para séries temporais, capturando padrões complexos que os modelos baseados em árvores podem não identificar com a mesma precisão. Apesar do tempo de processamento significativamente maior (44 minutos em comparação com os 14 a 17 minutos dos modelos baseados em árvores), o *LSTM* apresentou consistência nos resultados ao longo dos *folds* de validação cruzada, sugerindo sua capacidade de modelar relações temporais de longo prazo. No entanto, seu custo computacional elevado deve ser considerado, especialmente em contextos onde a escalabilidade e o tempo de resposta são fatores críticos.

Por fim, a análise dos resultados levanta hipóteses sobre a natureza dos dados

e os modelos utilizados. A presença de padrões temporais e interações não lineares nos dados favorece algoritmos como o *LSTM* e o *XGBoost*. A realização de mais experimentos, incluindo a introdução de variáveis explicativas adicionais e a avaliação de modelos híbridos, pode fornecer uma visão mais completa e robusta sobre a escolha do modelo ideal para diferentes cenários.

5 Conclusão

A proposta deste trabalho consistiu no estudo de modelos de aprendizado de máquina e redes neurais aplicados à previsão de demanda, com o objetivo de aprimorar a precisão das previsões em ambientes empresariais. Para isso, exploraram-se diferentes algoritmos, como *Decision Tree*, *Random Forest*, *XGBoost* e *Long Short-Term Memory (LSTM)*, avaliando seus desempenhos em cenários com características temporais e dependências não lineares.

Os resultados alcançados demonstraram a eficiência dos modelos *XGBoost* e *Random Forest* após otimizações nos hiperparâmetros, com desempenhos praticamente equivalentes. O *XGBoost* se destacou pelo menor erro percentual médio, enquanto o *Random Forest* apresentou um desempenho competitivo com menor complexidade de implementação. Já o *LSTM* mostrou-se robusto na captura de padrões temporais, sendo uma alternativa viável para cenários com forte dependência temporal, embora com um custo computacional significativamente maior.

Com base nas análises realizadas, conclui-se que os modelos desenvolvidos podem ser ferramentas valiosas para a gestão de recursos e a tomada de decisão em ambientes empresariais. A escolha do modelo mais apropriado deve considerar fatores como o tipo de dado, a complexidade da tarefa e os recursos computacionais disponíveis. Além disso, abordagens híbridas, que combinam as vantagens de diferentes algoritmos, surgem como alternativas promissoras para alcançar altos níveis de precisão em previsões de séries temporais.

5.1 Limitações

Este trabalho enfrentou diversas limitações que impactaram tanto a qualidade da modelagem quanto a precisão das previsões. Algumas dessas restrições estão relacionadas à qualidade dos dados disponíveis, enquanto outras dizem respeito a desafios metodológicos e computacionais.

Um dos principais desafios encontrados foi a qualidade e disponibilidade dos dados. A base utilizada continha ruídos, valores ausentes e *outliers*, especialmente nas variáveis relacionadas ao preço unitário e ao volume de vendas. A presença dessas inconsistências exigiu um pré-processamento rigoroso, incluindo técnicas de normalização, detecção e tratamento de *outliers*, bem como a criação de variáveis derivadas para compensar possíveis falhas nas informações originais.

A detecção de *outliers*, em particular, representou uma limitação significativa.

Embora a remoção de valores atípicos seja uma prática comum na modelagem de dados, a variação dos preços unitários ao longo do tempo para diferentes itens dificultou a identificação de padrões estáveis. Essa incerteza levanta questionamentos sobre a confiabilidade dos dados históricos e sua influência sobre a previsão de vendas.

Além disso, a modelagem de séries temporais apresentou desafios metodológicos e computacionais. A utilização de modelos baseados em redes neurais recorrentes, como *LSTM* (*Long Short-Term Memory*), mostrou-se promissora, mas exigiu um equilíbrio entre desempenho e eficiência computacional. O tempo de processamento foi uma limitação importante, visto que modelos mais complexos demandavam maior capacidade computacional, tornando inviável a execução de múltiplas simulações em um curto período.

Outro fator limitante foi a autocorrelação nos resíduos e a engenharia de *features*. Durante a análise exploratória, identificou-se que os resíduos das previsões apresentavam padrões recorrentes, sugerindo que variáveis adicionais poderiam ser incorporadas ao modelo. No entanto, a escolha das variáveis preditoras exigiu diversas iterações para evitar problemas como *overfitting* ou introdução de vieses nos resultados.

Por fim, limitações metodológicas também foram observadas. O estudo explorou diferentes abordagens de aprendizado de máquina, como *Decision Tree*, *Random Forest* e *XGBoost*, porém, cada uma dessas técnicas possui restrições específicas ao lidar com dados temporais. A necessidade de balancear interpretabilidade, acurácia e custo computacional foi um fator determinante na escolha dos modelos finais, o que pode ter impactado a precisão das previsões.

Apesar dessas limitações, as estratégias adotadas permitiram mitigar boa parte dos desafios encontrados, garantindo a obtenção de *insights* relevantes para a previsão de vendas. Trabalhos futuros podem explorar novas metodologias, aprimorar o tratamento de dados e utilizar infraestruturas computacionais mais robustas para superar as restrições identificadas neste estudo.

5.2 Sugestões para Trabalhos Futuros

Para futuras investigações, sugere-se a ampliação e o enriquecimento da base de dados utilizada, incorporando informações complementares – como fatores climáticos, dados promocionais e variáveis macroeconômicas – que podem influenciar significativamente a demanda. Além disso, a avaliação de modelos híbridos, combinando redes neurais e algoritmos baseados em árvores de decisão, pode proporcionar ganhos adicionais de desempenho, superando abordagens tradicionais.

Uma melhoria metodológica relevante diz respeito à divisão do *dataset*. Atualmente, utilizou-se a técnica *TimeSeriesSplit* para a separação cronológica dos dados; contudo,

estratégias mais sofisticadas podem ser exploradas, tais como a utilização de “sliding windows” (janelas móveis deslizantes) ou a validação com horizonte de previsão variável. Essas abordagens permitem uma avaliação mais robusta do desempenho do modelo ao longo do tempo, mitigando riscos de data *leakage* e possibilitando ajustes dinâmicos na divisão dos conjuntos de treino e teste conforme a evolução dos padrões temporais.

Outra vertente de aprimoramento refere-se à seleção de variáveis. Embora o presente trabalho tenha empregado a matriz de correlação para definir as *features*, métodos avançados – como o algoritmo *Boruta* (FUKAI; CAVALIERI; CASTRO, 2024) ou a avaliação de importância das variáveis via *Random Forest Feature Importance* (BREIMAN, 2001) – podem capturar correlações não lineares e interações complexas entre os preditores. Essas técnicas podem contribuir para a extração de variáveis mais relevantes, aprimorando a precisão do modelo.

Adicionalmente, recomenda-se a exploração de técnicas de *transfer learning* em redes neurais, permitindo que modelos pré-treinados em bases similares sejam adaptados a novos conjuntos de dados, reduzindo o tempo de treinamento e melhorando a capacidade de generalização. A aplicação dessas técnicas pode ser especialmente útil para domínios onde a coleta de dados é cara ou limitada.

Outro aspecto importante para futuras pesquisas é a avaliação da normalidade dos resíduos dos modelos preditivos. Testes estatísticos, como o Shapiro-Wilk ou o Kolmogorov-Smirnov, podem ser empregados para verificar se os resíduos seguem uma distribuição normal. Caso a normalidade não seja confirmada, a aplicação de transformações (por exemplo, Box-Cox ou Yeo-Johnson) para normalizá-los poderá aprimorar a robustez e a capacidade de generalização dos modelos, além de facilitar a aplicação de técnicas estatísticas mais avançadas.

Por fim, um aspecto essencial para trabalhos futuros é a explicabilidade dos modelos. Estratégias como *SHAP* e *LIME* podem ser aplicadas para tornar as previsões mais interpretáveis, facilitando sua aceitação e implementação prática em ambientes empresariais. Dessa forma, espera-se que este trabalho contribua tanto para o avanço da pesquisa acadêmica em previsão de demanda quanto para o desenvolvimento de soluções práticas e robustas para a indústria.

Referências

- ALI, Özden Gür et al. SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, Elsevier, v. 36, n. 10, p. 12340–12348, 2009. Citado 4 vezes nas páginas 11, 18.
- AYOTTE, B.; AL., et. Group Leakage Overestimates Performance: A Case Study in Keystroke Dynamics. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021. P. 1410–1417. Citado 1 vez na página 19.
- BERGSTRA, J.; AL., et. Algorithms for Hyper-Parameter Optimization. In: 24. ADVANCES in Neural Information Processing Systems. 2011. Citado 1 vez na página 27.
- BESSA, Adriana Bezerra. *Ensaaios sobre previsão de vendas no varejo de moda*. 2021. Tese (Doutorado). Citado 7 vezes nas páginas 11, 15, 16, 25, 28, 50, 53.
- BREIMAN, Leo et al. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984. P. 432. Citado 1 vez na página 20.
- BREIMAN, Leo. Random Forests. *Machine Learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado 3 vezes nas páginas 22, 57, 66.
- CHAI, T.; DRAXLER, R. R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? Arguments Against Avoiding RMSE in the Literature. *Geoscientific Model Development*, n. 7, p. 1247–1250, 2014. Citado 1 vez na página 26.
- CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. In: 22ND ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794. Citado 3 vezes nas páginas 24, 57.
- CHEN, Tianqi; HE, Tong et al. *Xgboost: Extreme Gradient Boosting*. 2015. P. 1–4. R package version 0.4-2. Citado 1 vez na página 24.
- CHOLLET, François. *Deep Learning with Python*. Simon & Schuster, 2021. Citado 1 vez na página 22.
- CURTIS, Alexander E et al. The mystery of the Z-score. *Aorta*, Thieme Medical Publishers, v. 4, n. 04, p. 124–130, 2016. Citado 2 vezes nas páginas 46, 50.
- DAWOOD, Edel Goreil. Geo-locating UEs Using Multi-output Decision Tree Regressor, 2019. Citado 2 vezes nas páginas 21, 24.
- DELGADO FILHO, Antonio Jorge Ferreira. *Análise de métodos de regressão para previsão de demanda de curto prazo*. 2020. Diss. (Mestrado) – Universidade Federal de Pernambuco. Citado 0 vez na página 21.

- DHALIWAL, Satvir; NAHID, Abdullah-Al; ABBAS, Rashid. Effective intrusion detection system using XGBoost. *Information*, Multidisciplinary Digital Publishing Institute, v. 9, n. 7, p. 149, 2018. Citado 1 vez na página 24.
- DIAS, George Paulus Pereira. Proposta de processo de previsão de vendas para bens de consumo. *Anais do XIX ENEGEP*, 1999. Citado 1 vez na página 11.
- DOBRA, A. *Classification and Regression Tree Construction*. 2002. Tese (Doutorado) – Cornell University. Citado 2 vezes nas páginas 20, 21.
- FLECK, L et al. Artificial neural networks: Basic principles. *Electronic Scientific Journal Innovation and Technology*, v. 1, n. 13, p. 47–57, 2016. Citado 1 vez na página 28.
- FRIEDMAN, Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, JSTOR, p. 1189–1232, 2001. Citado 1 vez na página 23.
- FUKAI, Fernanda Mayumi; CAVALIERI, Daniel Cruz; CASTRO, Fidelis Zanetti de. Long short-term memory neural networks applied in demand forecast in the retail market. In: IBERO-LATIN American Congress on Computational Methods in Engineering (CILAMCE). 2024. Citado 5 vezes nas páginas 16, 18, 53, 66.
- GEA, Cristiane et al. Análise de métodos de tratamento de outliers para predição dos retornos de índices de ações negociados em bolsa. In: SBC. ANAIS do XXXVI Simpósio Brasileiro de Bancos de Dados. 2021. P. 277–282. Citado 2 vezes na página 12.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 19, 28.
- HYNDMAN, Rob J; ATHANASOPOULOS, George. *Forecasting: Principles and Practice*. OTexts, 2018. Citado 3 vezes nas páginas 19, 25, 27.
- HYNDMAN, Rob J.; ATHANASOPOULOS, George. *Forecasting: Principles and Practice*. 3rd. Melbourne, Australia: OTexts, 2021. Accessed on: 06 fev. 2024. Disponível em: <http://otexts.com/fpp3/>. Citado 1 vez na página 14.
- JAMES, Gareth et al. *An Introduction to Statistical Learning*. New York: Springer, 2013. v. 112, p. 18. Citado 1 vez na página 15.
- MAKRIDAKIS, S. et al. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, México, 2018. Citado 2 vezes na página 18.
- MAKRIDAKIS, Spyros; WHEELWRIGHT, Steven C.; MCGEE, Victor E. *Forecasting: Methods and Applications*. 2nd. New York: John Wiley & Sons, 1983. Citado 1 vez na página 14.
- MARAMARQUE, Giulio Cesar Mirabella et al. Modelagem e previsão de preços de minério de ferro: um estudo utilizando ARIMA e Random Forest Regressor. Universidade Federal de Uberlândia, 2024. Citado 2 vezes nas páginas 50, 53.

- MITCHELL, T.M. *Machine Learning*. McGraw-Hill Education, 1997. (McGraw-Hill international editions - computer science series). ISBN 9780070428072. Disponível em: <https://books.google.com.br/books?id=x0GAngEACAAJ>. Citado 1 vez na página 17.
- MITRA, Arnab et al. A comparative study of demand forecasting models for a multi-channel retail company: a novel hybrid machine learning approach. In: SPRINGER, 4. OPERATIONS research forum. 2022. v. 3, p. 58. Citado 4 vezes nas páginas 16, 18, 53.
- MONARD, Maria Carolina et al. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, v. 1, n. 1, p. 32, 2003. Citado 1 vez na página 17.
- MORETTIN, Pedro A.; TOLOI, Clélia M. C. *Análise de séries temporais*. 2. ed. São Paulo: Edgard Blücher, 2006. Citado 2 vezes na página 17.
- PASSARI, Antonio Fabrizio Lima. *Exploração de dados atomizados para previsão de vendas no varejo utilizando redes neurais*. 2003. Tese (Doutorado) – Universidade de São Paulo. Citado 4 vezes nas páginas 11, 12, 14.
- PAVLYSHENKO, B. M. Machine-learning models for sales time series forecasting. *Data*, v. 4, n. 1, p. 15, 2019. Citado 1 vez na página 15.
- PEDREGOSA, F. et al. *Scikit-learn: Machine Learning in Python*. 2021. Disponível em: <https://scikit-learn.org/stable/>. Citado 1 vez na página 33.
- PHAN, Huy et al. Random Regression Forests for Acoustic Event Detection and Classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, IEEE, v. 23, n. 1, p. 20–31, 2014. Citado 1 vez na página 21.
- SANTANA, Adrielle De Carvalho. *Behavioral and Neurophysiological Representations of Speech Phonemic Units*. 2020. Tese de Doutorado – Université Grenoble Alpes [2020-....]; Universidade Federal de Minas Gerais. Citado 0 vez na página 20.
- SCIENCE, Towards Data. *Random Forest Regression - Along with its Implementation in Python*. 2019. Disponível em: <https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f>. Citado 0 vez na página 22.
- SHERSTINSKY, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, v. 404, p. 132306, 2020. Citado 1 vez na página 15.
- SILVEIRA, I. A. *Modelo de Previsão de Demanda com o Uso de Aprendizado Supervisionado de Máquina: Um Estudo de Caso em uma Empresa de Varejo*. 2019. Master's thesis – Florianópolis. Citado 1 vez na página 27.
- SIMIONATO, Daniel. *Estudo e comparação das técnicas de validação cruzada desenvolvidas para séries temporais*. 2022. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/16066>. Citado 1 vez na página 32.

- ULTRALYTICS. *Redes Neurais*. Disponível em: <https://www.ibm.com/br-pt/topics/neural-networks>. Acesso em: 18 mai. 2024. Citado 0 vez nas páginas 28, 31.
- VIANA, João José. *Administração de materiais: um enfoque prático*. Atlas, 2000. Citado 1 vez na página 41.
- WAN, Xiang et al. Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC medical research methodology*, Springer, v. 14, p. 1–13, 2014. Citado 3 vezes nas páginas 48, 49.
- YING, X. An Overview of Overfitting and its Solutions. *Physics: Conference Series*, v. 1168, p. 022022, 2018. Citado 1 vez na página 27.
- ZHANG, G. P. Avoiding Pitfalls in Neural Network Research. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, v. 37, n. 1, 2007. Citado 2 vezes na página 18.