

Previsão de demanda de cosméticos no varejo utilizando aprendizagem de máquina

Ana Clara Chaves Sousa



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

Ana Clara Chaves Sousa

Previsão de demanda de cosméticos no varejo

Dissertação apresentada ao curso Programa de Pós-Graduação em Informática
da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Mestre em Informática

Orientadora: Thaís Gaudencio do Rêgo

Abril de 2024



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Ana Clara Chaves Sousa, candidata ao título de Mestre em Informática na área de Sistemas de Computação, realizada em 28 de fevereiro de 2024.

Aos vinte e oito dias do mês de fevereiro do ano de dois mil e vinte e quatro, às 10h00min, no Centro de Informática da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final da discente Ana Clara Chaves Sousa, vinculada a esta Universidade sob a matrícula 20211000294, candidata ao grau de Mestre em Informática, na área de “*Sistemas de Computação*”, na linha de pesquisa “*Sinais, Sistemas Digitais e Gráficos*”, do Programa de Pós-Graduação em Informática. A comissão examinadora foi composta pelos professores: Thaís Gaudencio do Rêgo, Orientadora e Presidente; Tiago Maritan Ungulino de Araújo, Examinador Interno e Yuri de Almeida Malheiros Barbosa, Examinador Externo ao Programa. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou a finalidade da reunião e passou a palavra à candidata para que ela fizesse a exposição oral do trabalho de dissertação intitulado “**Previsão de demanda de cosméticos no varejo utilizando aprendizagem de máquina**”. Concluída a exposição, a candidata foi arguida pela Banca Examinadora que emitiu o seguinte parecer: “**aprovada**”. Do ocorrido, eu, Gilberto Farias de Sousa Filho, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da Banca Examinadora. João Pessoa, 28 de fevereiro de 2024.

Documento assinado digitalmente



GILBERTO FARIAS DE SOUSA FILHO
Data: 02/04/2024 07:24:24-0300
Verifique em <https://validar.iti.gov.br>

Pr

lho

Profa. Dra. Thaís Gaudencio do Rêgo
Orientador (PPGI-UFPB)

Documento assinado digitalmente



THAIS GAUDENCIO DO REGO
Data: 27/03/2024 14:54:51-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Tiago Maritan Ungulino de Araújo
Examinador Interno (PPGI-UFPB)

Documento assinado digitalmente



TIAGO MARITAN UGULINO DE ARAUJO
Data: 01/04/2024 14:16:00-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Examinador Externo (UFPB)

Documento assinado digitalmente



YURI DE ALMEIDA MALHEIROS BARBOSA
Data: 27/03/2024 17:45:26-0300
Verifique em <https://validar.iti.gov.br>

Catálogo na publicação
Seção de Catalogação e Classificação

S725p Sousa, Ana Clara Chaves.

Previsão de demanda de cosméticos no varejo
utilizando aprendizagem de máquina / Ana Clara Chaves
Sousa. - João Pessoa, 2024.

117 f. : il.

Orientação: Thaís Gaudencio do Rêgo.

Coorientação: Yuri de Almeida Malheiros Barbosa.

Dissertação (Mestrado) - UFPB/CI.

1. Aprendizado de máquina. 2. Previsão de séries
temporais. 3. Gradient boosting. 4. Prophet - Método.
5. Cosméticos no varejo. I. Rêgo, Thaís Gaudencio do.
II. Barbosa, Yuri de Almeida Malheiros. III. Título.

UFPB/BC

CDU 004(043)



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Dissertação do Programa de Pós-Graduação em Informática intitulada **Previsão de demanda de cosméticos no varejo** de autoria de Ana Clara Chaves Sousa, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Prof. Dr. Tiago Maritan Ugolino de Araújo
Universidade Federal da Paraíba

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba

João Pessoa, 26 de abril de 2024

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

“I must endure the presence of a few caterpillars if I wish to become acquainted with the butterflies.”

The Little Prince, Antoine de Saint-Exupéry.

Dedico esta dissertação à minha avó Maria, que, quando fui aprovada no mestrado, ficou mais feliz do que eu mesma. Hoje, ela não está mais ao meu lado fisicamente, mas tenho certeza que está celebrando essa conquista de onde estiver.

AGRADECIMENTOS

Agradeço à minha mãe, Cássia, por ter me mostrado desde a infância a importância de se ter dedicação para alcançar meus objetivos e por sempre cuidar de mim com todo o amor. Ao meu pai, Paulo Sérgio, por ter sempre me dado a liberdade de fazer minhas próprias escolhas. À minha irmã, Maura, por sempre acreditar na minha capacidade. Ao meu cunhado, Luciano, por sua constante prestatividade. À Liz, minha sobrinha e afilhada, por ser a alegria da casa. Tenho orgulho de chamar vocês de minha família.

Ao meu avô, Nhonhô, que não está mais nesse plano, mas que esteve presente na maior parte da minha vida. Aos meus avós paternos, Helena e João, pelo carinho. Às minhas madrinhas, Flávia, Inês Maria e Mariana e também aos familiares próximos, por todo o suporte.

Preciso agradecer também à parte da minha família que não é composta por humanos. À Mel, que salvou minha vida e me ensinou o maior amor do mundo. Ao Mamute, Nino, Chica, Pretinha, Belinha, Pandora, Taylor e Laila, por alegrarem todos os meus dias. Ao Tufão, Chacal, Pretinho, Nara e a todos os outros anjos de quatro patas que passaram pela minha vida e continuam dentro do meu coração.

Aos meus amigos por me apoiarem durante esse processo e, especialmente, à minha amiga Cecília. Se não fosse por ela, eu não estaria hoje na UFPB. À minha professora orientadora, Thaís, por ser a melhor orientadora que eu pude imaginar. Aos professores Yuri, Telmo, Léo e Tiago, por suas contribuições. Aos outros professores que tive em todas as fases da minha vida, do pré-escolar ao mestrado, por todos os ensinamentos. Aos líderes da empresa em que trabalho, pela compreensão, e aos colegas, pelo convívio leve do dia a dia.

E, por último, agradeço à Deus por ter me dado forças para lutar pelos meus sonhos.

RESUMO

A tomada de decisão baseada em dados foi facilitada devido à alta disponibilidade de dados e à maior capacidade de processamento dos computadores. Para auxiliar na tomada de decisão, é possível extrair informações dos dados através da Ciência de Dados. Um exemplo em que há grande aplicabilidade dessa ciência nas empresas é a previsão de demanda dentro da área de Gestão da Cadeia de Abastecimento. Fazer a previsão do volume de vendas não é uma tarefa trivial e, além disso, imprecisões nessa previsão podem causar ruptura de estoque ou afetar sua gestão. Neste estudo, será feita a previsão de vendas de dois canais de venda diferentes utilizando algoritmos de Aprendizado de Máquina para uma marca de uma grande empresa. Essa empresa está alocada no mercado de Higiene Pessoal, Perfumaria e Cosméticos, em que o Brasil é o quarto maior mercado consumidor do mundo. Foram utilizados dados dos anos de 2018 a 2023 de vendas que ocorreram em todos os estados brasileiros. As previsões foram feitas em três diferentes horizontes de tempo, sendo eles: curto prazo (próximo período), médio prazo (aproximadamente 3 meses à frente) e longo prazo (cerca de 7 meses à frente). O curto prazo é referente ao próximo ciclo para os métodos de regressão e próxima semana para o método de série temporal, o médio prazo é referente aos 5 ciclos à frente para os métodos de regressão e 15 semanas à frente para o método de série temporal e o longo prazo é referente à previsão de 10 ciclos à frente para os métodos de regressão e 30 semanas à frente para o método de série temporal. Sendo assim, a consistência dos modelos de Aprendizado de Máquina também foi avaliada. Os algoritmos analisados neste estudo foram: CatBoost, LightGBM, XGBoost e Prophet. Primeiramente, os métodos de *Gradient Boosting* mencionados foram comparados a fim de identificar qual dos três métodos indicou maior estabilidade ao prever múltiplos horizontes. O XGBoost apresentou os menores erros para o canal Loja na previsão em todos os três horizontes (10% para o curto prazo, 2,12% para o médio prazo e 6,4% para o longo prazo). Para o canal Venda Direta, o XGBoost não apresentou o menor WAPE em todos os horizontes, mas teve mais estabilidade em comparação ao CatBoost e ao LightGBM. Em sequência, o XGBoost foi comparado com um método de séries temporais, o Prophet. Comparando os dois modelos em cenários distintos, concluiu-se que o Prophet apresentou resultados mais satisfatórios e maior estabilidade na previsão de múltiplos horizontes temporais.

Palavras-chave: <Previsão de Séries Temporais>, <Aprendizado de Máquina>, <Gradient Boosting>, <Prophet>, <Varejo>.

ABSTRACT

Data-driven decision making was facilitated due to the high availability of data and the greater processing power of computers. To assist in decision making, it is possible to extract information from data through Data Science. An example in which there is great applicability of this science in companies is demand forecasting within the Supply Chain Management area. Forecasting sales volume is not a trivial task and inaccuracies in this forecast can cause stock-outs or affect its management. In this study, sales forecasts will be made for two different sales channels using Machine Learning algorithms for a brand owned by a large company. This company is in the Cosmetics, Fragrances and Toiletries market, where Brazil is the fourth largest consumer market in the world. Data was used from the years 2018 to 2023 on sales in all Brazilian states. Forecasts were made for three different time horizons: short term (next period), medium term (approximately 3 months ahead) and long term (approximately 7 months ahead). The short term refers to the next cycle for the regression methods and the next week for the time series method, the medium term refers to 5 cycles ahead for the regression methods and 15 weeks ahead for the time series method and the long term refers to the forecast of 10 cycles ahead for the regression methods and 30 weeks ahead for the time series method. Therefore, the consistency of the Machine Learning models was also evaluated. The algorithms analyzed in this study were CatBoost, LightGBM, XGBoost and Prophet. Firstly, the aforementioned Gradient Boosting methods were compared in order to identify which of the three methods showed the greatest stability when predicting multiple horizons. XGBoost had the lowest forecast errors for the Store channel in all three horizons (10% for the short term, 2.12% for the medium term and 6.4% for the long term). For the Direct Sales channel, XGBoost didn't have the lowest WAPE in all horizons, but it was more stable compared to CatBoost and LightGBM. Next, XGBoost was compared with a time series method, Prophet. Comparing the two models in different scenarios, it was concluded that Prophet showed more satisfactory results and more stability in forecasting multiple time horizons.

Key-words: <Time Series Forecasting>, <Machine Learning>, <Gradient Boosting>, <Prophet>, <Retail>.

LISTA DE FIGURAS

1	Exemplo de Árvore de Regressão.	27
2	Metodologia geral utilizada.	41
3	Detalhamento da metodologia utilizada para os algoritmos de GB.	42
4	Detalhamento da metodologia utilizada para algoritmo de séries temporais: Prophet.	43
5	Conjuntos de treino, validação e teste para otimização de hiperparâmetros.	56
6	WAPE por ciclo predito para previsão do próximo ciclo.	59
7	WAPE por ciclo predito para previsão de 5 ciclos à frente.	60
8	WAPE por ciclo predito para previsão de 10 ciclos à frente.	60
9	WAPE do CatBoost por ciclo predito e horizonte.	61
10	WAPE do LightGBM por ciclo predito e horizonte.	62
11	WAPE do XGBoost por ciclo predito e horizonte.	62
12	Decomposição aditiva da série temporal em relação à tendência, sazonalidade e resíduos entre o segundo semestre de 2019 e o primeiro semestre de 2023.	77

LISTA DE TABELAS

1	Principais hiperparâmetros de métodos GB.	30
2	Resumo dos trabalhos relacionados.	40
3	Variáveis disponíveis na base de dados.	44
4	Dados nulos e tratativas.	45
5	Variáveis criadas considerando produtos, campanhas e promoções.	47
6	Variáveis construídas calculando o <i>market share</i> do último ano.	48
7	Variáveis de <i>lag</i> criadas para os modelos GB considerando o ciclo utilizado na predição.	50
8	Variáveis de <i>lag</i> criadas para o método de séries temporais.	50
9	Variáveis construídas com informações dos ciclos para os métodos GB.	51
10	Variáveis construídas com informações dos ciclos para o método ST.	51
11	Variáveis automáticas construídas utilizando a biblioteca <i>scikit-learn</i>	52
12	Faixa de valores para otimização dos hiperparâmetros do XGBoost.	55
13	Faixa de valores para otimização dos hiperparâmetros do Prophet.	55
14	WAPE por canal.	58
15	WAPE por regional do canal Loja.	63
16	WAPE por regional do canal Venda Direta.	64
17	WAPE por categoria do canal Loja.	65
18	WAPE por categoria do canal Venda Direta.	66
19	WAPE por subcategoria do canal Loja.	66
20	WAPE por subcategoria do canal Venda Direta.	67
21	WAPE por marca do canal Loja.	68
22	WAPE por marca do canal Venda Direta.	69
23	WAPE por canal para previsão do próximo ciclo: Com FE vs Sem FE.	70
24	WAPE por canal para previsão de 5 ciclos à frente: Com FE vs Sem FE.	70
25	WAPE por canal para previsão de 10 ciclos à frente: Com FE vs Sem FE.	71
26	Atributos mais importantes para o modelo CatBoost.	72
27	Atributos mais importantes para o modelo LightGBM.	73

28	Atributos mais importantes para o modelo XGBoost.	74
29	Quantidade de atributos com interseção entre os três métodos para cada canal e horizonte.	75
30	WAPE do XGBoost por horizonte e canal: Com e sem otimização de hiperparâmetros.	76
31	WAPE por canal para previsão em múltiplos horizontes: Com FE vs Sem FE.	78
32	WAPE por canal para previsão em múltiplos horizontes: Com FE vs Sem FE.	79
33	Comparativo do WAPE por canal e horizonte.	80
34	Comparativo do WAPE por horizonte e regional no canal Loja.	80
35	Comparativo do WAPE por horizonte e regional no canal Venda Direta. . .	81
36	Comparativo do WAPE por horizonte e categoria no canal Loja.	81
37	Comparativo do WAPE por horizonte e categoria no canal Venda Direta. .	82
38	Comparativo do WAPE por horizonte e subcategoria no canal Loja.	82
39	Comparativo do WAPE por horizonte e subcategoria no canal Venda Direta.	83
40	Comparativo do WAPE por horizonte e marca no canal Loja.	83
41	Comparativo do WAPE por horizonte e marca no canal Venda Direta. . . .	83

LISTA DE ABREVIATURAS

AM - Aprendizado de Máquina

CTF - *Cosmetic, Fragrante and Toiletry* (Higiene Pessoal, Perfumaria e Cosméticos)

EFB - *Exclusive Feature Bundling* (pacote de atributos exclusivos)

GB - *Gradient Boosting* (Gradiente Descendente)

GOSS - *Gradient-Based One-Side Sampling* (amostragem unilateral baseada em gradiente)

IA – Inteligência Artificial

MAE - *Mean Absolute Error* (erro médio absoluto)

MAPE - *Mean Absolute Percentage Error* (erro percentual médio absoluto)

ML – *Machine Learning* (Aprendizagem de Máquina)

RMSE - *Root Mean Squared Error* (raiz do erro médio quadrado)

RMSPE - *Root Mean Squared Percentage Error* (raiz do erro percentual médio quadrado)

SCM – *Supply Chain Management* (Gestão da Cadeia de Abastecimento)

WAPE - *Weighted Average Percentage Error* (erro percentual médio ponderado)

WMAE - *Weighted Mean Absolute Error* (erro absoluto médio ponderado)

Sumário

1	INTRODUÇÃO	19
1.1	Contexto	21
1.2	Justificativa	21
1.3	Objetivo Geral	22
1.4	Objetivos Específicos	22
1.5	Contribuições	22
1.6	Estrutura da Dissertação	23
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	Séries Temporais	24
2.2	Árvores de Decisão	26
2.3	Gradient Boosting	28
2.4	Algoritmos Utilizados	30
2.4.1	CatBoost	31
2.4.2	LightGBM	32
2.4.3	XGBoost	33
2.4.4	Prophet	34
2.5	Avaliação dos Modelos	35
2.6	Testes de Hipótese	36
2.6.1	Teste de Shapiro-Wilk	36
2.6.2	Teste de Kruskal-Wallis	37
3	TRABALHOS RELACIONADOS	38
4	METODOLOGIA	41
4.1	Base de Dados	43
4.2	Hardware	44
4.3	Tecnologias Utilizadas	44
4.4	Pré-Processamento dos Dados	45

4.4.1	Preparação dos Dados para Algoritmos de Gradient Boosting	46
4.4.2	Preparação dos Dados para Algoritmo de Séries Temporais	46
4.5	Engenharia de Atributos	47
4.6	Modelagem	52
5	RESULTADOS	57
5.1	Comparativo entre modelos de Gradient Boosting	57
5.1.1	Análise dos Erros por Canal de Vendas	58
5.1.2	Análise dos Erros por Ciclo e por Canal	59
5.1.3	Análise dos Erros por Regional	63
5.1.4	Análise dos Erros dos Produtos	64
5.1.5	Testes de Hipótese	69
5.1.6	Impacto da Engenharia de Atributos	69
5.1.7	Impacto da Otimização de Hiperparâmetros	76
5.2	Análise dos Componentes de Séries Temporais	76
5.3	Análise do método de Séries Temporais: Prophet	78
5.3.1	Impacto da Engenharia de Atributos	78
5.3.2	Impacto da Otimização de Hiperparâmetros	78
5.4	Comparativo entre XGBoost e Prophet	79
6	CONSIDERAÇÕES FINAIS	85
	REFERÊNCIAS	87
	APÊNDICE A	90
	APÊNDICE B	105

1 INTRODUÇÃO

O desempenho e a posição das organizações perante o mercado estão sempre atrelados às decisões que são tomadas. Tanto o sucesso quanto o fracasso são atribuídos a essas decisões. Sendo assim, existe uma pressão para que os responsáveis por tomar as decisões sejam capazes de otimizá-las em tempo hábil (BARTKUS; MANNOR; CAMPBELL, 2022).

A disponibilidade de dados em grande volume e variedade, além da maior capacidade de processamento dos computadores, facilitaram o que é chamado de tomada de decisão baseada em dados (do inglês, *data-driven decision making*) (PROVOST; FAWCETT, 2013a). *Data-driven decision making* é a prática de tomar decisões com base em análise de dados em vez de intuição. Uma empresa orientada a dados tende a ser considerada mais produtiva (PROVOST; FAWCETT, 2013b).

Para auxiliar nessa tomada de decisão, a Ciência de Dados (do inglês, *Data Science* - DS) é muito utilizada. Refere-se a uma área preocupada com a coleta, preparação, análise, visualização, gestão e preservação de grandes conjuntos de dados (SALTZ; STANTON, 2017). DS foi definida resumidamente como a ciência de adquirir conhecimento por meio dos dados. Essa ciência trata de como os dados são utilizados para extrair conhecimento (OZDEMIR, 2016).

Pode-se dizer que DS é formada por três pilares (OZDEMIR, 2016):

- **Matemática e Estatística:** Refere-se a entender equações, funções e fórmulas para realizar análises e fazer modelagens matemáticas.
- **Programação de computadores:** Refere-se à capacidade de escrever códigos em alguma linguagem de programação para criar a solução.
- **Domínio do negócio:** Refere-se ao entendimento do negócio e problema a ser resolvido, podendo ser de medicina, finanças, indústria, dentre outros.

DS engloba um conjunto de princípios, definições de problemas e algoritmos. Além disso, está relacionada aos campos de Mineração de Dados e Aprendizado de Máquina (do inglês, *Machine Learning*) (KELLEHER; TIERNEY, 2018).

Aprendizado de Máquina (AM) é um campo de pesquisa que está na interseção de Inteligência Artificial, Ciência da Computação e Estatística e é também conhecido como análise preditiva ou aprendizado estatístico. Está presente na vida cotidiana das mais diversas formas, como em recomendações em *streaming* sobre quais filmes ou séries assistir e reconhecimento facial. Os algoritmos de AM mais conhecidos são aqueles que

automatizam os processos de tomada de decisão encontrando padrões, a partir de dados históricos conhecidos (MULLER; GUIDO, 2018).

Definições diferentes para Aprendizado de Máquina estão presentes na literatura, uma delas é “a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência” (MITCHELL, 1997). Sendo assim, os computadores são programados para aprender com a experiência passada (FACELI, 2011).

Com informações dos dados adquiridas através de técnicas de AM, é possível tomar decisões, fazer previsões, compreender tanto o presente quanto o passado, além de criar novos produtos e indústrias (OZDEMIR, 2016). Dentre as diversas áreas em que decisões baseadas em dados podem ter aplicabilidade, a de Gestão da Cadeia de Abastecimento (do inglês, *Supply Chain Management* - SCM) é um grande exemplo. Toda a área de SCM engloba logística, transporte, gestão de operações, compras, engenharia, pesquisa e desenvolvimento (SCHOENHERR; SPEIER-PERO, 2015).

Existem diversas possibilidades dentro dessa área, tais como análise de comportamento do cliente, análise de tendências e previsão de demanda (SEYEDAN; MAFAKHERI, 2020). Schoenherr e Speier-Pero (2015) fizeram um estudo para salientar os benefícios de utilizar análise preditiva na área, através de dados coletados de profissionais de SCM, trazendo como exemplos desses benefícios:

- Melhor tomada de decisão;
- Maior visibilidade;
- Melhor gerenciamento do risco da cadeia de suprimentos;
- Melhoria na eficiência da cadeia de suprimentos;
- Melhoria nos custos da cadeia de suprimentos;
- Recursos aprimorados de planejamento de demanda;
- Recursos aprimorados de planejamento de vendas e operações.

Tendo tudo isso em vista, o tema deste trabalho será previsão de demanda no varejo, em múltiplos horizontes temporais, utilizando técnicas de Aprendizado de Máquina em uma grande empresa de varejo e indústria do mercado de Higiene Pessoal, Perfumaria e Cosméticos (do inglês, *Cosmetic, Fragrance and Toiletry* - CFT). O nome da empresa será mantido em sigilo para preservar sua privacidade e integridade, visto que é uma condição necessária para que a colaboração na pesquisa continue e os dados detalhados sejam disponibilizados. Sendo assim, a empresa de estudo será referida como Empresa X.

Neste estudo, algoritmos de aumento de gradiente (do inglês, *Gradient Boosting* - GB) serão analisados, sendo eles o CatBoost (DOROGUSH; ERSHOV; GULIN, 2018), LightGBM (KE et al., 2017) e XGBoost (CHEN; GUESTRIN, 2016). Os métodos GB foram utilizados devido à complexidade dos dados disponíveis e à sua flexibilidade para lidar com atributos categóricos e *outliers*. Além dos métodos de regressão, um algoritmo de séries temporais (ST) também será estudado, o Prophet (TAYLOR; LETHAM, 2018). O Prophet foi o método de série temporal escolhido para este estudo em razão da detecção de sazonalidade ser parametrizável, além de sua versatilidade para fazer previsões em janelas de horizonte temporal.

1.1 Contexto

A Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos mostrou que o Brasil é o quarto maior mercado consumidor do mundo, o segundo maior mercado no ranking global de países que mais lançam produtos anualmente e também o segundo maior mercado em fragrâncias, produtos masculinos e desodorantes (ABIHPEC, 2022). É importante destacar que CFT é um mercado que apresenta bastante complexidade. Além disso, há grande interferência de indicadores macroeconômicos que não são possíveis de prever.

A Empresa X está presente não só em todos os estados do Brasil, mas também em outros países, tanto da América, quanto de outros continentes. É um negócio complexo, com múltiplos canais de venda (lojas, venda direta e *e-commerce*), diversas marcas próprias e um portfólio composto por milhares de produtos. Para delimitar o estudo, foi escolhida uma marca e dois dos três canais: loja e venda direta.

1.2 Justificativa

Prever o volume de vendas de uma empresa não é uma tarefa trivial. Quando uma previsão superestima a demanda, isso pode afetar a gestão de estoque, além de causar desperdícios. Por outro lado, quando a demanda é subestimada, isso causa ruptura de estoque (ANDRADE; CUNHA, 2022). Em empresas que estão no segmento do varejo e da indústria ao mesmo tempo, como é o caso da Empresa X, o problema é especialmente mais complexo. Para o produto estar disponível para compra nas prateleiras das lojas, nas mãos de representantes de venda ou no *e-commerce*, é preciso um planejamento com meses de antecedência, a depender do tempo de produção. É necessário saber quanto e quando adquirir a matéria-prima para a produção de um produto, quanto tempo leva para ser fabricado e acabado, qual o volume que será armazenado e transportado. A previsão de demanda afeta a cadeia como um todo. Em casos como esse, prever apenas as vendas do próximo mês, por exemplo, não é o suficiente.

1.3 Objetivo Geral

Propor uma solução para previsão de demanda para diferentes categorias, subcategorias, marcas e canais da Empresa X em múltiplos horizontes temporais, identificando padrões de consumo e entendendo qual dos modelos comparados possui uma performance mais estável nesses horizontes.

1.4 Objetivos Específicos

1. Construir variáveis considerando características dos produtos, promoções, preços, campanhas e informações temporais;
2. Explorar os componentes de séries temporais (tendência, sazonalidade, ciclo e ruído);
3. Comparar métodos de regressão para a previsão de demanda, focando os estudos na metodologia de *Gradient Boosting* através dos algoritmos CatBoost, LightGBM e XGBoost;
4. Comparar a performance dos modelos para previsão de curto prazo (próximo ciclo para GB e próxima semana para ST), médio prazo (5 ciclos a frente para GB e 15 semanas a frente para ST) e de longo prazo (10 ciclos a frente para GB e 30 semanas a frente para ST) e, assim, verificar a consistência dos modelos em horizontes diferentes;
5. Analisar o impacto da etapa de Engenharia de Atributos para todos os algoritmos estudados;
6. Analisar o impacto da etapa de otimização de hiperparâmetros para o algoritmo de melhor performance do GB (XGBoost) e o Prophet;
7. Comparar o método GB de melhor performance com o método ST utilizado no estudo (Prophet).

1.5 Contribuições

Este trabalho possui contribuições para a empresa de estudo, contribuições científicas e contribuições generalizadas para a previsão de demanda em empresas do varejo (especialmente para o mercado de CFT).

Para a Empresa X, pode-se enumerar as seguintes contribuições:

- Novas percepções sobre métodos diferentes para a previsão de demanda, de forma que os algoritmos utilizados atualmente pela empresa, com esse intuito, possam ser reavaliados;

- Exploração das *features* criadas neste estudo, podendo estas serem acopladas a soluções já existentes;
- Entendimento sobre a estabilidade dos métodos, que serve como auxílio para os casos em que as janelas de predição são maiores.

Além disso, até o momento, este trabalho resultou em duas produções científicas, sendo elas:

- Um artigo completo publicado e apresentado em conferência no XX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2023) intitulado *Comparing Gradient Boosting Algorithms to Forecast Sales in Retail*. O evento aconteceu na Universidade Federal de Minas Gerais (UFMG), em Belo Horizonte, entre os dias 25 de setembro e 29 de setembro de 2023. Este artigo teve como objetivo prever as vendas de uma empresa no mercado de CFT em múltiplos horizontes temporais, comparando a performance de três algoritmos de *Gradient Boosting* (CatBoost, LightGBM e XGBoost) (SOUSA et al., 2023). Este artigo pode ser encontrado no Apêndice A.
- Um artigo completo submetido para o periódico *Brazilian Journal of Production Engineering (BJPE)* intitulado *Previsão de Demanda de Longo Prazo Aplicada à uma Empresa de Varejo utilizando o Prophet*. Este artigo é complementar ao anterior e tem como objetivo fazer a previsão de demanda de longo prazo (30 semanas) utilizando o Prophet. Este artigo pode ser encontrado no Apêndice B.

Em relação à contribuição para outras empresas do varejo, pode-se ressaltar a possibilidade de aplicar a metodologia desenvolvida em bases de dados similares.

1.6 Estrutura da Dissertação

Este trabalho está organizado da seguinte forma:

- **Capítulo 1:** Introdução ao tema abordado e à pesquisa.
- **Capítulo 2:** Fundamentação Teórica.
- **Capítulo 3:** Exposição de trabalhos relacionados focados em previsão de demanda.
- **Capítulo 4:** Metodologia utilizada no desenvolvimento da pesquisa.
- **Capítulo 5:** Discussões e Resultados.
- **Capítulo 6:** Considerações Finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados conceitos gerais importantes para a temática do problema. A Seção 2.1 aborda séries temporais. Os conceitos de Árvore de Decisão (do inglês, *Decision Tree* - DT) aplicada em Regressão (Seção 2.2) e Árvores Impulsionadas por Gradiente (do inglês, *Gradient Boosting Trees* - GBT) (Seção 2.3) são necessários para o entendimento dos algoritmos de GB utilizados nos experimentos.

A Seção 2.4 aborda os algoritmos utilizados para fazer a comparação de suas performances para as previsões de demanda em horizontes distintos. Os métodos utilizados foram:

- CatBoost (DOROGUSH; ERSHOV; GULIN, 2018) (Seção 2.4.1)
- LightGBM (KE et al., 2017) (Seção 2.4.2)
- XGBoost (CHEN; GUESTRIN, 2016) (Seção 2.4.3)
- Prophet (TAYLOR; LETHAM, 2018) (Seção 2.4.4)

Em sequência, a Seção 2.5 fundamenta o WAPE, que foi a métrica de erro utilizada para avaliar a performance dos modelos. E, por último, a Seção 2.6 apresenta os testes de hipótese conduzidos tomando como base o WAPE calculado: o teste de normalidade de Shapiro-Wilk e o teste de Kruskal-Wallis.

2.1 Séries Temporais

A análise de séries temporais pode ser utilizada em diversas áreas. Na economia, por exemplo, é possível fazer acompanhamento de Produto Interno Bruto (PIB) e oferta e demanda. Além disso, é muito utilizada para acompanhamento de fenômenos da natureza, em análise demográfica, em marketing, na qualidade e na saúde. Tendo isso em vista, ela pode ser muito útil para a ciência de dados (SOUSA; SILVA; AL., 2021).

Devido à produção volumosa de dados, a análise de séries temporais têm ganhado grande importância. Essa geração de dados acontece por diversos meios, como, por exemplo, a Internet das Coisas (do inglês, *Internet of Things* - IoT), a digitalização dos sistemas e o surgimento de cidades inteligentes. Como consequência, a necessidade de análises de séries temporais eficientes com técnicas estatísticas e de AM também aumenta. O objetivo da análise de séries temporais é extrair informações significativas de pontos de dados organizados em ordem cronológica, diagnosticando comportamentos passados e predizendo comportamentos futuros (NIELSEN, 2019). Os computadores mais robustos, conjuntos de dados maiores e melhores ferramentas de programação permitiram mais experimentação

e capacidade de responder perguntas difíceis de análise e previsão de séries temporais (NIELSEN, 2019).

Após a Segunda Guerra Mundial, foram expostos pelo setor empresarial os problemas iniciais na análise prática de séries temporais. Dentre esses problemas, pode-se mencionar a previsão de demanda, estimativa de preços futuros de matéria-prima e alocação de custos de fabricação. As análises de séries temporais e AM passaram a ter uso prático no início da década de 1980 (NIELSEN, 2019).

Por definição, uma Série Temporal T é uma sequência ordenada de n variáveis de valor real (ESLING; AGON, 2012):

$$T = (t_1, \dots, t_n), \quad t_i \in R, \quad (1)$$

onde t representa a variável no instante de tempo e n é a quantidade de variáveis.

Uma Série Temporal é muitas vezes o resultado da observação de um processo subjacente do qual os valores são coletados a partir de medições feitas em instantes de tempo. A série pode ser univariada, como na Equação 1, ou multivariada, que ocorre quando várias séries abrangem simultaneamente diversas dimensões dentro de um mesmo intervalo de tempo (ESLING; AGON, 2012).

Alguns exemplos de séries temporais são (MORETTIN; TOLOI, 2018):

1. Índices diários de poluição na cidade de São Paulo;
2. Valores mensais de temperatura em uma cidade;
3. Precipitação atmosférica anual na cidade de Fortaleza;
4. Registros de marés no porto de Santos.

Nos exemplos de 1 a 3, temos séries temporais discretas, enquanto o último exemplo (4) mostra uma série temporal contínua. Em geral, a série temporal discreta é obtida através da amostragem de uma série temporal contínua nos mesmos intervalos de tempo Δt (MORETTIN; TOLOI, 2018).

Uma análise de séries temporais pode ter objetivos diferentes (MORETTIN; TOLOI, 2018), como, por exemplo:

- Investigar o mecanismo gerador da série temporal;
- Fazer previsões de valores futuros da série, podendo estes serem a curto prazo ou a longo prazo, sendo aplicados a previsão de vendas, produção, estoque, etc;

- Descrever o comportamento da série, como a presença de tendências, ciclos e sazonalidade;
- Encontrar periodicidades relevantes nos dados.

Em geral, uma variável que está sendo estudada pode ser classificada por quatro componentes, sendo que estes podem aparecer separadamente ou de maneira combinada (SOUSA; SILVA; AL., 2021):

- **Tendência:** É o que ocorre quando é observada uma tendência de crescimento, ou de queda, na evolução da variável ao longo do tempo.
- **Ciclo:** São movimentos ascendentes ou descendentes em uma linha temporal, que ocorrem de forma cíclica dentro de uma tendência.
- **Sazonalidade:** É explicado com um movimento na linha temporal, que ocorre repetidamente em períodos específicos.
- **Irregularidade:** É considerada um ruído presente no modelo de série temporal. Tudo que não representar uma tendência, um ciclo ou uma sazonalidade é um erro aleatório, ou seja, uma irregularidade.

A análise de séries temporais é comumente realizada através de métodos estatísticos tradicionais e por algumas técnicas de AM, como modelos de árvores e ajustes lineares (NIELSEN, 2019). Neste estudo, serão utilizados 3 métodos de AM que não foram desenvolvidos originalmente para dados de séries temporais e um método baseado em séries temporais. Estes serão apresentados na Seção 2.4.

A Seção 2.2 trata do conceito de Árvore de Decisão, que é fundamento para os algoritmos que serão utilizados neste estudo. É importante ressaltar que, no caso de metodologias baseadas em árvore, criar as características da série temporal é uma etapa necessária para utilizar esse tipo de método. Isso porque as árvores não são métodos próprios para considerar a variável tempo, ao contrário de um modelo ARIMA (do inglês, *Autoregressive Integrated Moving Average* - Modelo Autorregressivo Integrado de Médias Móveis), por exemplo (NIELSEN, 2019).

2.2 Árvores de Decisão

Os métodos de árvore de decisão são amplamente utilizados tanto para problemas de classificação, quanto de regressão. Em resumo e de forma análoga, o aprendizado é através de uma hierarquia de perguntas se/senão, levando a uma decisão (MULLER; GUIDO, 2018).

A Figura 1 mostra um exemplo simples hipotético de árvore de decisão para um problema de regressão. As ramificações lógicas são utilizadas para se chegar à predição de volume de vendas de dois perfumes em duas lojas hipotéticas do estado de Minas Gerais. Cada nó (ponto em que uma decisão é tomada) na árvore representa uma pergunta e, por fim, as respostas estão nas folhas (nós terminais que representam os resultados).

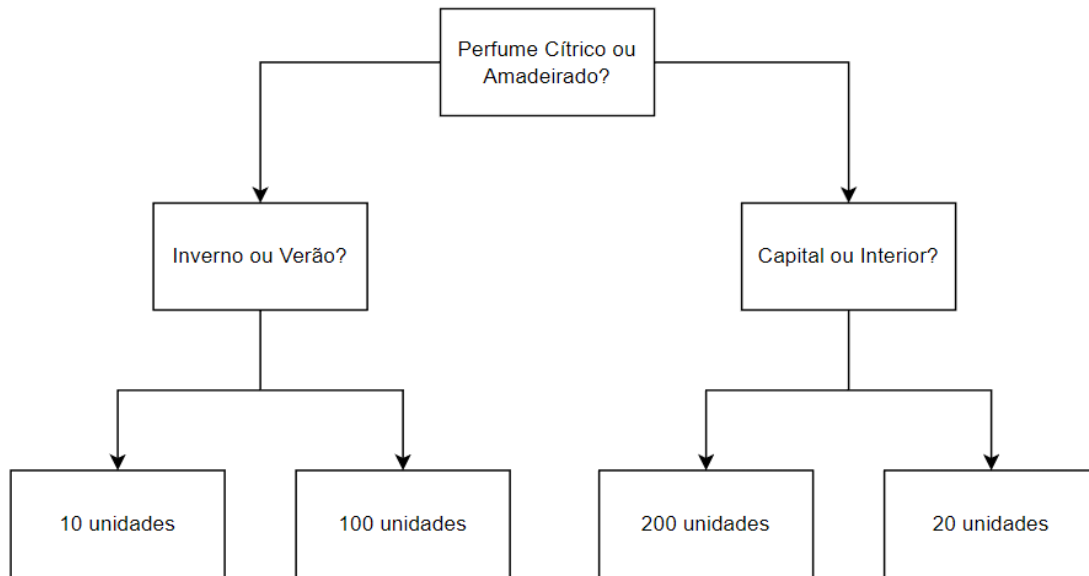


Figura 1: Exemplo de Árvore de Regressão.

Aprender através de uma árvore de decisão significa aprender a sequência de perguntas se/senão que nos leva à folha que apresenta menos erros. As perguntas podem ser ou não na forma de características binárias. Na Figura 1, a título ilustrativo, essas perguntas são binárias. Porém, elas também poderiam ser representadas como características contínuas, como, por exemplo: “O atributo i é maior a ?” (MULLER; GUIDO, 2018)

Em DT para problemas de regressão, a função de custo a ser minimizada usualmente é o erro quadrático. Nesse caso, a média é a constante que minimiza o erro quadrático. Isso significa que a constante associada às folhas de uma árvore deste tipo é a média dos valores do atributo alvo dos exemplos de treinamento que caem na folha (FACELI, 2011). Porém, é possível modificar para outra função, como é o caso do que foi mencionado na Seção 4.6.

A métrica Redução do Desvio Padrão (do inglês, *Standard Deviation Reduction* - SDR) foi proposta para estimar o mérito de uma partição obtida por um teste no valor de uma variável (BREIMAN et al., 1984). A Equação 2 mostra essa métrica.

$$sd(D, Y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2} \quad (2)$$

onde D é o conjunto de exemplos, n é a quantidade de exemplos e y é o alvo.

Considerando um teste hipotético h_A , é possível estimar a redução em variância obtida pela aplicação do teste h_A , segundo a Equação 3. Testando o atributo A com, por exemplo, $A \leq a$, serão gerados dois subconjuntos, D_L e D_R , com tamanhos respectivos de n_L e n_R , de forma que $n = n_L + n_R$ (FACELI, 2011).

$$SDR(h_A) = sd(D, y) - \frac{n_L}{n} \times sd(D_L, y) - \frac{n_R}{n} \times sd(D_R, y) \quad (3)$$

onde A é o atributo, D é o conjunto de exemplos e y é o alvo.

Existem vantagens na utilização de métodos baseados em árvore. Dentre elas, é possível destacar (FACELI, 2011):

- **Flexibilidade:** Não assumem nenhuma distribuição para os dados, são métodos não-paramétricos;
- **Robustez:** É menos sensível a *outliers*;
- **Seleção de atributos:** O processo de construção de uma árvore seleciona os atributos mais relevantes que serão usados na modelagem;
- **Interpretabilidade:** Todas as decisões são baseadas nos valores dos atributos;
- **Eficiência:** É um algoritmo guloso construído de cima para baixo com a estratégia “dividir para conquistar”, sem o processo de mover-se ao longo de uma borda em direção à raiz da árvore.

A Seção 2.3 apresenta os conceitos de GBT, que, como o próprio nome diz, é composto por árvores de decisão.

2.3 Gradient Boosting

Ensembles são métodos que combinam vários modelos de AM para criar modelos mais poderosos. Existem muitos modelos na literatura que pertencem a essa categoria, sendo um deles o de *Gradient Boosting*, que utiliza árvores de decisão como blocos de construção (MULLER; GUIDO, 2018).

A principal ideia por trás do GB é combinar modelos simples, conhecidos como “aprendizes fracos”. Cada uma dessas árvores conseguem fornecer boas previsões apenas

em uma parcela dos dados e, assim, mais árvores são adicionadas para melhorar o desempenho de maneira iterativa (MULLER; GUIDO, 2018). O *Gradient Boosting* cria modelos de formas sequenciais baseando-se no conceito de que os modelos posteriores devem corrigir os erros dos modelos antecessores e os dados desajustados pelos modelos antecessores devem ser ponderados pelos modelos posteriores. As GBTs tornaram-se metodologias muito utilizadas nos últimos anos para séries temporais na indústria e em competições de DS (NIELSEN, 2019).

É possível especificar arbitrariamente a função de perda e os modelos de aprendizado básico sob demanda. Dada alguma função de perda específica $\Psi(y, f)$ e/ou um aprendizado base personalizado $h(x, \theta)$, pode ser difícil obter a solução para as estimativas de parâmetros. Para isso, uma nova função $h(x, \theta_t)$ pode ser proposta. É possível formular o algoritmo de GB, de acordo com o que está descrito no Algoritmo 1 (NATEKIN; KNOLL, 2013).

Algorithm 1 Algoritmo GBT.

Entrada:

- Dados de entrada: $(x, y)_{i=1}^N$
- Número de iterações M
- Escolha da função de perda $\Psi(y, f)$
- Escolha do modelo de aprendizado base $h(x, \theta)$

Algoritmo:

1. Inicialize \hat{f}_0 com uma constante.
 2. Para $t = 1$ até M :
 - (a) Calcule o gradiente negativo $g_t(x)$
 - (b) Ajuste uma nova função de aprendizado base $h(x, \theta_t)$
 - (c) Encontre o melhor tamanho da etapa de descida do gradiente ρ_t :

$$\rho_t = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^N \Psi \left[y_i \hat{f}_{t-1}(x_i + \rho h(x_i, \theta_t)) \right]$$
 - (d) Atualize a estimativa da função:

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$$
 3. Fim da iteração.
 4. Fim do algoritmo.
-

Os métodos de *Gradient Boosting* apresentam algumas propriedades de interesse, uma vez que incorporam as características vantajosas das árvores, enquanto mitigam muitas das características desfavoráveis. A robustez figura entre as mais notáveis. Outra vantagem relevante consiste na seleção interna de atributos, uma vez que as árvores

demonstram ser bastante resilientes à inclusão de variáveis de entrada irrelevantes (FRIEDMAN, 2001).

A principal limitação inerente aos modelos de árvore única reside na imprecisão, que decorre da natureza de suas aproximações constantes por partes (especialmente para árvores menores), da instabilidade (especialmente para árvores maiores) e da prevalência de interações de ordem elevada. Todas essas considerações são atenuadas pela abordagem *Boosting*, uma vez que seus procedimentos geram aproximações constantes por partes com uma granularidade significativamente mais refinada (FRIEDMAN, 2001).

Os métodos GB são mais sensíveis às configurações de parâmetros. Esse ajuste nos hiperparâmetros pode acarretar em um treinamento mais demorado, mas, se definidos corretamente, podem fornecer uma melhor precisão (MULLER; GUIDO, 2018). Os principais hiperparâmetros que interferem no resultado de GBTs podem ser visualizados na Tabela 1.

Parâmetro	Descrição
n_estimators	Número máximo de árvores.
max_depth	Profundidade máxima da árvore.
learning_rate	Taxa de aprendizado que controla o quão fortemente cada árvore tenta corrigir os erros das árvores anteriores.

Tabela 1: Principais hiperparâmetros de métodos GB.

Os parâmetros **n_estimators** e **learning_rate** estão interconectados, visto que com uma menor taxa de aprendizado, mais árvores são necessárias para construir um modelo de complexidade similar. O parâmetro **max_depth** pode reduzir a complexidade de cada árvore, reduzindo o superajustamento (do inglês, *overfitting*) (MULLER; GUIDO, 2018).

A etapa de Otimização de Hiperparâmetros foi descrita na Seção 4.6, em que os principais parâmetros aqui descritos, juntamente com outros parâmetros, serão ajustados. A Seção 2.4 detalha os algoritmos baseados em *Gradient Boosting* utilizados neste estudo, que são: CatBoost, LightGBM e XGBoost.

2.4 Algoritmos Utilizados

Nesta seção, serão apresentados os conceitos gerais dos métodos utilizados para análise comparativa no estudo. As Seções 2.4.1, 2.4.2 e 2.4.3 tratam dos algoritmos de *Gradient Boosting* estudados: CatBoost, LightGBM e XGBoost, respectivamente. Em seguida, a Seção 2.4.4 fundamenta o algoritmo utilizado baseado em séries temporais, o Prophet.

2.4.1 CatBoost

O CatBoost foi um algoritmo criado com o intuito de lidar com atributos categóricos, tanto que seu nome vem de Impulsionamento Categórico (do inglês, *Categorical Boosting*). Além disso, utiliza uma estratégia que reduz o sobreajuste e permite usar todo o conjunto de dados para treinamento. Essa estratégia é aplicada no cálculo dos valores das folhas ao selecionar a estrutura da árvore (DOROGUSH; ERSHOV; GULIN, 2018).

Para lidar com as variáveis categóricas, é realizada uma permutação aleatória do conjunto de dados e, para cada exemplo, é calculado o valor médio do rótulo para o exemplo com o mesmo valor de categoria colocado antes do dado na permutação. Seja $\sigma = (\sigma_1, \dots, \sigma_n)$ a permutação, então $x_{\sigma_p, k}$ é substituído pela Equação 4 (DOROGUSH; ERSHOV; GULIN, 2018).

$$\frac{\sum_{j=1}^{p-1} [x_{\sigma_j, k} - x_{\sigma_p, k}] Y_{\sigma_j} + a \cdot P}{\sum_{j=1}^{p-1} [x_{\sigma_j, k} - x_{\sigma_p, k}] + a} \quad (4)$$

Na Equação 4, é adicionado um valor prévio de P e um parâmetro $a > 0$, que é o peso do valor prévio. Essa é uma prática comum que ajuda a reduzir o ruído obtido nas categorias de baixa frequência. Para tarefas de regressão, a técnica padrão para o cálculo do valor prévio é obter o valor médio do rótulo no conjunto de dados (DOROGUSH; ERSHOV; GULIN, 2018).

Ao fazer combinações de atributos, o CatBoost considera as combinações de forma gulosa. Nenhuma combinação é considerada para a primeira divisão na árvore. Para as próximas divisões, o algoritmo faz a união de todas as combinações e atributos categóricos presentes na árvore atual com todos os atributos categóricos do conjunto de dados. Os valores de combinação são convertidos em números em tempo real. O algoritmo também gera combinações de atributos numéricos e categóricos da seguinte forma: todas as divisões selecionadas na árvore são consideradas categóricas com dois valores e usadas em combinações da mesma forma que as categóricas (DOROGUSH; ERSHOV; GULIN, 2018).

O CatBoost usa uma abordagem para calcular valores de folha que permite usar várias permutações. No algoritmo, são geradas s permutações aleatórias do conjunto de dados de treinamento para aumentar sua robustez. É feita uma amostragem de uma permutação aleatória e os gradientes são obtidos com base nela. São utilizadas diferentes permutações para treinar modelos distintos, e, assim, não há sobreajuste (DOROGUSH; ERSHOV; GULIN, 2018).

Para validar o algoritmo, foram utilizados os conjuntos de dados *Epsilon* e *Criteo*. Segundo os experimentos, o CatBoost foi 60 vezes mais rápido que o LightGBM e 25 vezes

mais rápido que o XGBoost (DOROGUSH; ERSHOV; GULIN, 2018). Os algoritmos citados serão abordados em sequência, nas Subseções 2.4.2 e 2.4.3.

2.4.2 LightGBM

Os métodos mais antigos de GBTs, como XGBoost (Seção 2.4.3) e pGBRT, apresentam eficiência e escalabilidade insatisfatórias quando a dimensão dos atributos é alta e a quantidade de dados é grande. Isso acontece principalmente porque, ao fazer a varredura de todas as instâncias dos dados para estimar os ganhos de informação de todas as divisões possíveis, é preciso muito tempo. Sendo assim, o LightGBM surgiu com o intuito de resolver esse problema. O nome veio de Máquina de Impulsionamento Leve de Gradiente (do inglês, *Light Gradient Boosting Machine*). Foram propostas duas técnicas: amostragem unilateral baseada em gradiente (do inglês, *Gradient-based One-Side Sampling* - GOSS) e pacote de atributos exclusivos (do inglês, *Exclusive Feature Bundling* - EFB) (KE et al., 2017).

No GOSS, há redução significativa das instâncias de dados que possuem pequenos gradientes, sendo que apenas o restante é utilizado para estimar o ganho de informação. Como as instâncias de dados com gradientes maiores desempenham um papel mais importante nesse cálculo de ganho de informação, o GOSS consegue obter uma estimativa precisa do ganho de informação com bases de dados de tamanho muito menor (KE et al., 2017).

Resumidamente, o GOSS é feito mantendo todas as instâncias com grandes gradientes e executando amostragem aleatória nas instâncias com pequenos gradientes. Ao computar o ganho de informação, o GOSS introduz uma constante para multiplicar as instâncias de dados com pequenos gradientes, de forma que a influência na distribuição dos dados seja compensada. Primeiro, as instâncias de dados são classificadas de acordo com o valor absoluto de seus gradientes e as instâncias que possuem valores superiores a $a \times 100\%$ são selecionadas. Em seguida, o GOSS faz uma amostragem aleatória $b \times 100\%$ das instâncias que pertencem ao restante dos dados. Depois, os dados da amostra com pequenos gradientes são amplificados por uma constante $\frac{1-a}{b}$ ao calcular o ganho de informação. Fazendo isso, as instâncias subtreinadas terão mais foco sem que a distribuição original de dados tenha grande alteração (KE et al., 2017).

Dados de alta dimensão são geralmente muito esparsos. Isso oferece a possibilidade de projetar uma abordagem que consiga reduzir o número de atributos sem grandes perdas. Em dados esparsos, há muitos atributos que são mutuamente exclusivos, ou seja, que não assumem valores diferentes de zero simultaneamente. Sendo assim, o objetivo principal do EFB é reduzir o número de atributos (KE et al., 2017).

No EFB, os atributos mutualmente exclusivos são agrupados. Através de um algo-

ritmo de varredura de atributos, é possível construir os mesmos histogramas dos atributos dos “pacotes” ou grupos como aqueles dos atributos individuais. Assim, a complexidade da construção do histograma muda de $O(\#dados \times \#atributo)$ para $(\#dados \times \#pacote)$, enquanto $\#pacote \ll \#atributo$. Encontrar o agrupamento ideal de atributos exclusivos é NP-difícil, mas é possível atingir uma taxa de aproximação aceitável com um algoritmo guloso e, conseqüentemente, reduzir o número de atributos sem que a precisão da determinação do ponto de divisão seja muito prejudicada. Como consequência, o tempo de treinamento é acelerado (KE et al., 2017).

Para validar a técnica, foram feitos experimentos em cinco conjuntos de dados disponíveis publicamente: *Allstate*, *Flight Delay*, *LETOR*, *KDD10* e *KDD12*. Os experimentos mostraram que o LightGBM acelera o processo de treinamento do GB convencional em mais de 20 vezes, ainda alcançando a mesma precisão (KE et al., 2017).

O último algoritmo de GB utilizado no presente estudo será detalhado na Seção 2.4.3, em sequência.

2.4.3 XGBoost

XGBoost vem do nome Aumento de Gradiente Extremo (do inglês, *eXtreme Gradient Boosting*). O XGBoost se tornou um algoritmo reconhecido devido às competições e desafios do Kaggle, podendo ser aplicado a uma ampla gama de problemas. A etapa de Engenharia de Atributos e o domínio do problema foram primordiais para o sucesso das modelagens dessas competições, mas o fato de o XGBoost ser escolhido para resolução mostra o seu impacto (CHEN; GUESTRIN, 2016).

A proposta do algoritmo é que ele tenha capacidade de lidar com dados esparsos e também tenha um esboço de quantil ponderado teoricamente justificado para aprendizado aproximado. Além disso, o XGBoost é um sistema escalável de ponta a ponta para *tree boosting* que utiliza menos recursos (CHEN; GUESTRIN, 2016).

É comum que os dados de entrada sejam esparsos em problemas da vida real e, por isso, é importante tornar o algoritmo ciente do padrão de esparsidade nos dados. Para isso, o XGBoost adiciona uma direção padrão em cada nó da árvore. Quando falta um valor na matriz esparsa x , a instância é classificada na direção padrão, que é aprendida a partir dos dados (DOROGUSH; ERSHOV; GULIN, 2018).

Um dos principais problemas no aprendizado de árvores é encontrar a melhor divisão. Para isso, um algoritmo de descoberta de divisão enumera todas as divisões possíveis em todos os atributos. Esse é o chamado algoritmo guloso exato, que é computacionalmente custoso. Para fazer isso de forma eficiente, o algoritmo deve primeiro classificar os dados, de acordo com os valores dos atributos, e visitar os dados em ordem de classificação para acumular as estatísticas de gradiente. Executar isso com eficiência

quando os dados não cabem inteiramente na memória não é possível. Isso também acontece no cenário distribuído. Para suportar o aumento efetivo do GB nessas duas configurações, é necessário um algoritmo aproximado. O XGBoost suporta com eficiência tanto o algoritmo guloso exato na configuração de uma única máquina, quanto o algoritmo aproximado com métodos de proposta locais e globais para todas as configurações (DOROGUSH; ERSHOV; GULIN, 2018).

Um passo importante no algoritmo aproximado é propor pontos de divisão candidatos. Geralmente, os percentis de um atributo são utilizados para fazer com que os candidatos distribuam uniformemente os dados. No entanto, não há esboço de quantil existente para os conjuntos de dados ponderados. Por conseguinte, a maioria dos algoritmos aproximados existentes recorre à classificação em um subconjunto aleatório de dados com chance de falha ou à heurísticas que não possuem garantia teórica. Para solucionar esse problema, o XGBoost possui um algoritmo de esboço de quantil ponderado distribuído capaz de lidar com dados ponderados com uma garantia teórica demonstrável (DOROGUSH; ERSHOV; GULIN, 2018).

Após a conclusão da exposição dos princípios fundamentais dos métodos de Gradient Boosting, a seção subsequente, designada como seção 2.4.4, apresenta uma análise detalhada do método de série temporal empregado no estudo, o Prophet.

2.4.4 Prophet

O Prophet usa um modelo de série temporal decomponível com três componentes principais do modelo: tendência, sazonalidade e feriados, que são combinados na Equação 5, a seguir (TAYLOR; LETHAM, 2018):

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (5)$$

onde $g(t)$ é a função de tendência que modela as alterações não periódicas no valor da série temporal, $s(t)$ representa as mudanças periódicas, $h(t)$ representa os efeitos dos feriados e o termo de erro ϵ_t representa quaisquer mudanças idiossincráticas, que não são acomodadas pelo modelo.

No Prophet, o problema de previsão é enquadrado como um exercício de ajuste de curva, o que é diferente dos modelos de séries temporais que explicitamente levam em conta a estrutura de dependência temporal nos dados. Essa formulação oferece vantagens práticas, tais como (TAYLOR; LETHAM, 2018):

- A sazonalidade com vários períodos é facilmente configurada;

- As medições não precisam ser espaçadas regularmente e também não é necessário interpolar os valores ausentes;
- O ajuste é muito rápido;
- Os parâmetros são interpretáveis.

Em suma, o Prophet é um modelo de regressão simples e modular que geralmente funciona bem com parâmetros padrão, que permite medir e rastrear a precisão das previsões, sinalizando possíveis melhorias incrementais. Modelos simples e ajustáveis e o monitoramento de desempenho dimensionável permitem que um grande número de analistas preveja um grande número e uma variedade de séries temporais. Por esse motivo, é considerado uma previsão em escala (TAYLOR; LETHAM, 2018).

A Seção 2.5 detalha a métrica utilizada para avaliar as performances dos algoritmos de CatBoost, LightGBM, XGBoost e Prophet utilizados neste estudo.

2.5 Avaliação dos Modelos

MAPE é o erro percentual médio absoluto (do inglês, *Mean Absolute Percentage Error*). Ele é amplamente utilizado para avaliar modelos de previsão de demanda, como é o caso de alguns trabalhos relacionados mostrados na Seção 3. O cálculo é feito conforme mostrado na Equação 6.

$$\frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (6)$$

onde t refere-se ao instante de tempo, n é o n ésimo instante de tempo, A é o valor real e F refere-se ao valor previsto.

Porém, o MAPE é uma métrica que apresenta uma fraqueza em casos de vendas muito baixas, podendo apresentar resultados enganosos. Supondo que, em uma segunda-feira, em uma loja específica, tenha sido vendida apenas uma unidade de um perfume X. Porém, a previsão era de vender 2 unidades naquele dia. Isso significa um erro de 100%, como é possível ver na Equação 7. Um produto que teve pouco volume de vendas naquele dia “explodiu” a métrica de erro.

$$\frac{1}{1} \sum_{t=1}^1 \left| \frac{1 - 2}{1} \right| = 100\% \quad (7)$$

Para o estudo em questão, o WAPE se torna uma melhor opção. WAPE é o erro percentual médio ponderado (do inglês, *Weighted Average Percentage Error*) e seu cálculo pode ser visualizado na Equação 8.

$$\frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|} \quad (8)$$

onde t é o instante de tempo, n refere-se ao enésimo instante de tempo, A é valor real e F é o valor previsto.

O WAPE pode variar de 0% até qualquer valor positivo. Por exemplo, supondo que era esperado que uma loja de shopping vendesse 300 unidades de um hidratante, mas vendeu somente 100: nesse caso, o erro foi de 200%. 100% não é o erro máximo possível, mas o erro mínimo é 0%, ou seja, quanto mais próximo de 0, melhor foi o resultado da previsão.

2.6 Testes de Hipótese

Nesta Seção, serão fundamentados os testes de hipótese que serão utilizados para avaliar a diferença significativa entre os modelos em comparação. Os testes executados serão apresentados posteriormente nas Seções 5.1.5 e 5.4 do Capítulo 5.

Para garantir que os testes de hipótese não paramétricos são os adequados de acordo com as distribuições dos dados e, especificamente, do WAPE calculado (conforme explicado anteriormente na Seção 2.5), é preciso averiguar se a distribuição dos dados foge da normalidade. Por esse motivo, o Teste de Normalidade de Shapiro Wilk (seção 2.6.1) será realizado. Em seguida, o Teste de Kruskal-Wallis será fundamentado na Seção 2.6.2.

2.6.1 Teste de Shapiro-Wilk

Este teste univariado de normalidade foi proposto por Samuel Sanford Shapiro e Martin Wilk. As hipóteses para o teste de normalidade de Shapiro-Wilk são (SHAPIRO; WILK, 1965):

H_0 : A amostra provém de uma população normal.

H_a : A amostra não provém de uma população normal.

Dada uma amostra aleatória completa de tamanho n , x_1, x_2, \dots, x_n , o teste procede da seguinte forma (SHAPIRO; WILK, 1965):

1. Ordene as observações para obter uma amostra ordenada $y_1 \leq y_2 \leq \dots \leq y_n$;

2. Calcular:

$$S^2 = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (x_i - \bar{x})^2$$

3. (a) Se n for par, $n = 2k$, calcular:

$$b = \sum_{i=1}^k a_{n-i+1} (y_{n-i+1} - y_i)$$

(b) Se n for ímpar, $n = 2k + 1$, então, calcular:

$$b = a_{n-i+1}(y_n - y_1) + \dots + a_{k+2}(y_{k+2} - y_k)$$

4. Calcular $W = b^2/S^2$. Valores pequenos de W são significantes, ou seja, indicam não normalidade.

Os testes de normalidade foram aplicados para cada método (CatBoost, LightGBM, XGBoost, Prophet) / horizonte (curto, médio, longo) de erro de previsão, o WAPE.

2.6.2 Teste de Kruskal-Wallis

O teste de Kruskal-Wallis é um método para testar se três ou mais amostras independentes foram retiradas de populações com médias iguais, tendo como exigência apenas que a variável básica tenha distribuição contínua (STEVENSON et al., 1981).

O passo a passo para a realização do teste de Kruskal-Wallis pode ser resumido da seguinte maneira (STEVENSON et al., 1981):

1. Organizar todos os valores por ordem crescente, de modo que cada valor tenha uma posição atribuída;
2. Formular as hipóteses:
 - H_0 : Todas as médias são iguais.
 - H_a : Ao menos um par de médias é diferente.

3. Utilizar a estatística de teste:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N+1)$$

em que N é o tamanho da amostra combinada de diversas amostras; n_i é o número de indivíduos no grupo i e R_i é a média das posições no grupo i .

Após a realização do teste de normalidade, tendo a confirmação que todas as amostras (método / horizonte) de WAPE não possuem distribuição normal, o teste de Kruskal-Wallis pode ser utilizado de maneira adequada.

Em sequência, o Capítulo 3 apresenta cinco trabalhos relacionados com similaridades e disparidades com o estudo em questão.

3 TRABALHOS RELACIONADOS

A previsão de demanda a longo prazo é complexa e a robustez é um dos principais desafios enfrentados (ZHOU et al., 2022). Isso porque a confiabilidade do modelo em detectar a sazonalidade pode ser impactada por padrões encontrados em dados ruidosos (WU et al., 2021).

Um estudo utilizou os dados de vendas do Walmart, disponíveis no banco de dados da competição M5 do Kaggle, com o objetivo de comparar os resultados de diferentes modelos na previsão de 28 dias subsequentes. A base de dados inclui cinco anos de vendas de três categorias para três estados dos Estados Unidos da América. No comparativo, foram analisados tanto modelos mais tradicionais, quanto modelos mais recentes, e foi feito um *benchmark* de seus desempenhos. Foram eles: ARIMA (do inglês, *Autoregressive Integrated Moving Average*, tradução: média móvel integrada autorregressiva), LightGBM que foi desenvolvido pela Microsoft e o Prophet, que foi desenvolvido pela Meta (antigo Facebook). Os resultados mostraram que o modelo ARIMA superou ambos os modelos, de acordo com a raiz do erro médio quadrado (do inglês, *root mean squared error* - RMSE), mas o LightGBM alcançou menor custo computacional (HASAN et al., 2022).

Outro estudo recente (SPILOTIS et al., 2022) também utilizou os dados da competição M5 do Kaggle, e teve como objetivo comparar diferentes modelos de árvore: Árvore de Decisão, Floresta Aleatória (do inglês, *Random Forest*) e GB. O MAPE foi utilizado como métrica e o modelo que apresentou o menor erro foi o de GB (5%). A importância da etapa de Engenharia de Atributos (do inglês, *feature engineering*) para um resultado satisfatório de modelos em árvore foi ressaltada. Além disso, esse tipo de modelo é interpretável porque suas regras podem ser visualizadas, fazendo com que os tomadores de decisão consigam compreender os fatores que influenciam nas previsões (SPILOTIS et al., 2022).

Um trabalho do mesmo ano (LI, 2022) também mencionou o quanto a Engenharia de Atributos pode influenciar na análise preditiva de vendas. Para a criação de variáveis, foram utilizados algoritmos genéticos. Quatro algoritmos baseados em árvore foram implementados para prever as vendas do mês seguinte: *Random Forest*, XGBoost, LightGBM e CatBoost. A performance dos quatro modelos foi comparada através do MAPE e da raiz do erro percentual médio quadrado (do inglês, *root mean squared percentage error* - RMSPE), antes e depois da Engenharia de Atributos. Em sequência, os modelos que obtiveram os menores erros (*Random Forest* e LightGBM) foram comparados em previsões em horizontes temporais mais distantes: para o segundo e o terceiro mês. Ao chegar ao fim dos experimentos, foi constatado que o modelo de *Random Forest* foi o que obteve melhores resultados (LI, 2022).

Lopes (2022) fez seu estudo focado em um comparativo do quanto a performance dos modelos poderia variar em múltiplos horizontes de tempo. Os dados utilizados foram todas as transações com cartão capturadas por uma empresa adquirente brasileira, durante o período de 1º de janeiro de 2014 a 31 de julho de 2018, em um total de 1.673 observações diárias. As previsões foram feitas para quatro diferentes horizontes temporais: para o dia seguinte e para vários dias à frente (sete, trinta e noventa).

Salientou-se a importância de se levar em consideração feriados, datas especiais e efeitos diversos do calendário. Foram avaliados modelos tradicionais de previsão de séries temporais (Naïve, HoltWinters, TBATS e SARIMA) e também modelos de AM e Aprendizado Profundo (do inglês, *Deep Learning*) conhecidos. Os demais algoritmos que fizeram parte do experimento foram: Perceptron Multicamadas (do inglês, *Multilayer Perceptron* - MLP), (do inglês, Memória de Longo-Curto Prazo *Long-Short Term Memory* - LSTM), XGBoost e o modelo Prophet do Facebook. Dentre todos os modelos, o que obteve o menor erro no horizonte temporal mais distante foi o Prophet, com um RMSE de 17,2 (LOPES, 2022).

Seguindo também o objetivo de analisar a consistência das análises preditivas em diferentes horizontes temporais, Baržić et al. (2022) utilizou o XGBoost para fazer previsões diárias de volume de vendas para uma semana, totalizando sete previsões. Os dados utilizados foram fornecidos por uma indústria parceira, com os registros de vendas de cinco produtos. Os erros foram medidos através do erro médio absoluto (do inglês, *mean absolute error* - MAE) e do erro absoluto médio ponderado (do inglês, *weighted mean absolute error* - WMAE). O WMAE médio foi de 2,76.

A Tabela 2 mostra um resumo dos estudos aqui mencionados. Para os trabalhos de Lopes (2022) e Baržić et al. (2022), que tiveram foco em fazer as previsões em múltiplos horizontes temporais, foram inseridas as métricas correspondentes aos horizontes de tempo mais distantes.

A principal similaridade entre o presente estudo e os trabalhos relacionados é que todos têm como objetivo fazer a previsão de vendas dentro do setor do varejo. Além disso, todos os estudos utilizam pelo menos um algoritmo de *Gradient Boosting*. Entretanto, há também diferenças.

Este estudo difere dos trabalhos relacionados em diferentes pontos. Os trabalhos de Hasan et. al (2022) e Spiliotis et al. (2022) não abordaram previsões em diferentes horizontes de tempo. Li (2022) realizou um experimento para horizontes temporais mais distantes (para até três meses à frente), mas não foi o foco do estudo. Baržić et al. (2022) também comparou previsões de volume de vendas em instantes temporais diferentes, mas o período máximo foi de uma semana à frente.

Autores	Objetivo Geral	Base de Dados	Método	Métrica	Avaliação
(HASAN et al., 2022)	Prever as vendas do Walmart para os 28 dias subsequentes.	Competição M5 Kaggle (Walmart)	ARIMA LightGBM Prophet	RMSE	1,09 1,18 6,96
(SPILIOTIS et al., 2022)	Comparar algoritmos baseados em árvore para prever vendas.	Competição M5 Kaggle (Walmart)	Gradient Boosting Random Forest Decision Tree	MAPE	5,00% 5,90% 7,50%
(LI, 2022)	Ressaltar a importância da etapa de feature engineering para algoritmos que são baseados em árvore.	Kaggle Rossmann Store	Random Forest LightGBM CatBoost XGBoost	MAPE	10,20% 10,80% 13,50% 13,80%
(LOPES, 2022)	Comparar a performance de diferentes algoritmos de <i>Machine Learning</i> e séries temporais para previsões de venda em quatro horizontes de tempo.	Transações com cartão durante o período de janeiro de 2014 e julho de 2018.	Prophet XGBoost LSTM MLP HoltWinters SARIMA Naïve TBATS	RMSE	17,20 17,55 22,34 23,09 23,61 24,05 39,67 48,12
(BARŽIĆ et al., 2022)	Fazer previsões diárias durante uma semana.	Dados de vendas fornecidos por um parceiro.	XGBoost	MAE	2,44

Tabela 2: Resumo dos trabalhos relacionados.

O trabalho que possui maior similaridade com o presente estudo é o de Lopes (2022). Porém, ainda assim, há uma distinção importante a ser observada. Enquanto o trabalho de Lopes (2022) teve um horizonte temporal máximo equivalente a três meses, o presente estudo tem uma previsão de longo prazo maior: o equivalente a aproximadamente sete meses à frente.

Outra diferenciação significativa entre o estudo atual e as pesquisas anteriores é a presença de fatores contextuais e características de dados distintos. O estudo abrange diversas campanhas de atração de consumidores e considera produtos que não são classificados como *commodities* essenciais.

Enquanto os estudos mencionados lidaram com dados de séries temporais com espaçamento regular (como dados diários ou mensais, por exemplo), o contexto de tempo da Empresa X é alinhado com as estratégias de negócios, os chamados ciclos. Por esse motivo, os métodos GB foram implementados segundo as estratégias da Empresa X, na granularidade temporal de ciclo. Entretanto, como os ciclos não têm intervalos iguais, isso representa um desafio ao empregar métodos tradicionais de séries temporais, tornando-se necessário realizar algumas adaptações para a utilização de métodos dessa natureza. O Capítulo 4 detalha como foi feito o pré-processamento para os algoritmos estudados, de forma que a metodologia seja utilizada tanto para métodos GB, quanto para algoritmos de ST.

4 METODOLOGIA

A metodologia deste trabalho foi baseada na Metodologia CRISP-DM (*do inglês, Cross Industry Standard Process for Data Mining*), que significa Processo Padrão entre Indústrias para Mineração de Dados. É muito utilizada para projetos de Ciência de Dados e é uma metodologia que não está atrelada a nenhum software ou técnica, sendo independente (KELLEHER; TIERNEY, 2018).

A metodologia CRISP-DM consiste originalmente em seis etapas (entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação) (KELLEHER; TIERNEY, 2018), mas foi necessário fazer algumas adaptações para atender às necessidades deste trabalho. A Figura 2 é uma representação visual da metodologia que foi utilizada com as devidas adaptações.

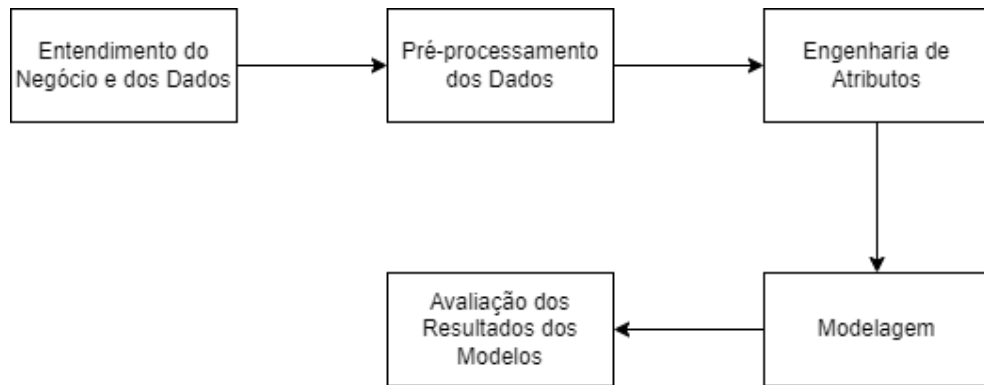


Figura 2: Metodologia geral utilizada.

1. **Entendimento do negócio e dos dados:** Na Capítulo 1, foi definido o problema a ser resolvido, de acordo com o contexto do negócio. Na sequência, os dados disponíveis para realizar a previsão de demanda foram mostrados na Seção 4.1.
2. **Pré-processamento dos dados:** Nesta etapa, foi feita toda a limpeza dos dados e formatação. A Seção 4.4 mostra esse passo com detalhes.
3. **Engenharia de Atributos:** A Seção 4.5 detalha esta etapa. A criação de novas variáveis é especialmente importante para métodos baseados em árvore, como já foi mencionado no Capítulo 3.
4. **Modelagem:** Esta etapa corresponde ao treinamento dos algoritmos de *Gradient Boosting* (CatBoost, LightGBM e XGBoost) e o Prophet. Além disso, é detalhada a técnica de validação e otimização de hiperparâmetros para os modelos em que essa etapa foi aplicada. A Seção 4.6 mostra os pormenores de Modelagem.
5. **Avaliação dos resultados dos modelos:** Por fim, é feita uma análise para visualizar os erros dos modelos e compará-los. Além disso, foram realizados testes

de hipótese (conforme previamente explicado na Seção 2.6) nos comparativos entre modelos para garantir que há diferença significativa entre eles. Esta etapa está descrita no Capítulo 5.

As Seções 4.2 e 4.3 mostram informações mais detalhadas sobre as ferramentas e máquina que foram utilizadas para a execução deste trabalho, tais como *hardware*, linguagens de programação, bibliotecas e ambiente em nuvem.

As Figuras 3 e 4 detalham melhor as metodologias utilizadas para os métodos de GB e séries temporais, respectivamente.

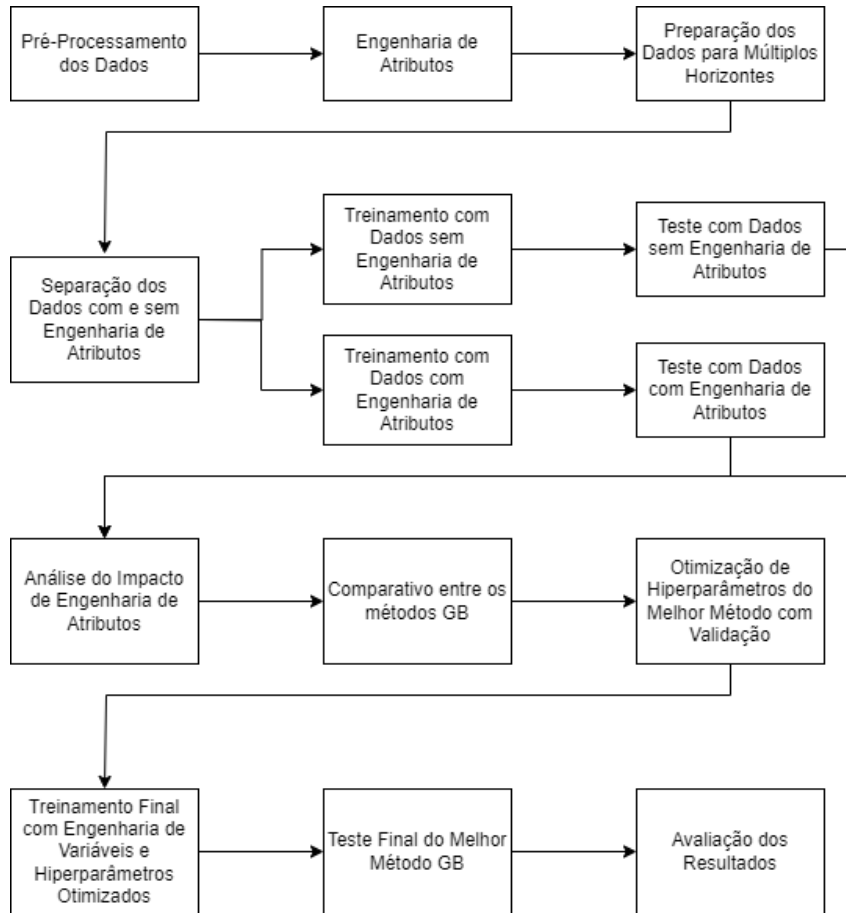


Figura 3: Detalhamento da metodologia utilizada para os algoritmos de GB.

A principal diferença entre as duas metodologias apresentadas nas Figuras 3 e 4 é na terceira etapa, preparação dos dados para múltiplos horizontes temporais (para métodos GB) e recorte dos horizontes temporais (para o Prophet). Para os métodos GB, essa preparação consiste em separar os conjuntos de dados em seis partes: uma para cada horizonte temporal e para cada canal de vendas. No caso da etapa de recorte dos horizontes temporais ilustrada na Figura 4, refere-se à separação do conjunto principal de dados considerando 30 semanas de previsão (sendo que, 30 semanas, é o equivalente ao longo prazo).

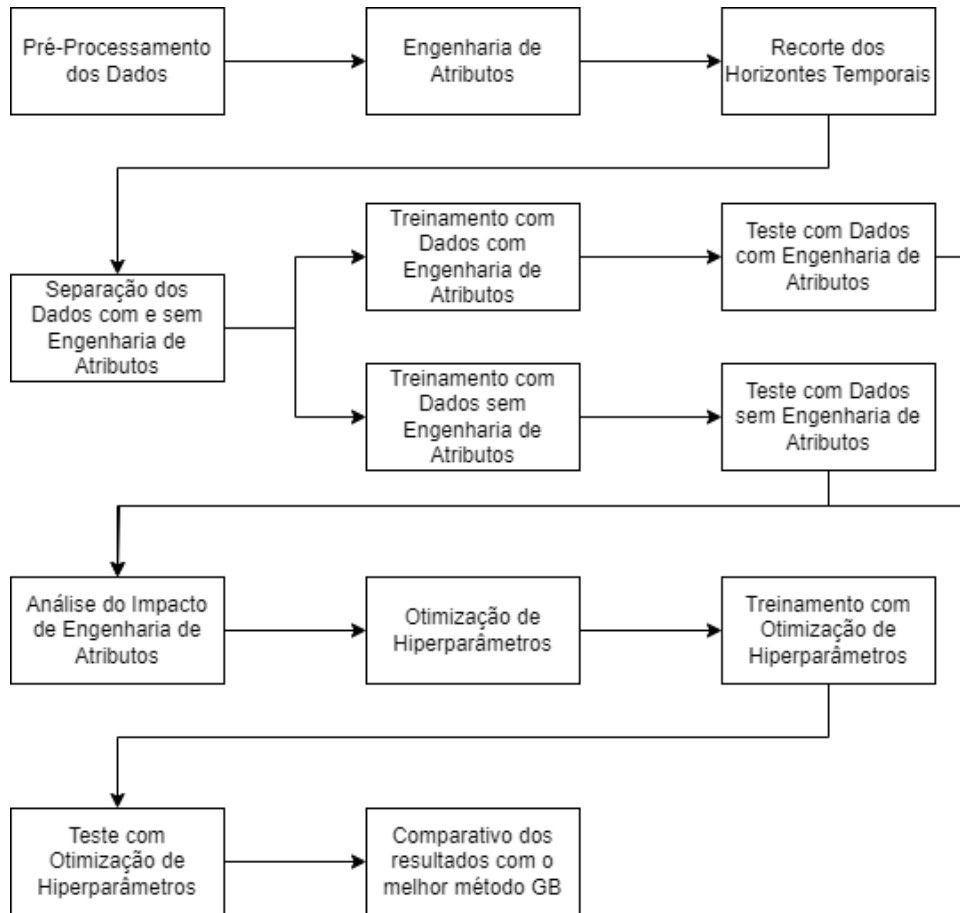


Figura 4: Detalhamento da metodologia utilizada para algoritmo de séries temporais: Prophet.

4.1 Base de Dados

A base de dados utilizada neste estudo foi fornecida pela Empresa X do mercado de CFT, contendo 22 variáveis e aproximadamente 93M de instâncias. Os dados são referentes às vendas entre os anos de 2018 e 2023. Os dados de 2018 foram utilizados somente para a criação de *features* de lag (Seção 4.5) e os dados de 2023 foram utilizados apenas como base de teste.

As variáveis brutas disponíveis estão indicadas na Tabela 3.

Variável	Descrição
Data de Venda	Data em que o produto foi vendido.
Ciclo	Período do ano, sendo que cada ano é composto por 17 ciclos.
Descrição do Ciclo	Definição do tipo de ciclo.
Data do início do ciclo	Data em que o ciclo foi iniciado.
Data do fim do ciclo	Data em que o ciclo termina.
Canal	Canal de vendas, podendo ser Venda Direta ou Loja.
UF	Unidade Federativa do Brasil.
Regional	Regional de campanha.
Categoria	Categoria do produto.
Subcategoria	Subcategoria do produto.
Marca	Marca do produto.
Código Produto	Código do Produto/SKU.
Descrição Produto	Descrição do Produto/SKU.
Volume	Variável-Alvo. Quantidade de produtos vendidos.
Valor Praticado	Valor de venda com descontos.
Valor Tabela	Valor de venda sem descontos.
Valor de Desconto	Valor do desconto.
Porcentagem de Desconto	Porcentagem do desconto.
Preço Unitário	Preço unitário considerando os descontos.
Promoção	Indica se o produto está promocionado ou não.
Foco	Indica se o produto está em foco de campanha ou não.
Subfoco	Indica se o produto está em subfoco de campanha ou não.

Tabela 3: Variáveis disponíveis na base de dados.

4.2 Hardware

Para rodar os experimentos, foi utilizado o *JupyterLab* no *workbench* do GCP (*Google Cloud Platform*). O *hardware* utilizado foi de 16 vCPUs e 60 GB RAM.

4.3 Tecnologias Utilizadas

A linguagem de programação empregada foi Python 3.7, juntamente com a linguagem de consulta estruturada SQL (*BigQuery* no GCP). Dentro do Python, as seguintes bibliotecas foram utilizadas:

- **pandas**, para manipulação dos dados através de *dataframes*;
- **numpy**, para realização de operações matemáticas e utilização de *arrays*;
- **matplotlib.pyplot**, para visualização dos dados através de gráficos;
- **seaborn**, para visualização dos dados também;
- **google.cloud**, para conexão dos dados com o *Google BigQuery*;

- **lightgbm**, para execução os experimentos utilizando o algoritmo LightGBM;
- **xgboost**, para execução os experimentos utilizando o algoritmo XGBoost;
- **catboost**, para execução os experimentos utilizando o algoritmo CatBoost;
- **sklearn**, para realização de aplicações gerais de *Machine Learning*;
- **optuna**, para otimização de hiperparâmetros;
- **prophet**, para modelagem utilizando o método;
- **statsmodel**, para análise de séries temporais;
- **scipi**, para realização de testes de hipótese;
- **time**, para cálculo do tempo de execução.

4.4 Pré-Processamento dos Dados

Nesta etapa, os dados foram primeiramente formatados, de acordo com cada variável (seja como *string*, *float*, inteiro ou outro formato). No geral, como os dados já foram estruturados anteriormente pelo time de Engenharia de Dados da empresa do estudo, não foi necessário fazer grandes tratativas nessa etapa.

Após a formatação, foi necessário analisar os valores nulos e a sua proporção em relação à base de dados completa e, em sequência, limpá-los. A Tabela 4 mostra as variáveis em que foi encontrado algum valor nulo, a porcentagem de valores nulos e como os dados foram tratados.

Variável	Porcentagem de nulos	Tratativa
Valor de Desconto	0,093%	Linhas removidas.
Porcentagem de Desconto	0,093%	Linhas removidas.
Valor Praticado	0,099%	Linhas removidas.
Foco	53,49%	Substituídos por 0.
Subfoco	44,43%	Substituídos por 0.

Tabela 4: Dados nulos e tratativas.

Além dos valores nulos que foram encontrados, a coluna Valor Praticado também apresentou 0,028% de instâncias com valores negativos que também tiveram suas linhas removidas. No total, a base foi reduzida em 0,22% após fazer a limpeza dos valores nulos e menores que zero. Podemos observar que, a porcentagem de nulos Valor de Desconto e Porcentagem de Desconto é exatamente a mesma, o que indica a possibilidade de se tratarem das mesmas instâncias, ou seja, mesmas linhas. As variáveis binárias que

apresentaram valores nulos (Foco e Subfoco) foram substituídas por 0, visto que valores nulos significam que o evento não ocorreu naquelas instâncias.

Além da limpeza dos dados nulos, foi feita também uma limpeza para remover produtos, categorias, subcategorias e marcas que estão desativados, visto que não há necessidade de fazer previsão de demanda de produtos que não fazem mais parte do portfólio da empresa. Nessa limpeza, 2,12% da base foi removida.

Após a limpeza dos dados, as bases foram preparadas, de acordo com as necessidades de cada algoritmo.

4.4.1 Preparação dos Dados para Algoritmos de Gradient Boosting

Para os algoritmos de GB, a modelagem foi feita no nível ciclo/canal/regional/categoria/subcategoria/marca. Ao agrupar os dados nessa granularidade, a base de dados foi reduzida para aproximadamente 1,62M de instâncias. O nível temporal de ciclo foi escolhido para a utilização desses algoritmos para respeitar a estratégia de negócio. Os ciclos são os períodos em que são feitas as mesmas campanhas. Cada ano possui 17 ciclos e a mediana de duração é de 21 dias. Essa duração pode variar a cada ciclo e a cada ano.

4.4.2 Preparação dos Dados para Algoritmo de Séries Temporais

Para os métodos de séries temporais, diferente dos algoritmos de GB, não é possível respeitar a estratégia de negócio no nível temporal de ciclo, visto que os ciclos não são igualmente espaçados. Sendo assim, os dados foram agrupados no nível semana/canal/regional/categoria/subcategoria/marca. Após o agrupamento, a base de dados apresentou aproximadamente 7,34M de instâncias. Sendo assim, os resultados para o Prophet foram obtidos utilizando uma base de dados em uma granularidade diferente da empregada para os métodos GB.

Para fazer o agrupamento a nível de semana, foi necessário criar uma variável que indica qual foi a data da segunda-feira referente à semana daquele dia em que foi efetuada a venda. Por exemplo, caso uma venda tenha sido efetuada no dia 10/02/23 (sexta-feira), a referência daquela semana será 06/02/23 (segunda-feira).

Como os ciclos podem começar e acabar no meio de uma semana, foi necessário também adaptar uma regra para incluir as informações dos ciclos e das campanhas (promoções, foco e subfoco). Nessa regra adaptada, foram consideradas as informações correspondentes ao ciclo de maior número de dias daquela semana. Por exemplo, se, em uma semana Z, 2 dias de vendas fazem parte do ciclo 202202 e 5 dias de vendas fazem parte do ciclo 202203, as informações utilizadas serão referentes ao ciclo 202203.

Além do agrupamento e das adaptações relacionadas a ele, também foi necessário fazer uma tratativa diferente para a variável categórica de descrição de ciclo. Para utilizar o método prophet, foi criada uma tabela auxiliar contendo a data da segunda-feira e a descrição do ciclo correspondente àquela semana, visto que o prophet possui um parâmetro chamado *holidays* em que é possível utilizar variáveis sazonais.

4.5 Engenharia de Atributos

Na etapa de Engenharia de Atributos, diferentes variáveis foram construídas para serem utilizadas como entrada para os modelos. Conforme estudos citados no Capítulo 3, esta etapa é muito importante, especialmente para algoritmos baseados em árvore.

A Tabela 5 detalha as variáveis que foram criadas considerando os produtos e suas características, além de promoções e campanhas.

Variáveis de origem	Variáveis criadas	Descrição
Produto, Ciclo, Categoria, Subcategoria, Marca	Quantidade de produtos no portfólio (1 variável)	Quantidade de produtos naquela categoria/subcategoria/marca.
Promoção, Produto, Ciclo, Categoria, Subcategoria, Marca, Regional	Produtos em promoção (1 variável)	Quantidade de produtos em promoção dentro da granularidade.
Porcentagem de Desconto, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Quantidade de produtos sem desconto (1 variável criada)	Quantidade de produtos que estão fora de promoção.
Porcentagem de Desconto, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Faixa de desconto (3 variáveis criadas)	Quantidade de produtos que pertencem a cada faixa de desconto (10%, 20%, 30%).
Foco, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Produtos em foco (1 variável)	Quantidade de produtos em foco dentro da granularidade.
Subfoco, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Produtos em subfoco (1 variável)	Quantidade de produtos em subfoco dentro da granularidade.

Tabela 5: Variáveis criadas considerando produtos, campanhas e promoções.

Na Tabela 6, foram mostradas as variáveis que calculam a fatia de mercado (do inglês, *market share*) em diferentes níveis. O *market share* foi calculado considerando o ano imediatamente anterior. Sendo assim, para os modelos de *Gradient Boosting*, o

cálculo foi feito considerando o período equivalente a um ano anterior ao ciclo utilizado na predição, ou seja, 17 ciclos anteriores a ele.

Para o modelo de séries temporais, o Prophet, o cálculo foi feito considerando as 52 semanas anteriores (duração de um ano) mais 30 semanas (horizonte do longo prazo - equivalente a 10 ciclos). Como um ciclo tem duração média de 3 semanas, a conversão de ciclo para semana do horizonte máximo é igual a 30.

Variáveis de origem	Variáveis criadas	Descrição
Categoria, Volume	Market share de volume da categoria (1 variável)	Volume de vendas de uma categoria em relação ao volume total no período do ano imediatamente anterior.
Categoria, Valor Tabela	Market share do valor tabela da categoria (1 variável)	Valor tabela de uma categoria em relação ao valor tabela total no período do ano imediatamente anterior.
Subcategoria, Volume	Market share de volume da subcategoria (1 variável)	Volume de vendas de uma subcategoria em relação ao volume total no período do ano imediatamente anterior.
Subcategoria, Valor Tabela	Market share do valor tabela da subcategoria (1 variável)	Valor tabela de uma subcategoria em relação ao valor tabela total no período do ano imediatamente anterior.
Marca, Volume	Market share de volume da marca (1 variável)	Volume de vendas de uma marca em relação ao volume total no período do ano imediatamente anterior.
Marca, Valor Tabela	Market share do valor tabela da marca (1 variável)	Valor tabela de uma marca em relação ao valor tabela total no período do ano imediatamente anterior.
Regional, Volume	Market share de volume da regional (1 variável)	Volume de vendas de uma regional em relação ao volume total no período do ano imediatamente anterior.
Regional, Valor Tabela	Market share do valor tabela da regional (1 variável)	Valor tabela de uma regional em relação ao valor tabela total no período do ano imediatamente anterior.

Tabela 6: Variáveis construídas calculando o *market share* do último ano.

As médias móveis calculadas para cada método foram adaptadas de forma a evitar o *data leakage*, o que não significa que são idênticas. Para os métodos GB, foi criada

uma base de dados para cada horizonte temporal, enquanto, para o Prophet, há uma base única devido à natureza do método. Para ilustrar melhor como o cálculo foi feito para GB e para o Prophet, será utilizado como exemplo uma previsão do horizonte de médio prazo. Supondo que deseja-se fazer a previsão de 5 ciclos à frente / 15 semanas à frente. Para fazer a previsão do ciclo 202206 no GB, utilizamos as informações históricas do ciclo 202201. O *market share*, então, foi calculado utilizando os ciclos 202101 até 202117 (duração de um ano). Fazendo a analogia com o Prophet, deseja-se fazer a previsão da semana 15 de 2022. Porém, como o Prophet possui uma base única em que será necessário fazer a previsão da janela completa de 30 semanas (correspondente ao longo prazo), as semanas que serão usadas serão entre as semanas 37 de 2020 e 37 de 2021. Sendo assim, a tratativa para tornar os métodos de regressão e séries temporais comparáveis é desafiadora devido à natureza de cada método. As adaptações são aproximadas, porém não completamente equivalentes.

As Tabelas 7 e 8 contém 21 variáveis de *lag* que foram criadas para os modelos GB e ST, respectivamente. Especialmente para os métodos de *Gradient Boosting*, essas variáveis são essenciais para prever a demanda já que GB são algoritmos de previsão e não são apropriados para séries temporais. As *lags* são as variáveis que permitem que o algoritmo de previsão considere a variável tempo, ou seja, elas extraem propriedades que podem ordenar os dados (BERGMEIR; BENÍTEZ, 2012). Em outras palavras, são atributos que refletem atrasos temporais. Por exemplo, neste estudo que utiliza dados de vendas em diferentes momentos no tempo, um exemplo de atributo de *lag* seria o volume de vendas de um mês anterior.

Para os modelos GB (Tabela 7), as *lags* foram calculadas considerando o ciclo utilizado na predição. É fundamental enfatizar que não há salto de tempo na previsão do primeiro horizonte temporal, o que significa que o ciclo a ser previsto e o ciclo usado na previsão são idênticos.

Para o Prophet (Tabela 8), os cálculos foram feitos convertendo ciclos para semanas, sendo que cada ciclo tem duração média de 3 semanas. Posto isso, para fazer a equivalência de *lags* entre os métodos GB e ST, a lógica geral utilizada é descrita na Equação 9.

$$l = (3 * c) + h \quad (9)$$

onde l é a *lag* em semanas, c é a quantidade de ciclos equivalente, utilizada conforme a Tabela 7 e h é o horizonte máximo aplicado aos métodos de séries temporais.

Variáveis de origem	Variáveis criadas	Descrição
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Lag de vendas de 1 ano (1 variável)	Volume de vendas daquele ciclo no ano anterior.
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Lag de vendas de 1 a 10 ciclos (10 variáveis)	Quantidade vendida em 1, 2, 3, 4, 5, 6, 7, 8, 9 e 10 ciclos anteriores.
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Média móvel do volume de vendas de 5 e 10 ciclos (2 variáveis)	Média móvel da quantidade vendida considerando os 5 e 10 ciclos anteriores.

Tabela 7: Variáveis de *lag* criadas para os modelos GB considerando o ciclo utilizado na predição.

Variáveis de origem	Variáveis criadas	Descrição
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Lag de vendas de 1 ano (1 variável)	Volume de vendas daquela semana no ano anterior.
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Lag de vendas de (3, 6, 9, 12, 15, 18, 21, 24, 27 e 30) + 30 semanas (10 variáveis)	Quantidade vendida em 33, 36, 39, 42, 45, 48, 51, 54, 57 e 60 semanas anteriores.
Volume de vendas, Ciclo, Produto, Categoria, Subcategoria, Marca, Regional	Média móvel do volume de vendas de 45 e 60 semanas (2 variáveis)	Média móvel da quantidade vendida considerando as 15+30 e 30+30 semanas anteriores.

Tabela 8: Variáveis de *lag* criadas para o método de séries temporais.

A Tabela 9 mostra variáveis que foram construídas com base nas informações dos ciclos para os métodos de GB, sendo que três deles são usados na previsão do próximo ciclo e seis são usados nos outros horizontes. Outro atributo relacionado às informações do ciclo que foi utilizado como entrada para o modelo é a descrição do ciclo. Ela não foi mostrada na Tabela 9 porque é uma *feature* original dos dados. Para 5 e 10 ciclos à frente, há também a descrição do ciclo a ser previsto.

Variáveis de origem	Variáveis criadas	Descrição
Data de início de ciclo, Data de fim de ciclo	Duração do ciclo (1 variável)	Duração do ciclo em dias, comumente 21 dias. (Ciclo utilizado na predição)
Data de início de ciclo, Data de fim de ciclo	Duração do ciclo (1 variável)	Duração do ciclo em dias, comumente 21 dias. (Ciclo a ser predito)
Ciclo	Ano do ciclo, Número do ciclo (2 variáveis)	Informação sobre o ciclo utilizado na predição.
Ciclo	Ano do ciclo, Número do ciclo (2 variáveis)	Informação sobre o ciclo a ser predito.

Tabela 9: Variáveis construídas com informações dos ciclos para os métodos GB.

A Tabela 10 mostra as variáveis de ciclo que foram aplicadas ao método de séries temporais. É importante lembrar que, conforme foi descrito na Seção 4.4, as informações de ciclos utilizadas são referentes ao ciclo daquela semana, correspondente à maior quantidade de dias.

Variáveis de origem	Variáveis criadas	Descrição
Data de início de ciclo, Data de fim de ciclo	Duração do ciclo (1 variável)	Duração do ciclo daquela semana em dias, comumente 21 dias.
Ciclo	Ano do ciclo, Número do ciclo (2 variáveis)	Informação sobre o ciclo daquela semana.

Tabela 10: Variáveis construídas com informações dos ciclos para o método ST.

Foram criadas *features* automáticas através do método *PolynomialFeatures* da biblioteca *scikit-learn*, utilizando o parâmetro *degree* igual a 2. Algumas dessas *features* estão evidenciadas na Tabela 11. As variáveis de origem utilizadas na construção dessas variáveis automáticas foram: quantidade de produtos no portfólio, quantidade de produtos em promoção, quantidade de produtos sem desconto, quantidade de produtos com 10% de desconto, quantidade de produtos com 20% de desconto, quantidade de produtos com 30% de desconto, quantidade de produtos em foco, quantidade de produtos em subfoco.

Além das *features* mostradas na Tabela 11, foram criadas ainda mais 28 variáveis automáticas utilizando como origem as mesmas *features* e o mesmo parâmetro anteriormente mencionado do método *PolynomialFeatures*. Essas 28 variáveis são os produtos de

cada uma dessas variáveis duas a duas, como, por exemplo, quantidade de produtos em portfólio multiplicada por quantidade de produtos em subfoco. Sendo assim, no total, 36 *features* automáticas foram construídas.

Variáveis de origem	Variáveis criadas	Descrição
Quantidade de produtos em portfólio	Quadrado de produtos em portfólio (1 variável)	Quantidade de produtos do portfólio ao quadrado.
Quantidade de produtos em promoção	Quadrado de produtos em promoção (1 variável)	Quantidade de produtos em promoção ao quadrado.
Quantidade de produtos sem desconto	Quadrado de produtos fora da promoção (1 variável)	Quantidade de produtos fora da promoção ao quadrado.
Quantidade de produtos nas faixas de 10%/20%/30% de desconto	Quadrado de produtos na faixa de 10%/20%/30% de desconto (3 variáveis)	Quantidade de produtos na faixa de 10%/20%/30% de desconto ao quadrado.
Quantidade de produtos em foco	Quadrado de produtos em foco (1 variável)	Quantidade de produtos em foco ao quadrado.
Quantidade de produtos em subfoco	Quadrado de produtos em subfoco (1 variável)	Quantidade de produtos em subfoco ao quadrado.

Tabela 11: Variáveis automáticas construídas utilizando a biblioteca *scikit-learn*.

4.6 Modelagem

Na etapa de modelagem, assim como na etapa de Engenharia de Atributos (Seção 4.5), foram feitas adaptações de forma que os métodos de GB e o método de ST possam ser comparados nos múltiplos horizontes temporais.

Para os métodos de GB, as previsões em horizontes temporais e canais foram feitas separadamente e, sendo assim, existem seis conjuntos de dados: para o de próximo ciclo, para a previsão de cinco ciclos à frente e para a previsão de dez ciclos à frente, sendo também divididos entre os dois canais de venda (Lojas e Venda Direta). Dessa forma, para cada método, foram executados seis modelos de AM.

As variáveis de entrada para a previsão de demanda do próximo utilizando métodos GB são 75 no total, conforme lista abaixo:

- 6 variáveis da granularidade da previsão (ciclo, regional, canal, categoria, subcategoria e marca);

- 1 variável original contendo a descrição do ciclo;
- 3 variáveis construídas considerando informações de ciclo (Tabela 9);
- 8 variáveis criadas considerando produtos, campanhas e promoções (Tabela 5);
- 8 variáveis de *market share* construídas (Tabela 6);
- 13 variáveis de *lag* (Tabela 7);
- 36 variáveis criadas de maneira automática (Tabela 11).

Para as previsões com saltos temporais (5 e 10 ciclos à frente) utilizando métodos GB, as variáveis são 89, sendo elas:

- 6 variáveis da granularidade da previsão (ciclo, regional, canal, categoria, subcategoria e marca);
- 1 variável referente ao ciclo a ser predito;
- 2 variáveis originais contendo a descrição do ciclo utilizado na predição e do ciclo a ser predito;
- 6 variáveis construídas considerando informações de ciclo, tanto para o ciclo utilizado na predição quanto para o ciclo a ser predito (Tabela 9);
- 8 variáveis criadas considerando produtos, campanhas e promoções (Tabela 5) no ciclo utilizado na predição;
- 8 variáveis criadas considerando produtos, campanhas e promoções (Tabela 5) no ciclo a ser predito;
- 8 variáveis de *market share* construídas (Tabela 6);
- 13 variáveis de *lag* (Tabela 7) considerando o ciclo usado na predição;
- 36 variáveis criadas de maneira automática (Tabela 11);
- 1 variável com o volume de vendas do ciclo utilizado na predição.

Para o Prophet, existe um conjunto de dados único. Porém, foi executado um modelo para cada série temporal (cada categoria, subcategoria, marca, regional e canal). Como nos métodos de séries temporais é possível fazer previsões de saltos de tempo, cada série temporal foi treinada para fazer previsões das próximas 30 semanas (o equivalente ao longo prazo) e, assim, foi feito um recorte de cada instante e tempo de interesse para

o cálculo dos erros, sendo eles: semana seguinte (curto prazo), quinze semanas à frente (médio prazo) e trinta semanas à frente (longo prazo).

Para as previsões utilizando o Prophet, as variáveis de entrada são similares às variáveis de entrada para o método GB de curto prazo. No total, existem 76 variáveis, sendo:

- 7 variáveis da granularidade da previsão (semana, ciclo, regional, canal, categoria, subcategoria e marca);
- 1 variável original contendo a descrição do ciclo;
- 3 variáveis construídas considerando informações de ciclo (Tabela 10);
- 8 variáveis criadas considerando produtos, campanhas e promoções (Tabela 5);
- 8 variáveis de *market share* construídas (Tabela 6);
- 13 variáveis de *lag* (Tabela 8);
- 36 variáveis criadas de maneira automática (Tabela 11).

Os primeiros treinamentos e testes de todos os métodos GB (CatBoost, LightGBM e XGBoost) e método ST (Prophet) foram feitos utilizando os hiperparâmetros padronizados de cada método. Os dados de 2019 a 2022 foram utilizados para treinamento e os dados de 2023 foram utilizados como teste. Os métodos GB, que foram construídos na granularidade de ciclo, tiveram 68 ciclos como treinamento e 6 ciclos como teste. Para o Prophet, o conjunto de treinamento possui aproximadamente 208 semanas de treinamento e 30 semanas de teste.

As análises comparativas entre os métodos GB estão descritas na Seção 5.1, em que os erros dos três algoritmos foram comparados nos múltiplos horizontes temporais em cenários distintos. Na Seção 5.4, há um comparativo entre o método GB de melhor performance e o Prophet, com modelos treinados a partir de hiperparâmetros otimizados.

A etapa de Otimização de Hiperparâmetros foi executada somente para o XGBoost, que foi o método GB com melhor performance, e para o Prophet. Nesta parte, a biblioteca *optuna* foi utilizada, com o objetivo de minimizar o erro. Os hiperparâmetros, a descrição e as faixas de valores buscadas na otimização estão descritos na Tabela 12 para o XGBoost e na Tabela 13 para o Prophet. Por motivos de custo computacional, foram feitas 5 tentativas para cada modelo. Todos os demais hiperparâmetros estão de acordo com o padrão de cada método.

Hiperparâmetro	Descrição	Valores
max_leaves	Limita a quantidade máxima de folhas que uma árvore pode ter.	[20, 3000]
max_depth	Limita a profundidade máxima das árvores de decisão.	[3, 12]
min_child_weight	Especifica o número mínimo de amostras necessárias em um compartimento para ser considerado válido em métodos baseados em histograma.	[100, 2000]
num_boost_rounds	Define o número máximo de árvores de decisão a serem construídas.	[80, 300]
learning_rate	Controla a taxa na qual o modelo aprende durante o treinamento.	[0.01, 0.3]

Tabela 12: Faixa de valores para otimização dos hiperparâmetros do XGBoost.

Hiperparâmetro	Descrição	Valores
changepoint_prior_scale	Determina a flexibilidade da tendência e, em particular, o quanto a tendência muda nos pontos de mudança de tendência.	[0.001, 0.5]
seasonality_prior_scale	Controla a flexibilidade da sazonalidade.	[0.01, 10]
holidays_prior_scale	Controla a flexibilidade para se adequar aos efeitos do feriado.	[0.01, 10]
seasonality_mode	Define se a sazonalidade será aditiva ou multiplicativa.	[additive, multiplicative]

Tabela 13: Faixa de valores para otimização dos hiperparâmetros do Prophet.

A Figura 5 mostra como foi feita a separação das bases de dados para treino, validação e teste. Na etapa de Otimização de Hiperparâmetros, foram divididos quatro conjuntos de treinamento e validação, utilizando dados de 2019 a 2022. Para o Prophet, os conjuntos de treinamento possuem 100 semanas e os conjuntos de validação possuem 30 semanas.

Para o XGBoost, os conjuntos de treinamento possuem tamanhos diferentes para cada horizonte: 43 ciclos para o próximo ciclo previsto, 38 ciclos para a previsão de 5 ciclos à frente e 33 ciclos para a previsão de 10 ciclos à frente. Os conjuntos de validação correspondem aos dados mais recentes, imediatamente posteriores aos respectivos conjuntos de treinamento, com tamanho de 6 ciclos.

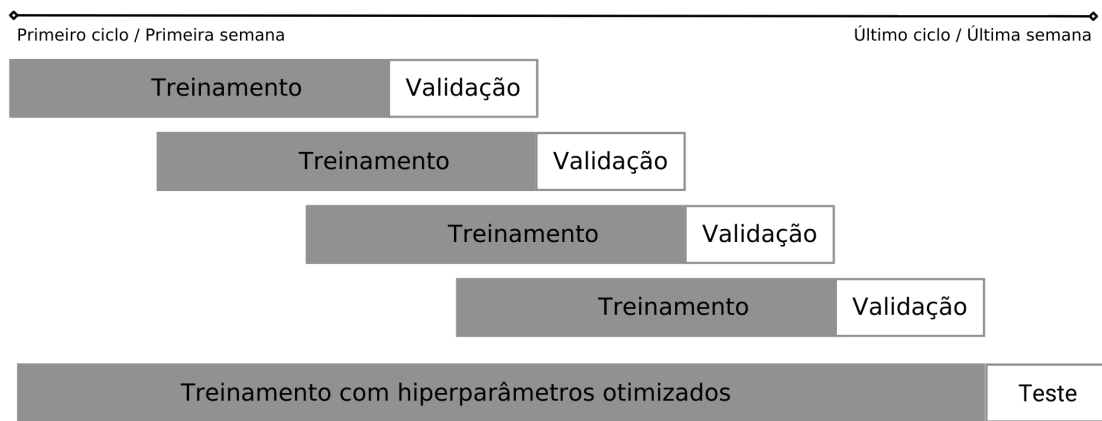


Figura 5: Conjuntos de treino, validação e teste para otimização de hiperparâmetros.

Por fim, foi feito um treinamento para cada modelo com os dados de 2019 até 2022 com os hiperparâmetros otimizados. Os dados do ano de 2023 foram utilizados como teste para avaliar a performance do XGBoost e do Prophet. Para o XGBoost, o conjunto de teste tem a duração de seis ciclos e, para o Prophet, ele tem o tamanho de trinta semanas. As Seções 5.1.7 e 5.3.2 mostram o impacto da etapa da otimização de hiperparâmetros para o XGBoost e para o Prophet, respectivamente.

O Capítulo 5 apresenta uma análise comparativa entre os modelos de *Gradient Boosting* (Seção 5.1), a decomposição da série temporal (Seção 5.2), a análise dos resultados do Prophet (Seção 5.3) e, por fim, um comparativo entre o XGBoost e o Prophet (Seção 5.4).

5 RESULTADOS

Neste capítulo, serão apresentados os resultados para a previsão do volume de vendas de cada categoria/subcategoria/marca por regional e canal (Venda Direta e Loja) em múltiplos horizontes temporais. Para os métodos GB, as previsões foram feitas a nível de ciclo. Para os métodos ST, elas foram feitas a nível de semana.

Os horizontes temporais para cada tipo de método foram:

- **Curto Prazo:** 1 ciclo à frente (métodos de GB) e 1 semana à frente (métodos de ST);
- **Médio Prazo:** 5 ciclos à frente (métodos de GB) e 15 semanas à frente (métodos de ST), o equivalente a aproximadamente 4 meses;
- **Longo Prazo:** 10 ciclos à frente (métodos de GB) e 30 semanas à frente (métodos de ST), o equivalente a aproximadamente 8 meses.

Este capítulo está dividido em 5 seções, conforme descrição abaixo:

- **Seção 5.1:** Comparativo entre os modelos de CatBoost, LightGBM e XGBoost, além do impacto das etapas de Engenharia de Atributos e Otimização de Hiperparâmetros;
- **Seção 5.2:** Análise dos componentes das séries temporais.
- **Seção 5.3:** Análise dos impactos das etapas de Engenharia de Atributos e Otimização de Hiperparâmetros no Prophet;
- **Seção 5.4:** Comparativo entre o XGBoost (melhor método de GB) e Prophet.

5.1 Comparativo entre modelos de Gradient Boosting

Esta seção está subdividida da seguinte forma:

- **Seção 5.1.1:** Análise geral dos erros para cada canal de vendas;
- **Seção 5.1.2:** Análise dos erros na visão ciclo a ciclo em cada canal;
- **Seção 5.1.3:** Comparativo dos erros na visão Regional;
- **Seção 5.1.4:** Comparativo do WAPE utilizando variáveis dos produtos;
- **Seção 5.1.5:** Análise dos testes de hipótese;

- **Seção 5.1.6:** Análise do impacto da etapa de Engenharia de Atributos;
- **Seção 5.1.7:** Análise do impacto da etapa de Otimização de Hiperparâmetros para o método de melhor performance.

5.1.1 Análise dos Erros por Canal de Vendas

Conforme mencionado na Seção 2.5, a métrica utilizada para avaliar a performance dos modelos foi o WAPE. A Tabela 14 mostra os erros gerais de cada modelo, sendo que existe um modelo para cada horizonte, canal e método.

Modelo	Canal	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	Lojas	18,27%	1,42%	12,23%
LightGBM	Lojas	22,77%	2,82%	13,67%
XGBoost	Lojas	10,00%	2,12%	6,40%
CatBoost	Venda Direta	0,83%	10,42%	5,19%
LightGBM	Venda Direta	1,98%	9,78%	5,85%
XGBoost	Venda Direta	0,86%	7,31%	5,41%

Tabela 14: WAPE por canal.

Ao avaliar os resultados na visão por canal e horizonte, é possível notar que o XGBoost obteve a melhor performance em todos os horizontes do canal Loja. Já no canal Venda Direta, nenhum modelo teve a melhor performance em todos os horizontes, mas o CatBoost apresentou os menores erros, tanto no horizonte de curto prazo (próximo ciclo), quanto no horizonte de longo prazo (10 ciclos à frente). O LightGBM, por sua vez, não apresentou o menor erro em nenhum dos canais e nenhum dos horizontes.

Como o propósito do estudo é entender que método é mais robusto para fazer a previsão de volume em múltiplos horizontes temporais, a estabilidade do modelo é importante. Sendo assim, a diferença em pontos percentuais do maior para o menor erro também será um fator considerado dentro das análises para avaliar a consistência de cada método.

Tendo isso em vista, o XGBoost possuiu a menor diferença em pontos percentuais em ambos os canais, sendo 7,88%, para o canal Loja, e 6,45%, para o canal Venda Direta. O CatBoost apresentou a segunda menor amplitude de erros no canal Loja (16,85%), enquanto o LightGBM obteve a segunda menor diferença no canal Venda Direta (6,45%). Em resumo, pode-se dizer que na visão por canal, o XGBoost obteve a melhor performance, em comparação aos outros dois.

5.1.2 Análise dos Erros por Ciclo e por Canal

Na investigação dos resultados das previsões em cada canal e horizonte, a nível de ciclo, os erros serão visualizados ao longo dos seis ciclos da base de teste, que são os seis primeiros ciclos do ano de 2023: 202301 até 202306. A primeira parte da análise será entender, numa visão isolada, por horizonte e canal, como os três métodos se comportam. Em seguida, a estabilidade de cada um dos três métodos, em cada canal, ao longo dos horizontes temporais, será analisada.

Para a previsão de curto prazo, ou seja, para o próximo ciclo, a Figura 6 mostra os resultados para o canal Loja e para o canal Venda Direta. No canal Loja, o XGBoost apresentou o menor erro em praticamente todos os ciclos, com exceção do ciclo 202306, onde o CatBoost apresentou o menor WAPE. Na Figura 6, é possível observar que a diferença de performance entre os modelos não é tão clara em Venda Direta, quanto é em Loja, como já foi observado anteriormente na Seção 5.1.1.

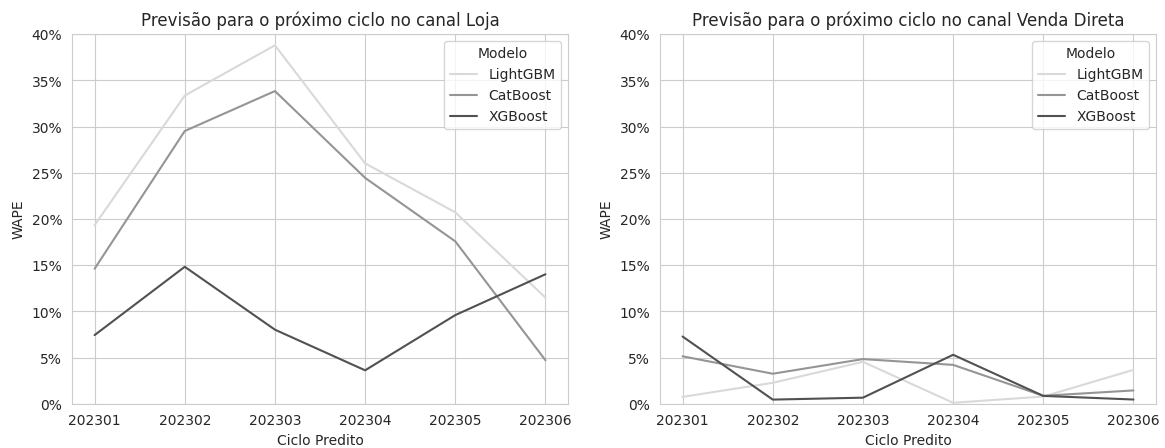


Figura 6: WAPE por ciclo predito para previsão do próximo ciclo.

Segundo o que é mostrado na Figura 6, o LightGBM apresentou o menor erro nos ciclos 202301, 202304 e 202305. O XGBoost apresentou o menor WAPE nos ciclos 202302, 202303 e 202306, enquanto o CatBoost não obteve a melhor performance em nenhum dos ciclos analisados. Em resumo, na previsão de curto prazo, o XGBoost obteve a melhor performance no canal Loja, mas sua performance foi igualmente satisfatória à performance do LightGBM, no canal Venda Direta.

Na previsão de médio prazo, ou seja, para prever 5 ciclos à frente, a Figura 7 apresenta os erros para os dois canais de venda. No canal Loja, o XGBoost apresentou o menor erro nos ciclos 202301 e 202302, o CatBoost apresentou o menor erro nos ciclos 202303 e 202306 e o LightGBM apresentou o menor erro nos ciclos 202304 e 202305. Sendo assim, os três métodos se comportaram de maneira muito similar nesse cenário. No canal Venda Direta, o CatBoost apresentou o menor erro no ciclo 202301 somente. O WAPE

do XGBoost foi menor nos demais ciclos, mostrando que na previsão de médio prazo para o canal Venda Direta, ele teve a melhor performance dentre os três modelos de GB.

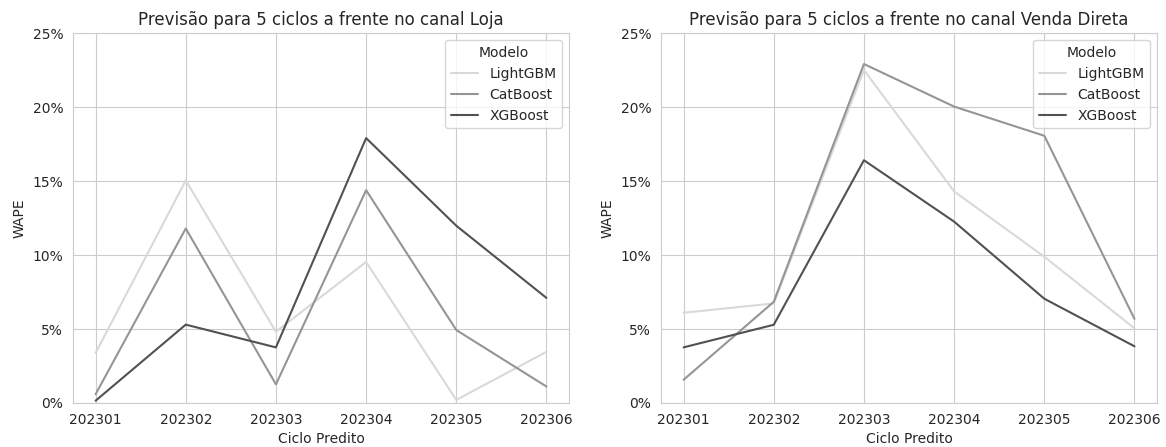


Figura 7: WAPE por ciclo predito para previsão de 5 ciclos à frente.

Na previsão de longo prazo, que é equivalente a aproximadamente uma previsão de 8 meses à frente, a Figura 8 e mostra os erros para os canais Loja e Venda Direta. No canal Loja, o XGBoost apresentou o menor erro em quase todos os ciclos, com exceção do ciclo 202306, onde o LightGBM obteve o menor WAPE. Ainda assim, é possível notar que a diferença entre os erros do XGBoost e do LightGBM no ciclo 202306 não é significativa.

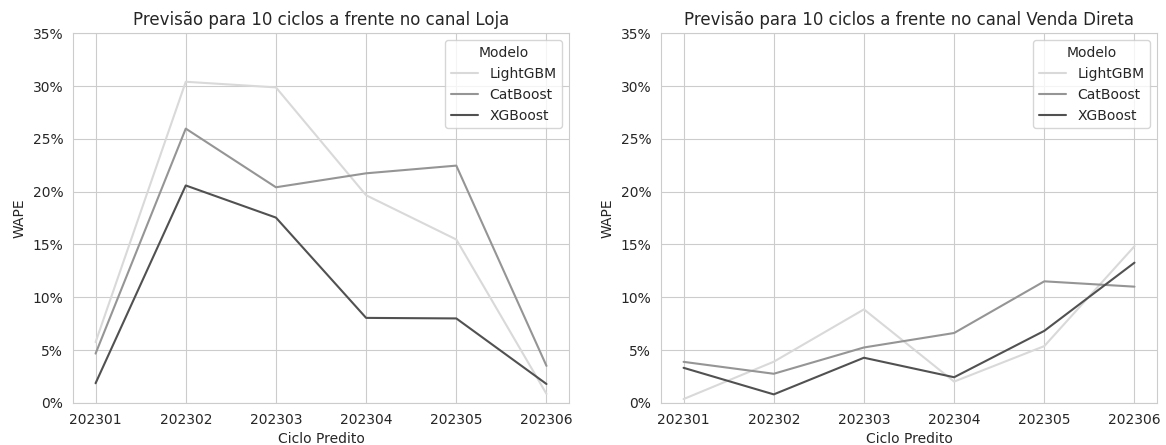


Figura 8: WAPE por ciclo predito para previsão de 10 ciclos à frente.

Na Figura 8, a previsão de 10 ciclos à frente no canal Venda Direta mostra que o LightGBM obteve o menor erro na maioria dos ciclos: 202301, 202304 e 202305. O XGBoost apresentou o menor WAPE nos ciclos 202302 e 202303, enquanto o CatBoost apresentou o menor erro somente no ciclo 202306. Com base nisso, pode-se dizer que o XGBoost obteve a melhor performance na previsão de longo prazo, para o canal Loja, e o LightGBM, para o canal Venda Direta.

Para fazer a análise da performance dos três métodos, de acordo com a estabilidade de cada um deles, ao longo dos horizontes, foram gerados gráficos para cada método em cada canal. A Figura 9 mostra os erros, ao longo dos horizontes para o modelo de CatBoost. Os resultados do LightGBM estão apresentados na Figura 10. Por último, os erros do XGBoost podem ser visualizados na Figura 11.

Analisando os resultados do CatBoost, vemos que os ciclos que apresentaram menor diferença percentual entre os erros foram os ciclos 202306, no canal Loja, e 202301, no canal Venda Direta, o que, curiosamente, são ciclos extremos nas bases de teste. O ciclo 202303 foi o que apresentou maior instabilidade em ambos os canais, sendo que, no canal Loja, teve uma diferença de aproximadamente 30 pontos percentuais e cerca de 20 pontos percentuais no canal Venda Direta. No canal Venda Direta, o horizonte que apresentou os menores erros foi o de curto prazo, enquanto no canal Loja, isso aconteceu para a previsão de médio prazo.

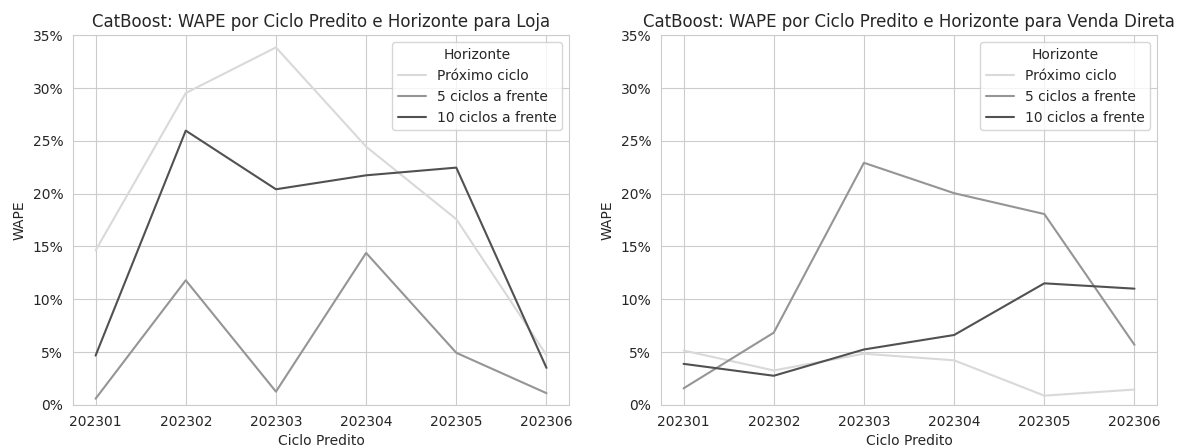


Figura 9: WAPE do CatBoost por ciclo predito e horizonte.

Na Figura 10, pode-se observar que o LightGBM apresentou um comportamento similar ao do CatBoost, no sentido de que o ciclo que apresentou a maior instabilidade foi o 202303, tanto no canal de Loja, quanto no canal Venda Direta. Também similar ao CatBoost, a predição de 5 ciclos à frente foi a que apresentou os menores erros de maneira geral no canal Loja, enquanto no canal Venda Direta, a previsão para o próximo ciclo teve os menores erros em praticamente todos os ciclos.

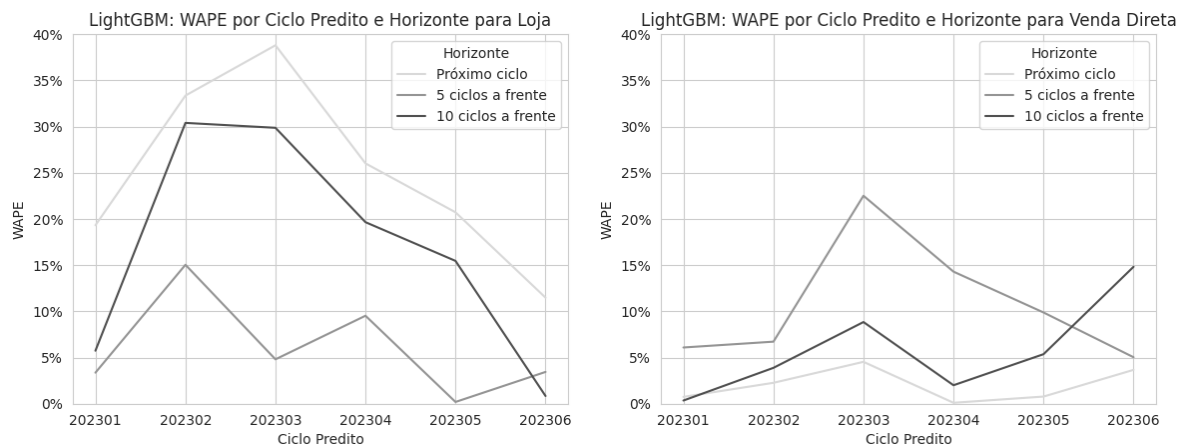


Figura 10: WAPE do LightGBM por ciclo predito e horizonte.

Na Figura 11, é constatado que o ciclo que se apresentou menos estável nas predições em múltiplos horizontes para o XGBoost foi o 202302, com cerca de 15 pontos percentuais de diferença entre os erros. O ciclo 202305 foi o mais estável entre todos, com cerca de somente 5 pontos percentuais de diferença entre os três horizontes. No canal Venda Direta, o ciclo que apresentou maior instabilidade também foi o 202302 e o mais estável foi o ciclo 202301.

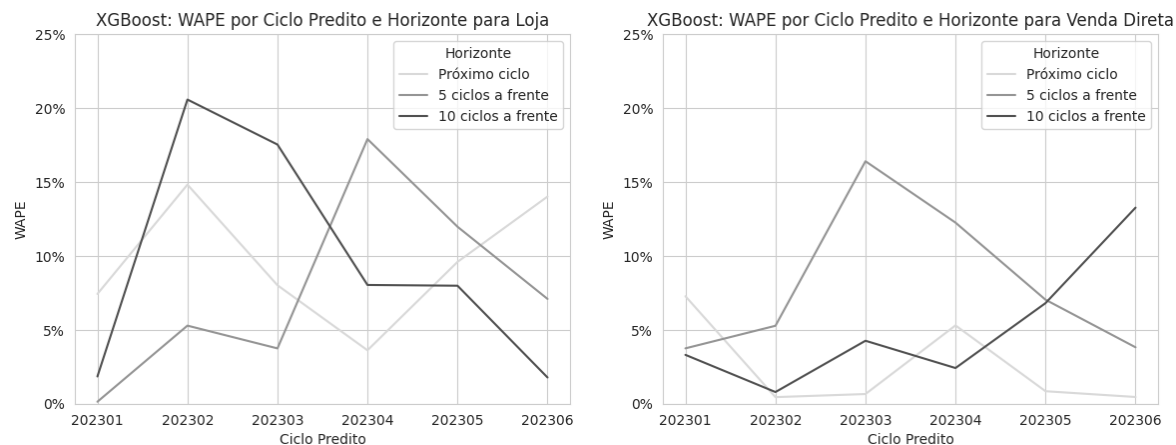


Figura 11: WAPE do XGBoost por ciclo predito e horizonte.

É possível notar que o XGBoost apresentou erros máximos menores que os outros dois métodos em ambos os canais. Enquanto o CatBoost (Figura 9) e o LightGBM (Figura 10) tiveram um erro máximo próximo de 40% no canal Loja, o XGBoost (Figura 11) teve um erro próximo de 20%. No canal Venda Direta, tanto o CatBoost (Figura 9), quanto o LightGBM (Figura 10), tiveram um erro máximo de aproximadamente 25%, enquanto o XGBoost apresentou um erro máximo de cerca de 16% (Figura 11). Com base nessa análise, mais uma vez, o XGBoost se mostrou não só mais consistente ao longo dos horizontes temporais que os demais modelos, mas também obteve uma melhor

performance por ter tido erros menores, em grande parte dos cenários observados.

5.1.3 Análise dos Erros por Regional

Para avaliar os erros a nível de regional, a Tabela 15 mostra o WAPE, para o canal Loja, e a Tabela 16, para o canal Venda Direta. Como a regional de campanha é definida de maneira estratégica, os estados que compõem cada regional não serão identificados por questão de sigilo da informação. São 6 regionais de campanha no país, sendo que a Regional 1 (R1) se refere à regional com maior volume de vendas e a Regional 6 (R6) se refere à regional com o menor volume de vendas no Brasil. É importante ressaltar que não necessariamente a regional que possui maior volume de vendas em Loja é a mesma que possui o maior volume de vendas em Venda Direta, essa nomenclatura foi feita com base no volume total de vendas em ambos os canais no país inteiro.

Na Tabela 15, é possível observar que o XGBoost apresentou o menor WAPE em praticamente todos os cenários, com exceção de apenas três casos na previsão de 5 ciclos à frente, para R1, R3 e R5, sendo que, na R1 e na R3, o LightGBM apresentou o menor WAPE e, na R5, o CatBoost obteve o menor WAPE. Além disso, conforme foi mencionado na Seção 5.1.1, a variação dos erros ao longo dos horizontes também é importante de ser analisada. Analisando por esse lado, o XGBoost novamente se destacou em relação aos outros métodos, que teve a menor diferença em pontos percentuais, em todas as regionais ao longo dos horizontes.

Modelo	Regional	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	R1	16,92%	3,59%	7,42%
LightGBM	R1	19,44%	1,22%	11,95%
XGBoost	R1	8,96%	5,20%	5,26%
CatBoost	R2	2,57%	1,42%	13,62%
LightGBM	R2	2,73%	2,82%	9,97%
XGBoost	R2	5,96%	0,49%	4,08%
CatBoost	R3	20,30%	1,51%	14,93%
LightGBM	R3	24,69%	1,32%	14,52%
XGBoost	R3	13,33%	3,13%	7,13%
CatBoost	R4	19,45%	3,89%	10,55%
LightGBM	R4	20,62%	2,79%	9,85%
XGBoost	R4	7,35%	1,40%	3,08%
CatBoost	R5	19,70%	1,76%	17,61%
LightGBM	R5	28,06%	9,40%	20,39%
XGBoost	R5	12,93%	2,48%	12,99%
CatBoost	R6	27,58%	5,58%	19,51%
LightGBM	R6	40,25%	15,81%	29,67%
XGBoost	R6	19,76%	1,19%	14,84%

Tabela 15: WAPE por regional do canal Loja.

Ao visualizar a Tabela 16, a diferença de performance entre os modelos não é tão perceptível quanto é para Loja, o que corrobora com as análises anteriores apresentadas nas Seções 5.1.1 e 5.1.2. Em Venda Direta, o XGBoost foi o modelo que mais acertou as previsões de curto e médio prazo. Porém, para longo prazo, o CatBoost apresentou o menor erro em mais regionais. Do ponto de vista da análise da variação dos erros ao longo dos três horizontes temporais, o XGBoost apresentou a menor variação em 4 regionais: R1, R2, R4 e R6. O CatBoost foi o modelo que mostrou menor estabilidade em todas as regionais e horizontes, enquanto o LightGBM se mostrou mais estável nas regionais R3 e R5.

Modelo	Regional	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	R1	0,31%	7,83%	4,68%
LightGBM	R1	3,96%	9,87%	7,30%
XGBoost	R1	0,32%	5,55%	4,79%
CatBoost	R2	6,54%	16,30%	13,46%
LightGBM	R2	11,53%	19,15%	17,38%
XGBoost	R2	8,72%	14,92%	13,76%
CatBoost	R3	4,49%	6,21%	1,05%
LightGBM	R3	0,81%	4,20%	0,97%
XGBoost	R3	2,49%	4,15%	0,11%
CatBoost	R4	3,73%	8,11%	1,14%
LightGBM	R4	1,96%	6,40%	0,34%
XGBoost	R4	1,31%	6,63%	0,65%
CatBoost	R5	7,39%	9,20%	0,59%
LightGBM	R5	9,10%	2,95%	5,27%
XGBoost	R5	6,22%	0,02%	3,13%
CatBoost	R6	7,94%	9,65%	0,35%
LightGBM	R6	8,40%	2,82%	3,81%
XGBoost	R6	5,19%	2,16%	0,30%

Tabela 16: WAPE por regional do canal Venda Direta.

Em suma, o XGBoost teve uma performance muito superior aos demais modelos na visão Regional no canal Loja. No canal Venda Direta, a performance dos modelos teve uma variação maior, a depender de qual cenário está sendo analisado. Ainda assim, ao avaliar todos os quesitos em conjunto para a visão Regional, no canal Venda Direta, o XGBoost é relativamente melhor que o LightGBM e o CatBoost.

5.1.4 Análise dos Erros dos Produtos

Para entender como os modelos se comportaram considerando as características dos produtos, serão analisados os erros das três categorias com maior volume de vendas, das três subcategorias com maior volume e também das três marcas que possuem maior

representatividade nas vendas. Por questão de confidencialidade, as categorias serão anonimizadas como C1, C2 e C3, as subcategorias como S1, S2 e S3, e as marcas como M1, M2 e M3. Da mesma forma que os dados foram mascarados na Seção 5.1.3, as nomenclaturas foram construídas considerando o volume total de vendas, para a criação do *ranking* das categorias, subcategorias e marcas e, como os canais de venda possuem comportamentos diferentes, não necessariamente a ordenação é a mesma.

Na Tabela 17, é possível visualizar os erros nas 3 maiores categorias no canal Loja. Cada um dos métodos apresentou o menor erro em pelo menos um horizonte de uma categoria. Na categoria mais vendida, o XGBoost obteve a melhor performance no curto prazo, enquanto o LightGBM apresentou o menor WAPE nos demais horizontes. Na segunda categoria mais vendida, o XGBoost obteve o menor WAPE no curto prazo e ao longo prazo, enquanto o WAPE do CatBoost foi menor no médio prazo. Na terceira categoria mais vendida, mais uma vez, o XGBoost obteve o menor wape no horizonte de curto e longo prazo e, no médio prazo, o LightGBM obteve a melhor performance. Em questão de consistência dos modelos nos múltiplos horizontes temporais, o XGBoost se mostrou mais estável que o CatBoost e o LightGBM.

Modelo	Categoria	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	C1	16,39%	8,86%	9,26%
LightGBM	C1	17,20%	0,51%	3,60%
XGBoost	C1	4,46%	7,23%	6,93%
CatBoost	C2	24,72%	2,47%	11,74%
LightGBM	C2	27,18%	5,49%	19,18%
XGBoost	C2	13,40%	4,47%	10,55%
CatBoost	C3	31,99%	3,40%	14,97%
LightGBM	C3	29,81%	0,44%	22,27%
XGBoost	C3	23,99%	3,60%	10,13%

Tabela 17: WAPE por categoria do canal Loja.

A Tabela 18 apresenta os erros das 3 categorias mais vendidas na visão do canal Venda Direta. Na categoria mais vendida, o CatBoost apresentou o menor erro na previsão do próximo ciclo e de 10 ciclos à frente, enquanto na previsão de 5 ciclos à frente, o método que melhor performou foi o XGBoost. Na segunda categoria mais vendida, o XGBoost apresentou o menor WAPE em todos os horizontes temporais. Na terceira categoria mais representativa, o CatBoost apresentou o melhor resultado apenas para o curto prazo e o XGBoost obteve o menor WAPE nos demais horizontes. O LightGBM não apresentou o menor erro em nenhuma categoria e nenhum horizonte. Em questão de estabilidade, o XGBoost foi o método mais consistente nas duas primeiras categorias e o LightGBM apresentou maior estabilidade para a terceira categoria mais vendida.

Modelo	Categoria	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	C1	7,87%	14,82%	11,11%
LightGBM	C1	10,10%	17,80%	15,29%
XGBoost	C1	8,97%	12,13%	13,34%
CatBoost	C2	7,36%	10,58%	6,49%
LightGBM	C2	5,84%	9,46%	7,89%
XGBoost	C2	4,55%	7,31%	4,02%
CatBoost	C3	6,81%	10,76%	4,04%
LightGBM	C3	10,29%	11,48%	3,95%
XGBoost	C3	23,56%	6,90%	1,72%

Tabela 18: WAPE por categoria do canal Venda Direta.

Na Tabela 19, estão descritos os erros nas subcategorias S1, S2 e S3 para o canal de lojas. Na previsão de curto prazo, o XGBoost obteve a melhor performance em todas as subcategorias. O LightGBM apresentou o menor erro para a subcategoria mais vendida, tanto na previsão de médio prazo, quanto na de longo prazo, enquanto o mesmo aconteceu para o XGBoost, quando é analisada a segunda subcategoria mais vendida. Para a terceira categoria mais vendida, a previsão de 5 ciclos à frente que obteve o menor WAPE foi a do XGBoost, enquanto na previsão de 10 ciclos à frente, o melhor modelo foi o LightGBM. O CatBoost não obteve o menor erro em nenhum dos cenários. Como o LightGBM apresentou erros máximos acima de 20% em mais de um cenário e maior WAPE do XGBoost foi no mínimo de 10 pontos percentuais menor (10,07%), o XGBoost se mostrou uma alternativa na previsão a nível de subcategoria para o canal Loja.

Modelo	Subcategoria	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	S1	14,93%	10,12%	6,85%
LightGBM	S1	14,43%	3,96%	1,09%
XGBoost	S1	3,28%	7,75%	5,81%
CatBoost	S2	17,07%	0,34%	8,39%
LightGBM	S2	20,81%	1,91%	15,35%
XGBoost	S2	6,83%	0,33%	6,52%
CatBoost	S3	19,78%	6,08%	14,41%
LightGBM	S3	22,96%	9,37%	8,69%
XGBoost	S3	6,79%	5,80%	10,07%

Tabela 19: WAPE por subcategoria do canal Loja.

Na Tabela 20, podemos ver que o CatBoost teve uma performance bem diferente no canal Venda Direta, quando comparada com o canal Loja (Figura 19). O CatBoost obteve a melhor performance na previsão de curto prazo em todas as 3 subcategorias mais vendidas no canal Venda Direta. Além disso, também foi o modelo que melhor performou em todos os horizontes para a terceira subcategoria mais vendida. O LightGBM, por outro lado, não apresentou o menor erro em nenhuma subcategoria e nenhum horizonte. O XG-

Boost apresentou o menor WAPE na previsão de médio prazo para as duas subcategorias mais vendidas e para a previsão de 10 ciclos à frente da subcategoria S2. Ao comparar a estabilidade dos dois modelos que melhor performaram, O CatBoost apresentou a maior estabilidade a médio prazo (diferença de 5,52 pontos percentuais) e a longo prazo (diferença de 6,53 pontos percentuais), enquanto o XGBoost obteve maior consistência na previsão de curto prazo (diferença de 4,3 pontos percentuais). Nesse cenário analisado, o CatBoost foi levemente melhor que o XGBoost e o LightGBM não se mostrou uma boa opção.

Modelo	Subcategoria	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	S1	5,22%	12,69%	8,75%
LightGBM	S1	8,09%	15,56%	13,12%
XGBoost	S1	5,55%	6,59%	9,85%
CatBoost	S2	5,29%	10,81%	6,02%
LightGBM	S2	8,50%	11,55%	8,47%
XGBoost	S2	6,67%	8,65%	2,65%
CatBoost	S3	11,74%	18,03%	14,29%
LightGBM	S3	12,83%	20,67%	18,46%
XGBoost	S3	12,44%	18,97%	17,33%

Tabela 20: WAPE por subcategoria do canal Venda Direta.

As Tabelas 21 e 22 mostram as últimas análises para o comparativo dos métodos de GB, em que o WAPE das marcas mais vendidas em cada um dos canais é descrito. A Tabela 21 retrata os erros, para o canal Loja, e a Tabela 21, para o canal Venda Direta. No canal Loja (Figura 21), todos os métodos apresentaram o menor erro, em pelo menos um dos cenários, e o mesmo ocorre no canal Venda Direta (Figura 22).

Na Tabela 21, é possível notar que para a subcategoria mais vendida, o CatBoost obteve a melhor previsão de curto prazo e o LightGBM apresentou as melhores previsões de médio e longo prazo. Tanto para a subcategoria S2, quanto para a S3, o XGBoost fez as melhores previsões de curto e longo prazo, enquanto a melhor previsão de médio prazo foi a do CatBoost. Tanto o LightGBM, quanto o CatBoost, apresentaram erros acima de 20%, enquanto o maior erro apresentado pelo XGBoost foi de 13,77%. Pode-se concluir que o XGBoost, de maneira geral, possuiu uma melhor performance nesse cenário em relação aos demais métodos de GB.

Modelo	Marca	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	M1	11,97%	4,27%	8,51%
LightGBM	M1	21,78%	3,69%	7,45%
XGBoost	M1	13,77%	5,95%	8,23%
CatBoost	M2	23,92%	8,46%	17,79%
LightGBM	M2	23,84%	10,14%	20,07%
XGBoost	M2	8,42%	9,46%	9,75%
CatBoost	M3	13,68%	7,01%	9,84%
LightGBM	M3	24,69%	10,51%	15,85%
XGBoost	M3	0,81%	10,45%	1,66%

Tabela 21: WAPE por marca do canal Loja.

A Tabela 22 mostra que, para a marca mais vendida, a melhor previsão de curto prazo foi a apresentada pelo modelo LightGBM, enquanto as melhores previsões de médio e longo prazo foram apresentadas pelo modelo CatBoost. Para a segunda subcategoria mais vendida, o XGBoost apresentou o menor WAPE nos três horizontes temporais. Na segunda subcategoria mais vendida, o método XGBoost obteve o menor WAPE para as previsões de curto prazo e longo prazo, enquanto o LightGBM apresentou a melhor performance para a previsão de médio prazo. Os erros máximos de todos os três métodos para todas as subcategorias e horizontes foram entre 17% e 19%.

Em relação à estabilidade das previsões de cada subcategoria, ao longo dos horizontes temporais, nenhum método se sobressaiu em relação aos demais. Na subcategoria mais vendida, o método que apresentou maior estabilidade foi o LightGBM (0,55 pontos percentuais de diferença contra 5,08 do CatBoost e 5,2 do XGBoost). Na subcategoria S2, o XGBoost foi mais consistente que os demais métodos (XGBoost apresentou 4,67 pontos percentuais de diferença, CatBoost apresentou 7,05 pontos percentuais e o LightGBM apresentou 14,44 pontos percentuais). Para a terceira subcategoria mais vendida, o modelo mais estável foi o CatBoost (6,23 pontos percentuais de diferença contra 8,49 pontos do LightGBM e 14,56 do CatBoost). Nesse cenário em específico (análise por marcas mais vendidas no canal Venda Direta), não houve grande diferença entre a performance dos três modelos.

Modelo	Marca	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
CatBoost	M1	6,28%	1,20%	2,25%
LightGBM	M1	4,22%	4,77%	4,84%
XGBoost	M1	8,80%	3,60%	5,19%
CatBoost	M2	6,61%	8,79%	1,74%
LightGBM	M2	18,50%	16,13%	4,06%
XGBoost	M2	13,21%	9,51%	14,18%
CatBoost	M3	11,34%	17,57%	16,20%
LightGBM	M3	8,77%	17,26%	15,08%
XGBoost	M3	3,09%	17,65%	13,75%

Tabela 22: WAPE por marca do canal Venda Direta.

Por fim, a análise dos erros considerando as características de produtos, o XGBoost apresentou a melhor performance em 4 dos 6 cenários analisados, sendo eles: análise de categorias em ambos os canais, análise de subcategoria para o canal Loja e análise de marca também para o canal Loja. Como nos dois cenários restantes (análise de subcategoria e análise de marca no canal Venda Direta), um deles foi inconclusivo, é possível considerar que o método XGBoost apresenta uma solução mais estável.

5.1.5 Testes de Hipótese

Com base no WAPE calculado a partir dos dados de teste, foram realizados também testes de hipótese para garantir que há diferença significativa entre os resultados do CatBoost, LightGBM e XGBoost. O primeiro teste realizado foi o de Shapiro-Wilk para verificar se os dados possuem distribuição normal ou não. Após a execução do teste de normalidade, concluiu-se que todas as amostras analisadas (cada método / canal / horizonte) não apresentaram distribuição normal.

Como os testes de significância serão feitos a nível de horizonte e três métodos estão sendo estudados, torna-se necessário utilizar um teste em que duas ou mais amostras podem ser comparadas. Por esse motivo, o teste de Kruskal-Wallis foi selecionado alternativa para a realização dos três comparativos, com alpha igual a 0,05. Os p-valores encontrados foram $1,382364479382202 \times 10^{10}$ para o curto prazo, $2,418575625750961 \times 10^{23}$ para o médio prazo e $3,555983012984274 \times 10^{21}$ para o longo prazo. Diante desses resultados, pode-se inferir que a hipótese nula foi rejeitada e que há evidências de diferença significativa entre os grupos.

5.1.6 Impacto da Engenharia de Atributos

A etapa de *Feature Engineering* (FE - do inglês, Engenharia de Atributos) foi descrita na Seção 4.5. Nesta seção, o impacto da etapa de Engenharia de Atributos será

analisado. Primeiramente, serão visualizados os erros gerais por canal e por horizonte com FE, contra o WAPE dos modelos sem FE. Em sequência, serão analisadas as 10 variáveis mais importantes para cada modelo, canal e horizonte.

A Tabela 23 mostra um comparativo dos resultados para a previsão de curto prazo com modelos *baseline* (base de referência) e modelos com FE. É possível observar que, para o canal Venda Direta, todos os modelos com as variáveis criadas performaram melhor. Porém, para o canal de Lojas, o mesmo não aconteceu. Isso pode ser um indicativo de que pode ser necessário criar variáveis que extraíam características desse canal em específico, para que a previsão de curto prazo tenha um melhor resultado.

Modelo	Canal	Com FE	Sem FE
CatBoost	Lojas	18,27%	8,07%
LightGBM	Lojas	22,77%	2,27%
XGBoost	Lojas	10,00%	2,86%
CatBoost	Venda Direta	0,83%	21,94%
LightGBM	Venda Direta	1,98%	4,44%
XGBoost	Venda Direta	0,86%	8,85%

Tabela 23: WAPE por canal para previsão do próximo ciclo: Com FE vs Sem FE.

Na Tabela 24, o WAPE dos modelos *baseline* com os modelos com a etapa de FE foi comparado. Nesse caso, nota-se que todos os modelos que tiveram a etapa de Engenharia de Atributos performaram melhor que os modelos *baseline*. Isso ressalta a maior importância dessa etapa para previsões com saltos temporais.

Modelo	Canal	Com FE	Sem FE
CatBoost	Lojas	1,42%	2,70%
LightGBM	Lojas	2,82%	46,86%
XGBoost	Lojas	2,12%	28,18%
CatBoost	Venda Direta	10,42%	14,50%
LightGBM	Venda Direta	9,78%	18,43%
XGBoost	Venda Direta	7,31%	14,28%

Tabela 24: WAPE por canal para previsão de 5 ciclos à frente: Com FE vs Sem FE.

Os erros para a previsão a longo prazo com FE e sem FE podem ser visualizados na Tabela 25. Todos os modelos com Engenharia de Atributos performaram melhor que os modelos *baseline*, com exceção do modelo CatBoost, no canal Venda Direta. Porém, a diferença é de apenas 0,03 pontos percentuais.

Modelo	Canal	Com FE	Sem FE
CatBoost	Lojas	12,23%	17,65%
LightGBM	Lojas	13,67%	20,35%
XGBoost	Lojas	6,40%	6,94%
CatBoost	Venda Direta	5,19%	5,16%
LightGBM	Venda Direta	5,85%	23,90%
XGBoost	Venda Direta	5,41%	23,77%

Tabela 25: WAPE por canal para previsão de 10 ciclos à frente: Com FE vs Sem FE.

Para prosseguir a análise do impacto dos atributos em cada um dos métodos GB empregados no estudo, é importante visualizar quais são as 10 variáveis mais importantes para cada método, canal e horizonte. Devido à confidencialidade dos dados, os nomes dos atributos mais importantes não serão identificados, mas serão divididos em grupos. Os grupos foram nomeados conforme lista abaixo para facilitar o entendimento:

- G-O: Atributo original da base de dados;
- G-Pro: Atributo construídos considerando produtos, promoções e campanhas (Tabela 5);
- G-Share: Atributo construídos calculando o *market share* (Tabela 6);
- G-Lag: Atributo de *lag* criados (Tabela 7);
- G-Ciclo: Atributo construídos a partir de informações dos ciclos (Tabela 9);
- G-Auto: Atributos automáticos criados (Tabela 11).

As Tabelas 26, 27 e 28 mostram os dez atributos mais importantes em ambos os canais para o CatBoost, LightGBM e XGBoost, respectivamente.

Para a previsão de curto prazo do CatBoost no canal Loja (Tabela 26), a segunda e a penúltima variável no *ranking* são originais da base de dados. Das demais variáveis, 7 são variáveis de lag e 1 é referente às variáveis de características dos ciclos. Para o canal Venda Direta, as dez variáveis mais importantes foram criadas na etapa de Engenharia de Atributos. As 8 variáveis mais importantes são de *lag*, conforme foi descrito anteriormente na Tabela 7 da Seção 4.5. A nona variável mais importante foi da categoria de produtos, promoções e campanhas, e a última variável do *ranking* foi uma automática.

Na previsão de médio prazo utilizando o CatBoost, a Tabela 26 mostrou que 9 das 10 variáveis mais importantes para tanto para o modelo de Venda Direta quanto para o modelo de Loja foram criadas na metodologia do presente estudo. Em Venda Direta, 3 variáveis são referentes ao grupo de produtos, campanhas e promoções, e 6 variáveis são

de *lag*. Para o canal Loja, as variáveis de *lag* também foram a maioria (totalizando 5), 3 variáveis foram oriundas das características de ciclo e a variável restante foi originária das características de produtos, promoções e campanhas.

Na Tabela 26, também é possível visualizar os 10 atributos mais importantes para o CatBoost no horizonte de 10 ciclos à frente. No canal Venda Direta, todas as variáveis mais importantes foram construídas dentro da metodologia e, no canal Loja, 9 das 10 variáveis mais importantes foram criadas. No canal Venda Direta, 7 variáveis são de *lag* e 3 variáveis são originárias de características de produtos, promoções e campanhas. No canal Loja, 4 variáveis são de *lag*, uma variável é de produtos, campanhas e promoções e 4 variáveis foram criadas a partir de características de ciclo.

Ordem de Importância	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
Canal Loja			
1	G-Lag	G-Ciclo	G-Lag
2	G-O	G-Lag	G-Ciclo
3	G-Lag	G-Lag	G-Lag
4	G-Ciclo	G-O	G-Ciclo
5	G-Lag	G-Lag	G-Pro
6	G-Lag	G-Ciclo	G-Lag
7	G-Lag	G-Lag	G-Lag
8	G-Lag	G-Lag	G-Ciclo
9	G-O	G-Ciclo	G-Ciclo
10	G-Lag	G-Pro	G-O
Canal Venda Direta			
1	G-Lag	G-Pro	G-Lag
2	G-Lag	G-Lag	G-Pro
3	G-Lag	G-Pro	G-Pro
4	G-Lag	G-Pro	G-Lag
5	G-Lag	G-Lag	G-Pro
6	G-Lag	G-Lag	G-Lag
7	G-Lag	G-Lag	G-Lag
8	G-Lag	G-Lag	G-Lag
9	G-Pro	G-O	G-Lag
10	G-Auto	G-Lag	G-Lag

Tabela 26: Atributos mais importantes para o modelo CatBoost.

Para o modelo LightGBM (Tabela 27), 8 entre as 10 variáveis mais importantes foram criadas conforme o que foi mostrado na Seção 4.5 tanto para o canal Venda Direta quanto para o canal Loja na previsão do próximo ciclo. Para o canal Venda Direta, 5 das variáveis mais importantes são de *lag* e as 3 restantes são do grupo de produtos, campanhas e promoções. No canal Loja, a maior parte (5) das variáveis mais importantes também foram de *lag*, uma foi do grupo de produtos, campanhas e promoções, um único atributo foi criado a partir das características de ciclo e o último atributo foi referente a

uma variável de *market share*.

Na previsão de 5 ciclos à frente utilizando o LightGBM (Tabela 27), no modelo de Venda Direta somente uma variável dentre as mais importantes é original da base de dados e, para o canal Loja, apenas duas. No canal Venda Direta, 3 das variáveis mais importantes são oriundas de características de produtos, promoções e campanhas, 5 são variáveis de lag e uma é do grupo das variáveis de *market share*. Para o canal Loja, as variáveis de *lag* também foram maioria (totalizando em 4 dentre as mais importantes), duas são do grupo de produtos, campanhas e promoções e as outras duas são originárias das características de ciclo.

No modelo do horizonte de longo prazo, 8 das 10 variáveis mais importantes do LightGBM foram construídas, em ambos os canais (Tabela 27). No canal Loja, 5 variáveis são de *lag* e 3 variáveis são de produtos, campanhas e promoções. Para Venda Direta, 4 das variáveis mais importantes para o modelo de Venda Direta são de *lag*, 3 são de produtos e promoções e uma é de *market share*.

Ordem de Importância	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
Canal Loja			
1	G-Lag	G-O	G-Lag
2	G-O	G-Lag	G-O
3	G-O	G-Lag	G-O
4	G-Ciclo	G-O	G-Pro
5	G-Lag	G-Ciclo	G-Lag
6	G-Lag	G-Pro	G-Pro
7	G-Pro	G-Ciclo	G-Pro
8	G-Lag	G-Pro	G-Lag
9	G-Lag	G-Lag	G-Lag
10	G-Share	G-Lag	G-Lag
Canal Venda Direta			
1	G-O	G-Pro	G-O
2	G-Lag	G-O	G-Lag
3	G-Pro	G-Pro	G-Pro
4	G-O	G-Pro	G-Pro
5	G-Pro	G-Lag	G-Pro
6	G-Lag	G-Lag	G-Lag
7	G-Pro	G-Lag	G-Lag
8	G-Lag	G-Lag	G-O
9	G-Lag	G-Lag	G-Share
10	G-Lag	G-Share	G-Lag

Tabela 27: Atributos mais importantes para o modelo LightGBM.

A Tabela 28 mostra os 10 atributos mais importantes para ambos os canais e todos os horizontes para o modelo XGBoost.

Ordem de Importância	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
Canal Loja			
1	G-Lag	G-Ciclo	G-Lag
2	G-O	G-Lag	G-Ciclo
3	G-Lag	G-Lag	G-Lag
4	G-Auto	G-Lag	G-Pro
5	G-Lag	G-O	G-Lag
6	G-Lag	G-Ciclo	G-Ciclo
7	G-Auto	G-Pro	G-Lag
8	G-Pro	G-O	G-Pro
9	G-Share	G-Lag	G-Lag
10	G-Auto	G-Ciclo	G-O
Canal Venda Direta			
1	G-Lag	G-Lag	G-Lag
2	G-Lag	G-Lag	G-Lag
3	G-Pro	G-O	G-Pro
4	G-Auto	G-Lag	G-Auto
5	G-Pro	G-O	G-Pro
6	G-Lag	G-Lag	G-Auto
7	G-Auto	G-Pro	G-Lag
8	G-Pro	G-Lag	G-Pro
9	G-Lag	G-Pro	G-Lag
10	G-Pro	G-Auto	G-Pro

Tabela 28: Atributos mais importantes para o modelo XGBoost.

Para a previsão do próximo ciclo no canal Venda Direta, todas as 10 variáveis mais importantes foram criadas dentro da metodologia da pesquisa e, para o canal Loja, apenas uma variável é original da base de dados. No canal Venda Direta, 4 variáveis do *ranking* são de *lag*, 4 atributos foram criados extraindo características de produtos, campanhas e promoções e 2 variáveis estão no grupo de variáveis automáticas. No canal Loja, 4 variáveis são de *lag*, 3 variáveis são automáticas, 1 variável é da categoria de produtos, campanhas e promoções e o atributo restante é de *market share*.

Para a previsão de médio prazo do XGBoost (Tabela 28), 8 dos 10 atributos mais importantes em ambos os canais foram construídos, de acordo com o que foi mencionado na Seção 4.5. Para Venda Direta, 5 das 8 variáveis criadas dentre as mais importantes são *lags*, 2 são do grupo de produtos, promoções e campanhas (Tabela 5) e uma é automática. Para o canal de Loja, 4 variáveis são de *lag*, uma variável é de produtos, campanhas e promoções e as outras 3 variáveis são oriundas das características de ciclo.

Por último, na previsão de 10 ciclos à frente utilizando o método XGBoost, todas os 10 atributos mais importantes do canal Venda Direta foram criados e, para o canal Loja, somente um atributo é original da base de dados. No canal Venda Direta, 4 atributos são de *lag* (Tabela 7), 4 variáveis são de produtos, campanhas e promoções (Tabela 5) e

dois são atributos automáticos. No canal Loja, 5 das variáveis criadas são de *lag*, 2 são de produtos e campanhas e 2 são oriundas de características dos ciclos (Tabela 9).

Ao comparar os atributos mais importantes nos diferentes métodos para o mesmo canal e o mesmo horizonte, alguns deles se repetem. A Tabela 29 ilustra quantos atributos se repetem entre os três métodos na visão de mesmo canal e mesmo horizonte.

Canal	Próximo Ciclo	5 ciclos à frente	10 ciclos à frente
Lojas	4	5	5
Venda Direta	2	5	6

Tabela 29: Quantidade de atributos com interseção entre os três métodos para cada canal e horizonte.

Dentre os atributos que se repetem para a previsão de curto prazo, 5 deles são variáveis de *lag* e o outro atributo foi construído considerando informações de produtos, campanhas e promoções. Para o médio prazo, 6 atributos são de *lag*, 2 são atributos de produtos, campanhas e promoções e 2 foram atributos criados considerando informações de ciclos. No longo prazo, novamente, 6 atributos contados são *lags*, 4 são atributos construídos a partir de informações de produtos, promoções e campanhas e um único atributo é original da base de dados. Tendo isso em vista, é possível notar que os atributos que causaram mais impacto entre todos os métodos e todos os horizontes foram as *lags*, e, em seguida, os atributos relacionados aos produtos e suas promoções. Considerando que as *lags* são importantes para tornar os métodos de regressão conscientes de tempo e que promoções e características dos produtos vendidos são alavancas de vendas, esses resultados são compatíveis tanto com a natureza do negócio quanto com a natureza dos métodos GB utilizados no estudo.

Em resumo, ao comparar os erros nos modelos com *feature engineering* e *baseline* nas Tabelas 23, 24 e 25, foi observado que não foram em todos os cenários que os modelos de *baseline* tiveram o menor WAPE, apesar de ter acontecido para ambos os canais, nos horizontes de médio e longo prazo e para o canal de Venda Direta, no horizonte de curto prazo. Porém, vale relembrar que a consistência dos modelos ao longo dos horizontes é muito importante nas análises e os atributos criados contribuíram para a estabilidade deles. Ao analisar o *ranking* dos atributos mais importantes para todos os modelos, canais e horizontes, pode-se observar que pelo menos 80% dos atributos em todos os modelos fazem parte das variáveis que foram criadas. Isso mostra que os atributos construídos na etapa de Engenharia de Atributos (Seção 4.5) da Metodologia tiveram grande contribuição para os resultados de todos os modelos de *Gradient Boosting*.

5.1.7 Impacto da Otimização de Hiperparâmetros

Conforme foi mencionado na Seção 4.6, o único método de *Gradient Boosting* que passou pela etapa de Otimização de Hiperparâmetros foi o XGBoost, que teve uma performance melhor que o CatBoost e o LightGBM.

O XGBoost foi reconhecido como o método mais eficaz de *Gradient Boosting*, conforme discutido anteriormente neste capítulo. No entanto, é viável resumir algumas das análises para estabelecer essa conclusão de forma mais concisa. Pela Tabela 14, que mostra o WAPE para os dois canais de venda nos três horizontes temporais, por exemplo, foi identificado que o XGBoost demonstrou uma maior estabilidade nos múltiplos horizontes temporais que o CatBoost e o LightGBM. Na Seção 5.1.2, em que os erros foram analisados em 6 ciclos de predição para cada canal e horizonte temporal, o XGBoost apresentou erros consideravelmente menores que os demais modelos, sendo que os erros máximos do CatBoost e do LightGBM foram praticamente o dobro do erro máximo apresentado pelo XGBoost.

A Tabela 30 mostra o comparativo dos erros dos modelos em múltiplos horizontes temporais, em ambos os canais, antes e depois dos hiperparâmetros otimizados.

Horizonte	Canal	Com otimização	Sem otimização
Próximo ciclo	Lojas	8,24%	10,00%
Próximo ciclo	Venda Direta	11,40%	0,86%
5 ciclos à frente	Lojas	6,17%	2,12%
5 ciclos à frente	Venda Direta	19,45%	7,31%
10 ciclos à frente	Lojas	1,83%	6,40%
10 ciclos à frente	Venda Direta	15,32%	5,41%

Tabela 30: WAPE do XGBoost por horizonte e canal: Com e sem otimização de hiperparâmetros.

Com base na Tabela 30, é possível notar que a etapa de otimização de hiperparâmetros não garante que seu impacto seja sempre positivo. Dos 6 cenários comparados na tabela, somente dois deles tiveram melhores resultados com os hiperparâmetros otimizados: a previsão para o próximo ciclo do canal Loja e a previsão de 10 ciclos à frente também para o canal Loja. Tendo isso em vista, os modelos que serão comparados posteriormente com o Prophet serão aqueles que apresentaram o menor erro, independente de terem sido treinados antes ou depois da etapa de otimização de hiperparâmetros.

5.2 Análise dos Componentes de Séries Temporais

Conforme foi citado na Seção 2.1, as séries temporais possuem quatro componentes. Como a Seção 5.3 trata do método de séries temporais utilizado neste estudo, o Prophet, é

importante analisar os componentes das séries temporais para entender o comportamento dos dados. Nesta Seção, serão analisados os componentes: tendência, sazonalidade e resíduos.

Como são 1556 séries temporais no total, se torna inviável fazer a análise de cada uma separadamente. Sendo assim, a série temporal completa foi investigada. Além disso, por questão de confidencialidade dos dados, o volume de vendas foi normalizado entre os valores de 0 e 1. A Figura 12 mostra a série temporal decomposta.

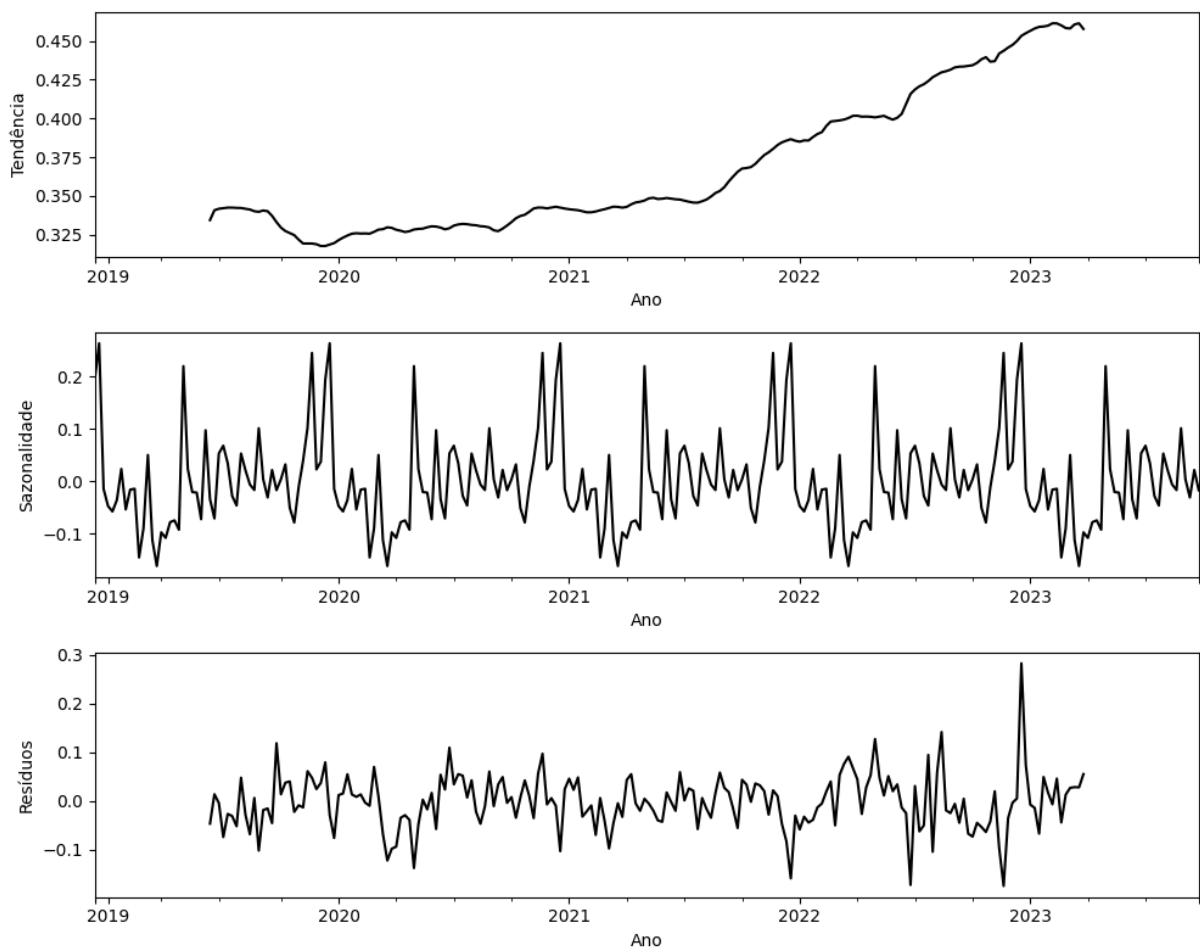


Figura 12: Decomposição aditiva da série temporal em relação à tendência, sazonalidade e resíduos entre o segundo semestre de 2019 e o primeiro semestre de 2023.

Ao avaliar a tendência na Figura 12, é possível notar que ela é ascendente, ou seja, o volume total de vendas está aumentando ao longo dos anos. Em relação à sazonalidade, percebe-se que há picos de vendas em determinados momentos dos anos e esses picos se repetem. No varejo, datas especiais como Natal, Dia das Mães e Dia dos Pais, por exemplo, são também datas comerciais. Essas datas movimentam o mercado de consumo e isso pode ser visualizado através do segundo gráfico da Figura 12. No terceiro gráfico da Figura 12, pode-se perceber que o modelo aditivo é aderente à série em questão, dado

que os resíduos podem ser considerados estacionários visualmente, com exceção do pico apresentado no início de 2023.

5.3 Análise do método de Séries Temporais: Prophet

Nesta seção, o impacto de duas etapas será avaliado: Engenharia de Atributos (Seção 5.3.1) e Otimização de Hiperparâmetros (Seção 5.3.2). A visão dos erros em diferentes cenários será exposta na Seção 5.4, em que os resultados do Prophet e do XGBoost serão comparados.

5.3.1 Impacto da Engenharia de Atributos

Para entender o impacto da etapa de Engenharia de Atributos, a Tabela 31 mostra o WAPE para cada horizonte temporal e canal, com os atributos que foram criados e sem esses atributos. Diferente da análise de Impacto de Engenharia de Atributos que foi feita na Seção 5.1.6 para os métodos GB, nesta seção não será avaliada a importância dos atributos através de gráficos, visto que cada série temporal é um modelo e são mais de 1500 séries temporais presentes neste estudo.

Horizonte	Canal	Com FE	Sem FE
Curto Prazo	Lojas	1,81%	49,45%
Médio Prazo	Lojas	6,34%	25,33%
Longo Prazo	Lojas	4,05%	20,89%
Curto Prazo	Venda Direta	2,17%	31,96%
Médio Prazo	Venda Direta	8,94%	5,85%
Longo Prazo	Venda Direta	4,90%	32,00%

Tabela 31: WAPE por canal para previsão em múltiplos horizontes: Com FE vs Sem FE.

A Tabela 31 mostra que todos os modelos com os atributos criados possuíam melhor performance, com exceção do horizonte de médio prazo para o canal Venda Direta. Ainda assim, a diferença é de somente 3,09 pontos percentuais, o que ainda justifica o impacto positivo da etapa de Engenharia de Atributos.

5.3.2 Impacto da Otimização de Hiperparâmetros

Para avaliar o impacto da etapa de Otimização de Hiperparâmetros, a Tabela 32 mostra o WAPE para cada horizonte temporal e canal, antes e depois da etapa da Otimização de Hiperparâmetros. Assim como foi observado na Seção 5.1.7, a etapa de Otimização de Hiperparâmetros não garante que os resultados serão melhores. Para o

Prophet, os resultados foram melhores apenas em dois horizontes para o canal Venda Direta: médio e longo prazo. Serão comparados com o XGBoost os resultados em que o WAPE foi menor.

Horizonte	Canal	Sem otimização	Com otimização
Curto Prazo	Lojas	1,81%	55,81%
Médio Prazo	Lojas	6,34%	200,75%
Longo Prazo	Lojas	4,05%	171,5%
Curto Prazo	Venda Direta	2,17%	2,96%
Médio Prazo	Venda Direta	8,94%	1,95%
Longo Prazo	Venda Direta	4,90%	0,74%

Tabela 32: WAPE por canal para previsão em múltiplos horizontes: Com FE vs Sem FE.

5.4 Comparativo entre XGBoost e Prophet

Nesta seção, serão analisados os resultados do XGBoost e do Prophet. Os resultados utilizados para comparação estão de acordo com os menores erros encontrados nas análises das Seções 5.1.7 e 5.3.2, seja com os hiperparâmetros padrão (antes da otimização), ou com os hiperparâmetros otimizados. Sendo assim, os seguintes erros serão analisados:

- Curto prazo no canal Loja: XGBoost com hiperparâmetros, após otimização e Prophet, com hiperparâmetros predefinidos;
- Curto prazo no canal Venda Direta: Ambos os modelos com hiperparâmetros predefinidos;
- Médio prazo no canal Loja: Ambos os modelos com hiperparâmetros predefinidos;
- Médio prazo no canal Venda Direta: XGBoost, com hiperparâmetros predefinidos e Prophet, com hiperparâmetros otimizados;
- Longo prazo no canal Loja: XGBoost, com hiperparâmetros após otimização e Prophet, com hiperparâmetros padronizados;
- Longo prazo no canal Venda Direta: XGBoost, com hiperparâmetros predefinidos e Prophet, com hiperparâmetros otimizados;

A Tabela 33 expõe o WAPE nos 3 horizontes temporais para cada canal de venda. No canal Loja, o Prophet fez a melhor predição no curto prazo e o XGBoost apresentou as melhores predições para os horizontes com salto temporal. No canal Venda Direta,

o inverso aconteceu. O Prophet apresentou mais estabilidade na predição dos múltiplos horizontes que o XGBoost em ambos os canais.

Modelo	Canal	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	Lojas	8,24%	2,12%	1,83%
Prophet	Lojas	1,80%	6,34%	4,04%
XGBoost	Venda Direta	0,86%	7,31%	5,41%
Prophet	Venda Direta	2,17%	1,95%	0,74%

Tabela 33: Comparativo do WAPE por canal e horizonte.

Na Tabela 34 e na Tabela 35, é feito um comparativo dos erros a nível de regional para os canais Loja e Venda Direta, respectivamente. R1 se refere à regional com o maior volume de vendas e R6 à regional com o menor volume de vendas no país. O Prophet apresentou o menor erro no horizonte de curto prazo para todas as regionais no canal Loja (Tabela 34), enquanto o XGBoost teve a melhor performance em todas as regionais no mesmo canal para a previsão de 5 ciclos à frente / 15 semanas à frente. No horizonte de longo prazo, o Prophet apresentou o melhor resultado em praticamente todas as regionais, com exceção de R4. O maior erro observado para o XGBoost é de 25,25% (previsão para R6 no curto prazo), o que é considerado um WAPE elevado. O maior erro observado para o Prophet é de 12,10% (previsão para R4 no longo prazo), mais da metade em pontos percentuais. Sendo assim, mais uma vez, o Prophet se mostrou um modelo mais estável que o XGBoost.

Modelo	Regional	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	R1	4,88%	5,20%	7,09%
Prophet	R1	0,66%	7,58%	5,57%
XGBoost	R2	4,23%	0,49%	4,06%
Prophet	R2	4,14%	4,21%	1,35%
XGBoost	R3	8,40%	3,13%	2,43%
Prophet	R3	0,80%	6,29%	0,59%
XGBoost	R4	9,16%	1,40%	1,65%
Prophet	R4	3,49%	5,45%	12,10%
XGBoost	R5	12,78%	2,48%	7,90%
Prophet	R5	2,33%	5,94%	3,05%
XGBoost	R6	25,25%	1,19%	14,98%
Prophet	R6	2,31%	9,29%	3,13%

Tabela 34: Comparativo do WAPE por horizonte e regional no canal Loja.

No canal Venda Direta (35), os menores erros ficaram mais distribuídos. Para R1, o XGBoost obteve a melhor performance no horizonte de curto prazo e para a previsão de 10 ciclos à frente / 30 semanas à frente. Para R2, o Prophet apresentou os menores erros em todos os horizontes temporais. Para R3, o XGBoost foi o modelo que apresentou o

melhor resultado nos três horizontes de tempo. Para R4, a melhor previsão de curto prazo foi do modelo XGBoost e, nas demais previsões para essa regional, o Prophet obteve o melhor resultado. Para R5, acontece o inverso do que aconteceu para a regional R1. Para a regional R6, o Prophet performou melhor no curto e médio prazo, enquanto o XGBoost performou melhor no longo prazo. Os resultados da Tabela 35, assim, não são muito conclusivos para definir qual método teve resultado mais satisfatório.

Modelo	Regional	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	R1	0,32%	5,55%	4,79%
Prophet	R1	3,33%	4,42%	4,83%
XGBoost	R2	8,72%	14,92%	13,76%
Prophet	R2	0,32%	0,87%	3,34%
XGBoost	R3	2,49%	4,15%	0,11%
Prophet	R3	5,35%	4,98%	3,59%
XGBoost	R4	1,31%	6,63%	0,65%
Prophet	R4	7,47%	0,80%	0,19%
XGBoost	R5	6,22%	0,02%	3,13%
Prophet	R5	1,71%	3,89%	1,44%
XGBoost	R6	5,19%	2,16%	0,30%
Prophet	R6	0,20%	1,12%	1,08%

Tabela 35: Comparativo do WAPE por horizonte e regional no canal Venda Direta.

Em sequência, os erros para as 3 categorias com maior volume de vendas serão analisados, sendo C1 referente à categoria mais vendida, C2 à segunda categoria com mais vendas e C3 à terceira categoria com maior volume. A Tabela 36 mostra os resultados por categoria para o canal Loja e a Tabela 37 para o canal Venda Direta. No canal Loja (Tabela 36), o Prophet obteve a melhor performance para todos os horizontes temporais para a categoria C2. Para a categoria C1, o Prophet apresentou os menores erros para as previsões de curto e médio prazo, enquanto o XGBoost performou melhor no longo prazo. Para a categoria C3, novamente, o Prophet obteve a melhor performance em dois horizontes temporais (curto e longo prazo) e o XGBoost em um horizonte (médio prazo). Com base na Tabela 36, o Prophet exibiu os resultados melhores.

Modelo	Categoria	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	C1	3,03%	7,23%	6,15%
Prophet	C1	3,02%	6,14%	9,36%
XGBoost	C2	12,66%	4,47%	3,55%
Prophet	C2	5,84%	3,12%	2,19%
XGBoost	C3	12,90%	3,60%	7,31%
Prophet	C3	2,75%	4,20%	6,81%

Tabela 36: Comparativo do WAPE por horizonte e categoria no canal Loja.

Na Tabela 37, observa-se que o Prophet obteve a melhor performance para as categorias C1 e C2, em todos os horizontes temporais estudados. Para C3, o Prophet apresentou os melhores resultados para as predições de curto e médio prazo, enquanto o XGBoost apresentou o menor erro para a predição de 10 ciclos à frente / 30 semanas à frente. Mais uma vez, pode-se dizer que o Prophet apresentou resultados mais satisfatórios que os do XGBoost.

Modelo	Categoria	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	C1	8,97%	12,13%	13,34%
Prophet	C1	1,69%	0,92%	0,84%
XGBoost	C2	4,55%	7,31%	4,02%
Prophet	C2	2,38%	0,64%	3,74%
XGBoost	C3	23,56%	6,90%	1,72%
Prophet	C3	16,88%	2,93%	11,32%

Tabela 37: Comparativo do WAPE por horizonte e categoria no canal Venda Direta.

Prosseguindo com as análises utilizando características dos produtos, as Tabelas 38 e 39 indicam os erros para os canais Loja e Venda Direta, respectivamente, ao estudar o nível de subcategoria. Os erros foram calculados para as 3 subcategorias mais vendidas. Para S1 no canal Loja (Tabela 38), o XGBoost apresentou os menores erros para as predições de curto e médio prazo e o inverso acontece para S3. Para S2, o Prophet obteve os menores erros para as predições de curto e longo prazo, enquanto o XGBoost teve o menor WAPE no horizonte de médio prazo. A Tabela 39 mostra que o Prophet apresentou os menores erros em todas as subcategorias e todos os horizontes temporais. Na análise por subcategoria, o Prophet também obteve resultados mais satisfatórios.

Modelo	Subcategoria	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	S1	0,04%	7,75%	9,19%
Prophet	S1	0,79%	8,64%	7,89%
XGBoost	S2	7,66%	0,33%	8,22%
Prophet	S2	7,25%	0,47%	0,70%
XGBoost	S3	8,11%	5,80%	0,02%
Prophet	S3	6,95%	2,11%	11,71%

Tabela 38: Comparativo do WAPE por horizonte e subcategoria no canal Loja.

Modelo	Subcategoria	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	S1	5,55%	6,59%	9,85%
Prophet	S1	5,19%	4,11%	1,02%
XGBoost	S2	6,67%	8,65%	2,65%
Prophet	S2	1,00%	5,70%	0,84%
XGBoost	S3	12,44%	18,97%	17,33%
Prophet	S3	2,86%	10,01%	5,28%

Tabela 39: Comparativo do WAPE por horizonte e subcategoria no canal Venda Direta.

Na visão por marca, as Tabelas 40 e 41 mostram os comparativos para os canais Loja e Venda Direta, respectivamente. As 3 marcas com maior volume de vendas foram consideradas e, mais uma vez, foram nomeadas de acordo com o maior volume (M1), segundo maior volume (M2) e terceiro maior volume (M3). Para M1 e M3 no canal Loja (Tabela 40), o WAPE do Prophet foi melhor nas previsões de curto e médio prazo. Para M3, o XGBoost apresentou o menor WAPE para as previsões com salto temporal. No canal Venda Direta (Tabela 41), o menor WAPE calculado para as marcas M2 e M3 em todos os três horizontes temporais foi o do Prophet. Para M1, o XGBoost apresentou o menor WAPE para os horizontes de médio e longo prazo.

Modelo	Marca	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	M1	7,55%	5,95%	11,63%
Prophet	M1	1,60%	2,84%	11,73%
XGBoost	M2	14,39%	9,46%	1,62%
Prophet	M2	3,22%	10,14%	5,03%
XGBoost	M3	14,75%	10,45%	0,29%
Prophet	M3	2,98%	2,71%	7,46%

Tabela 40: Comparativo do WAPE por horizonte e marca no canal Loja.

Modelo	Marca	Curto Prazo	Médio Prazo	Longo Prazo
XGBoost	M1	8,80%	3,60%	5,19%
Prophet	M1	5,28%	13,21%	19,72%
XGBoost	M2	13,21%	9,51%	14,18%
Prophet	M2	8,88%	1,02%	2,70%
XGBoost	M3	3,09%	17,65%	13,75%
Prophet	M3	0,62%	7,65%	0,37%

Tabela 41: Comparativo do WAPE por horizonte e marca no canal Venda Direta.

Após analisar as Tabelas 33, 34, 35, 36, 37, 38, 39, 40 e 41 presentes nesta Seção, conclui-se que o Prophet foi capaz de fazer melhores previsões que o XGBoost, visto que diversos cenários distintos foram estudados.

Em síntese, tal conclusão é alcançada por diversas razões e pode-se salientar duas delas. Primeiramente, a Tabela 33 com os erros gerais por canal de venda mostra que o Prophet apresentou maior estabilidade que o XGBoost nos horizontes temporais distintos. Em segundo lugar, as Tabelas 36 e 37 que apresentam o WAPE das 3 categorias mais vendidas nos canais Loja e Venda Direta demonstram que o Prophet apresentou o menor erro em pelo menos dois horizontes temporais de todas as 3 categorias em ambos os canais.

Para complementar a análise e garantir que há evidência de diferença significativa entre os resultados, foram conduzidos testes de hipótese, conforme foi feito anteriormente na Seção 5.1.5 no comparativo entre métodos de *Gradient Boosting*. O WAPE foi calculado no nível de granularidade canal / regional / categoria / subcategoria / marca / horizonte e os testes de hipótese foram realizados sobre os horizontes temporais.

Ao realizar o teste de normalidade de Shapiro-Wilk tanto para as amostras de WAPE do XGBoost quanto para as amostras do Prophet, os p-valores indicaram que a hipótese nula foi rejeitada e que os dados não apresentam distribuição normal. Com isso, é necessário aplicar um teste não-paramétrico nos dados para verificar se há diferença significativa entre os grupos. Mais uma vez, o teste de Kruskal-Wallis foi aplicado.

Após a execução dos testes de hipótese, os p-valores encontrados foram: $1,511106123734102 \times 10^{90}$ para a previsão de curto prazo, $7,649976821309141 \times 10^{79}$ para a previsão de médio prazo e $4,765499844580534 \times 10^{61}$. Sendo assim, com alpha igual a 0,05, os p-valores são menores que alpha e a hipótese nula é rejeitada. Isso significa que há evidências de diferença significativa entre os resultados do XGBoost e do Prophet.

O Capítulo 6 resume as considerações finais deste trabalho, após toda a análise e discussão dos resultados.

6 CONSIDERAÇÕES FINAIS

A previsão de demanda é fundamental em vários mercados e indústrias, em diferentes setores, e não é uma tarefa trivial. Para empresas que estão alocadas tanto no segmento da indústria, quanto do varejo, por exemplo, é primordial saber quando, quais e quantos produtos serão produzidos para que estes estejam disponíveis para compra para os clientes no momento desejado e de acordo com as suas demais necessidades. A previsão da demanda em um cenário de longo prazo é ainda mais desafiadora e é comum para empresas que estão em ambos os segmentos (ZHOU et al., 2022).

Este estudo teve como objetivo principal propor uma solução para previsão de demanda para diferentes categorias, subcategorias, marcas e canais de uma empresa em múltiplos horizontes temporais, identificando padrões de consumo e entendendo qual dos modelos comparados possui uma performance mais estável nesses horizontes. Foram utilizados dados reais de uma grande empresa brasileira do mercado de CFT, intitulada de maneira sigilosa como Empresa X. As previsões foram feitas em três horizontes de tempo, sendo estes: curto prazo (próximo período), médio prazo (aproximadamente três meses à frente) e longo prazo (cerca de sete meses à frente). Os algoritmos CatBoost, LightGBM, XGBoost e Prophet foram estudados.

Na preparação de dados para a modelagem, foram construídos atributos considerando características dos produtos, promoções, preços, campanhas e informações temporais, descritos na Seção 4.5. As Seções 5.1.6 e 5.3.1 mostraram o impacto da etapa de Engenharia de Atributos para todos os modelos.

A Seção 5.1 teve como objetivo comparar exclusivamente os métodos de *Gradient Boosting* estudados (CatBoost, LightGBM e XGBoost) e identificar qual dos três métodos indicou maior estabilidade ao prever múltiplos horizontes. O XGBoost apresentou os menores erros para o canal Loja na previsão em todos os três horizontes (10% para o curto prazo, 2,12% para o médio prazo e 6,4% para o longo prazo). Para o canal Venda Direta, o XGBoost não apresentou o menor WAPE em todos os horizontes, mas teve mais estabilidade em comparação ao CatBoost e ao LightGBM.

Após o comparativo entre métodos GB, o XGBoost foi comparado com um método de séries temporais, o Prophet. Por ser um método dessa natureza, os componentes da série temporal foram explorados na Seção 5.2. Tanto o XGBoost quanto o Prophet passaram pela etapa de Otimização de Hiperparâmetros e o impacto dessa etapa foi analisado nas Seções 5.1.7 e 5.3.2.

A Seção 5.4 apresentou os resultados comparativos entre o XGBoost e o Prophet. Ao comparar a performance do XGBoost e do Prophet em cenários distintos, tanto do ponto de vista de localização (regional), quanto do ponto de vista de características dos

produtos (categoria, subcategoria e marca), concluiu-se que o Prophet apresentou resultados mais satisfatórios e maior estabilidade na previsão de múltiplos horizontes. Porém, o XGBoost também apresentou uma performance consistente.

A principal limitação deste estudo reside no fato de que as análises conduzidas se restringiram à avaliação exclusiva da performance dos métodos em diversos cenários, utilizando a métrica WAPE. Portanto, outras avaliações se fazem necessárias para determinar qual modelo seria mais apropriado para implementação prática. Aspectos cruciais para a compreensão da viabilidade das opções incluem a análise do tempo de execução e do custo computacional.

Para trabalhos futuros, os aprimoramentos podem ser derivados:

- da implementação da metodologia apresentada com dados de outras empresas;
- do refinamento da etapa de otimização de hiperparâmetros, caso tenha mais disponibilidade de poder computacional, realizando mais tentativas de busca, por exemplo;
- da condução de experimentos por meio da formação de um *ensemble* entre os modelos;
- da realização de experimentos com outros algoritmos, como o ARIMA (do inglês, *Autoregressive integrated moving average* - Modelo Autorregressivo Integrado de Médias Móveis) e o LSTM (do inglês, *Long Short-Term Memory*).

REFERÊNCIAS

- ABIHPEC. **A Indústria de Higiene Pessoal, Perfumaria e cosméticos**. 2022. Disponível em: https://abihpec.org.br/site2019/wp-content/uploads/2021/04/Panorama_do_Setor_Atualizado_Novembro2021_V2.pdf.
- ANDRADE, L.; CUNHA, C. B. Disaggregated retail forecasting: A gradient boosting approach. **Available at SSRN 4129889**, 2022.
- BARTKUS, V. O.; MANNOR, M. J.; CAMPBELL, J. T. Fast and rigorous: Configurational determinants of strategic decision-making balance. **Long Range Planning**, Elsevier, v. 55, n. 3, p. 102142, 2022.
- BARŽIĆ, M. et al. Forecasting sales in retail with xgboost and iterated multi-step ahead method. In: IEEE. **2022 International Conference on Smart Systems and Technologies (SST)**. [S.l.], 2022. p. 153–158.
- BERGMEIR, C.; BENÍTEZ, J. M. On the use of cross-validation for time series predictor evaluation. **Information Sciences**, Elsevier, v. 191, p. 192–213, 2012.
- BREIMAN, L. et al. **Classification and Regression Trees**. Taylor & Francis, 1984. ISBN 9780412048418. Disponível em: <https://books.google.com.br/books?id=JwQx-WOmSyQC>.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.
- DOROGUSH, A. V.; ERSHOV, V.; GULIN, A. Catboost: gradient boosting with categorical features support. **arXiv preprint arXiv:1810.11363**, 2018.
- ESLING, P.; AGON, C. Time-series data mining. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 45, n. 1, p. 1–34, 2012.
- FACELI, K. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: Grupo Gen - LTC, 2011. ISBN 9788521618805.
- FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, JSTOR, p. 1189–1232, 2001.
- HASAN, M. R. et al. A comparative study on forecasting of retail sales. **arXiv preprint arXiv:2203.06848**, 2022.
- KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, 2017.
- KELLEHER, J.; TIERNEY, B. **Data Science**. [S.l.]: MIT Press, 2018. (The MIT Press Essential Knowledge series). ISBN 9780262347037.
- LI, J. A feature engineering approach for tree-based machine learning sales forecast, optimized by a genetic algorithm based sales feature framework. In: IEEE. **2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)**. [S.l.], 2022. p. 133–139.

LOPES, G. The wisdom of crowds in forecasting at high-frequency for multiple time horizons: A case study of the brazilian retail sales. **Brazilian Review of Finance**, v. 20, n. 2, p. 77–115, 2022.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673.

MORETTIN, P.; TOLOI, C. **Análise de séries temporais: modelos lineares univariados**. [S.l.]: BLUCHER., 2018. ISBN 9788521213529.

MULLER, A.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. [S.l.]: O'Reilly Media, Incorporated, 2018. ISBN 9789352134571.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. **Frontiers in neurorobotics**, Frontiers Media SA, v. 7, p. 21, 2013.

NIELSEN, A. **Practical time series analysis: Prediction with statistics and machine learning**. [S.l.]: O'Reilly Media, 2019.

OZDEMIR, S. **Principles of Data Science**. [S.l.]: Packt Publishing, 2016. ISBN 9781785888922.

PROVOST, F.; FAWCETT, T. Data science and its relationship to big data and data-driven decision making. **Big data**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 1, n. 1, p. 51–59, 2013.

_____. **Data Science for Business**. [S.l.]: O'Reilly, 2013. ISBN 978-1-4493-6132-7.

SALTZ, J.; STANTON, J. **An Introduction to Data Science**. [S.l.]: SAGE Publications, 2017. ISBN 9781506377520.

SCHOENHERR, T.; SPEIER-PERO, C. Data science, predictive analytics, and big data in supply chain management: Current state and future potential. **Journal of Business Logistics**, Wiley Online Library, v. 36, n. 1, p. 120–132, 2015.

SEYEDAN, M.; MAFAKHERI, F. Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. **Journal of Big Data**, SpringerOpen, v. 7, n. 1, p. 1–22, 2020.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). **Biometrika**, JSTOR, v. 52, n. 3/4, p. 591–611, 1965.

SOUSA, A. C. C. et al. Comparing gradient boosting algorithms to forecast sales in retail. In: SBC. **Anais do XX Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2023. p. 596–609.

SOUSA, A. S.; SILVA, C. D.; AL., J. D. et. **Análise de séries temporais**. [S.l.]: SAGAH, 2021. ISBN 9786556902876.

SPILOTIS, E. et al. Decision trees for time-series forecasting. **Foresight: The International Journal of Applied Forecasting**, International Institute of Forecasters, n. 64, p. 30–44, 2022.

STEVENSON, W. J. et al. Estatística aplicada à administração. In: **Estatística aplicada a administracao**. [S.l.: s.n.], 1981. p. 495–495.

TAYLOR, S. J.; LETHAM, B. Forecasting at scale. **The American Statistician**, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018.

WU, H. et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. **Advances in Neural Information Processing Systems**, v. 34, p. 22419–22430, 2021.

ZHOU, T. et al. Treedrnnet: A robust deep model for long term time series forecasting. **arXiv preprint arXiv:2206.12106**, 2022.

APÊNDICE A

Comparing Gradient Boosting Algorithms to Forecast Sales in Retail

Ana Clara Chaves Sousa¹, Thaís Gaudencio do Rêgo¹,
Yuri de Almeida Malheiros Barbosa¹, Telmo de Menezes e Silva Filho²

¹Universidade Federal da Paraíba (UFPB)

²University of Bristol, UK

ana.chaves@academico.ufpb.br, gaudenciothais@gmail.com

yuri@ci.ufpb.br, telmo.silvafilho@bristol.ac.uk

Abstract. *The availability of data and the increased processing power of computers have made it easier to make decisions based on data, specially with Artificial Intelligence. One area where AI is widely applicable in companies is Supply Chain Management, particularly in demand forecasting. This paper aims to forecast sales for a company in the Cosmetic, Fragrance, and Toiletry market. Data from 2019 to 2023 were used from two different sales channel. To predict the demand, three Gradient Boosting algorithms (CatBoost, LightGBM, and XGBoost) were compared, and forecasts were made for three different time horizons (next period, five and ten periods ahead). After the experiments, LightGBM showed more stability compared to the other models.*

1. Introduction

Big data and the advancement of computer processing power have facilitated data-driven decision-making, which involves making decisions based on data analysis instead of intuition [Provost and Fawcett 2013a]. Over the last twenty years, there has been a significant amount of investment in business infrastructure, resulting in a better capacity to gather data. Companies are contemplating the possibility of leveraging their data expertise to gain a competitive edge [Provost and Fawcett 2013b].

One area where data-driven decisions can be applied is Supply Chain Management (SCM), which includes logistics, transportation, operations management, procurement, engineering, research, and development [Schoenherr and Speier-Pero 2015]. Customer behavior analysis, trend analysis, and demand forecasting are among the various possibilities in this field [Seyedan and Mafakheri 2020].

In this paper, the focus will be on demand forecasting in a real-world context using Machine Learning techniques. The proposed methodology will be implemented within a company operating in the Cosmetic, Fragrance, and Toiletry (CFT) industry. The name of the company will be kept confidential to preserve its privacy and integrity, which is a necessary condition for continued collaboration in the research and obtaining more detailed and accurate information.

Demand forecasting is a complex task that can have significant impacts on inventory management. If the demand is overestimated, it may cause waste reduction, whereas if it is underestimated, it is possible for stockouts to occur [Andrade and Cunha 2022].

Planning for product availability for consumer purchase requires months of advance planning, depending on the production lead time. The supply chain refers to the series of processes that connect suppliers and customers, as well as the companies involved, from the point of origin of raw materials to the final consumption of the finished product [Cox et al. 1995]. Having said that, demand forecasting affects the entire supply chain by influencing decisions on raw material purchases, storage, and transportation.

The goal of this study is to compare different Gradient Boosting (GB) methods for demand forecasting over multiple time horizons, which are: one period ahead, five periods ahead and ten periods ahead. While comparing the methods, these being CatBoost, LightGBM and XGBoost, it was analyzed which one presented the better performance considering all horizons. GB was used due to the complexity of the data available and its flexibility to handle with categorical features and outliers. The performance of the models will be evaluated using Weighted Absolute Percentage Error (WAPE).

The structure of this paper is presented as follows: In Section 2, five related works on demand forecasting using machine learning techniques are outlined. Section 3 outlines the methodology employed in this study. The results are reported in Section 4. Finally, Section 5 provides concluding remarks.

2. Related Work

Robustness is a big challenge faced in the long-term demand forecast, which makes this task very complex [Zhou et al. 2022]. The model's reliability in detecting seasonality can be affected by patterns found in noisy data and that is the reason why robustness is difficult in this kind of problem [Wu et al. 2021].

A research project used Walmart sales data, which is available in Kaggle's M5 competition dataset, to compare various models for predicting the next 28 days. The dataset contains five years of sales data for three categories and three states in the United States of America. The study examined both parametric and non-parametric models, including Autoregressive Integrated Moving Average (ARIMA), LightGBM, and Prophet. The performance of these models was analyzed and compared, with ARIMA showing the best results presenting a Root Mean Squared Error (RMSE) of 1.09. Although LightGBM had a higher RMSE of 1.18, it was more computationally efficient [Hasan et al. 2022].

Another recent investigation from 2022 also used data from Kaggle's M5 competition to compare tree models: Decision Tree, Random Forest, and Gradient Boosting (GB). The metric used was the Mean Average Percentage Error (MAPE). The model with the lowest error was GB (5.5%), while Random Forest made an error of 5.9% and Decision Tree made an error of 7.5%. The significance of the feature engineering stage in obtaining satisfactory results from tree models was emphasized. Furthermore, this model type is interpretable since its rules can be visualized, enabling decision makers to comprehend the factors that impact the predictions [Spiliotis et al. 2022].

In the same year, a study also discussed the impact of variable engineering on predictive sales analysis. Genetic algorithms were used to achieve the goal. Four tree-based algorithms, Random Forest, XGBoost, LightGBM, and CatBoost, were employed to predict sales for the following month. The performance of the four models was compared using MAPE before and after feature engineering. Then, the models with the lowest errors

(Random Forest and LightGBM) were evaluated for predictions at further time horizons, namely the second and third months. At the conclusion of the study, it was determined that the Random Forest model achieved the best outcomes, with a MAPE of 10.2% in contrast to LightGBM's 10.8%, CatBoost's 13.5%, and XGBoost's 13.8% [Li 2022].

A study was made focusing on comparing the performance of various models across multiple time horizons. The dataset contains all card transactions captured by a Brazilian acquiring company from January 1, 2014, to July 31, 2018, with a total of 1,673 daily observations. Predictions were made for four different time horizons: the next day, seven days ahead, thirty days ahead, and ninety days ahead. The significance of considering holidays, special dates, and other calendar effects was highlighted. Traditional time series forecasting models such as Naïve, HoltWinters, TBATS (acronym for Trigonometric seasonality, Box-Cox transformation, ARIMA errors, Trend, Seasonal components), and Seasonal Autoregressive Integrated Moving Average (SARIMA) were evaluated, along with established Machine Learning and Deep Learning models, such as Multilayer Perceptron (MLP), Long-Short Term Memory (LSTM), XGBoost, and Prophet. Among all the models, Prophet achieved the lowest error at the furthest time horizon, with an RMSE of 17.2 [Lopes 2022].

In another research aimed at assessing the reliability of predictive analytics across various time horizons, XGBoost was employed to generate daily sales volume forecasts for a week, resulting in seven forecasts in total. The data was sourced from a partner company and consisted of sales records for five products. The performance of the model was evaluated using mean absolute error (MAE) and weighted mean absolute error (WMAE) metrics. Four of the five products showed higher errors in the forecasts on day 7 compared to day 1. The average WMAE was found to be 2.76, while the MAE was 2.44 [Baržić et al. 2022].

The primary commonality between the current research and previous studies is their focus on predicting sales within the retail industry using at least one Gradient Boosting algorithm. Nevertheless, there are also discrepancies. The studies conducted by [Hasan et al. 2022] and [Spiliotis et al. 2022] did not explore forecasts across various time horizons. [Li 2022] performed an experiment for distant time horizons of up to three months but did not focus on it. Although [Baržić et al. 2022] also compared sales volume forecasts at different time intervals, they only did so for a maximum of one week. The study that is most similar to the current one is [Lopes 2022], but an essential distinction to note is that while [Lopes 2022] had a maximum time horizon equivalent to three months, the present study has a more extended term forecast of around eight months ahead.

A significant differentiation between the current study and previous research is the presence of distinct contextual factors and data characteristics. Unlike previous studies that typically dealt with regularly spaced time series data (such as daily or weekly data), this study focuses on a time context aligned with business strategies, referred to as cycles. These cycles do not have equal intervals, which poses challenges when employing traditional time series methods. Furthermore, the study encompasses diverse consumer attraction campaigns and considers products that are not classified as essential commodities.

3. Methodology

The methodology of this work is described in Figure 1, with all the steps used to conduct the study.

1. **Data Preprocessing:** All data cleaning and formatting were carried out in this step, which is detailed in Section 3.3.
2. **Feature Engineering:** Section 3.4 provides a detailed explanation of this step. Creating new variables is particularly important for tree-based methods, as previously mentioned in Section 2. In addition, they are necessary to forecast sales when traditional time series methods are not being used.
3. **Data Preparation for Multiple Time Horizons:** In this step, three datasets were created separately for each time horizon. Details are provided in Section 3.5.
4. **Hyperparameter Optimization with Validation Set:** This step involved separating the data into several training and validation sets for each horizon and model, and then the hyperparameters were optimized. Details are also in Section 3.5.
5. **Model Training and Testing:** In this step, the GB algorithms were trained and tested. Section 3.5 provides further details on this part.
6. **Model Evaluation:** Finally, an analysis was carried out to understand the errors of the models and compare them, as outlined in Section 4.

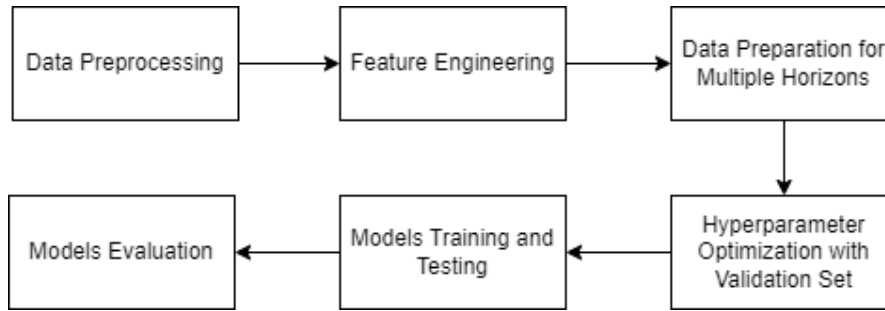


Figure 1. Methodology.

Section 3.1 shows information about the data used in this study. Section 3.2 shows more detailed information about the machine that was used to run this work and other tools, such as programming languages, libraries and cloud.

3.1. Data

This study used a dataset from a partner company in the CFT market, which included originally around 8.8 million instances (in the cycle/state/channel/product level) and 16 variables. The granularity of the raw data can be seen with more details in Table 1. The data pertained to sales made between 2018 and 2023. The data from 2018 was excluded from the training set and used solely for generating lag features, as described in Section 3.4. The data available from 2023 was used as holdout set to test the models.

Table 1 displays the available raw variables. To conduct the study, products were grouped by brand, subcategory, and category. Consequently, the modeling was carried out based on the following granularity: cycle, channel, state, category, subcategory, and brand.

Table 1. Variables available in the dataset.

Feature	Description
Cycle	Period of the year, with each year consisting of 17 cycles.
Cycle description	Cycle type definition.
Date of cycle start	Date on which cycle was started.
Date of cycle end	Date cycle ends.
Channel	Sales channel, may be Direct Sales or Stores.
State	Federative Unit of Brazil.
Category	Category of the product.
Subcategory	Subcategory of the product.
Brand	Product brand.
Product code	Product code/SKU.
Sold amount	Target. Quantity of products sold.
Practiced value	Sale value with discounts.
Table value	Sale value without discounts.
Discount value	Discount value applied.
Discount percentage	Discount percentage.
Unit Price	Unit price considering the discounts.
Promotion	Indicates if the product is promoted or not.
Focus	Indicates if the product is in focus campaign or not.
Subfocus	Indicates if the product is in campaign subfocus or not.

3.2. Hardware and Technologies

To run the experiments, JupyterLab on the GCP (Google Cloud Platform) workbench was used. The hardware utilized was 8 vCPUs and 30 GB RAM. Python and SQL were the programming languages used in this study. The following libraries were employed within Python 3.7:

- **pandas**, for manipulating the data through dataframes;
- **numpy**, for performing mathematical operations and using arrays;
- **matplotlib.pyplot**, for data visualization through charts;
- **seaborn**, for data visualization as well;
- **google.cloud**, for connecting the data to Google BigQuery;
- **lightgbm**, for running the experiments using the LightGBM algorithm;
- **xgboost**, to run the experiments using the XGBoost algorithm;
- **catboost**, for the execution of experiments using the CatBoost algorithm;
- **sklearn**, for performing general applications of Machine Learning;
- **optuna**, for hyperparameter optimization.

3.3. Data Preprocessing

During this step, the data was initially formatted according to each variable. Since the data had been previously structured by the Data Engineering team of the company, not much processing was required at this stage. Following the formatting process, the missing values were analyzed and their proportion was determined concerning to the complete dataset. Subsequently, the rows with the missing values were removed. In total, the dataset was reduced by only 0.28% after cleaning the nulls.

In addition to cleaning missing data, a further cleaning process was carried out to eliminate deactivated products, categories, subcategories, and brands. This was done because there was no need to forecast demand for products that were no longer part of the company's portfolio. As a result of this cleaning process, 5.78% of the dataset was removed.

As stated in Section 3.1, the modeling was conducted at the following levels of granularity: cycle, channel, state, category, subcategory, and brand. It is worth noting that the data were grouped at this level, resulting in a reduced dataset of approximately 950k instances.

3.4. Feature Engineering

At this stage, different features were created to serve as input to the models. Table 2 shows the features that were created considering the products, promotions and campaigns.

Table 2. Features created considering the products, promotions and campaigns.

Original Features	Features Created	Description
Product, Cycle, Category, Subcategory, Brand	Quantity of Products (1 feature)	How many products are in that category/subcategory/brand.
Promotion, Cycle, Product, Category, Subcategory, Brand, State	Products on Promotion (1 feature)	Quantity of products on promotion within the primary key.
Discount Percentage, Cycle, Product, Category, Subcategory, Brand, State	Discount Range (4 features created)	Number of products belonging to each discount range (0%, 10%, 20%, 30%) within the primary key.
Focus, Cycle, Product, Category, Subcategory, Brand, State	Products in Focus (1 feature)	Quantity of products in focus within the primary key.
Subfocus, Cycle, Product, Category, Subcategory, Brand, State	Products in Subfocus (1 feature)	Quantity of products in subfocus.

Lag features were constructed since they are the ones that make the regression time-aware, that is, that extract properties that can sort the data. The reason for creating such features is that GB methods are not like traditional time series models. Therefore, to enable the models to make predictions, it is necessary to create this type of feature, as pointed out by [Bergmeir and Benítez 2012]. The lag features can be seen in Table 3.

21 features were created considering the cycle used in prediction. It is crucial to emphasize that there is no time leap in the first time horizon prediction, meaning that the cycle to be predicted and the cycle used in prediction are identical. Table 4 shows the features built over cycle information, whereby three of them are used in the next cycle prediction and six are used in the other horizons.

Table 3. Lag features created considering the cycle used in prediction.

Original Features	Features Created	Description
Amount Sold, Cycle, Product, Category, Subcategory, Brand, State	Target 1 year lag (1 feature)	Amount sold in the year before.
Amount Sold, Cycle, Product, Category, Subcategory, Brand, State	Target 1 to 10 cycle lag (10 features)	Amount sold in 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 cycles before.
Amount Sold, Cycle, Product Code, Category, Subcategory, Brand, State	Target moving average of 5 and 10 cycles (2 features)	Amount sold moving average considering the previous 5 and 10 cycles.

Table 4. Features created considering cycle information.

Original Features	Features Created	Description
Cycle Start Date, Cycle End Date	Cycle Length (1 feature)	Cycle duration in days, usually 21 days. (Cycle used in prediction)
Cycle Start Date, Cycle End Date	Cycle Length (1 feature)	Cycle duration in days, usually 21 days. (Cycle to be predicted)
Cycle	Cycle Year, Cycle Number (2 features)	Information about the cycle used in prediction.
Cycle	Cycle Year, Cycle Number (2 features)	Information about the cycle to be predicted.

Other feature related to cycle's information that was used as input to the model is the cycle description. It was not showed in Table 4 because it is an original feature from the data. For 5 and 10 cycles ahead, there is also the description of the cycle to be predicted.

3.5. Data Modeling

This section describes three of the main steps showed in Figure 1: data preparation for multiple time horizons, hyperparameter optimization, training the models and testing them.

The time horizons predictions were made separately, so there are three datasets: one for predicting the next period, one for predicting 5 cycles ahead and one for predicting 10 cycles ahead. The next cycle prediction dataset had 31 input features, that being:

- 6 features regarding the granularity (cycle, state, channel, category, subcategory and brand);
- 1 original feature with the cycle description;
- 3 features created considering the cycle information (Table 4);
- 8 features created regarding the products, promotions and campaigns (Table 2);
- 13 lag features (Table 3).

The predictions for 5 and 10 cycles ahead had 45 input features each, that being:

- 6 features regarding the granularity (cycle, state, channel, category, subcategory and brand);
- 1 feature with the cycle to be predicted;
- 2 original features with the description of the cycle used in prediction and the cycle to be predicted;
- 6 features created considering the cycle information (Table 4) for both cycle used in prediction and cycle to be predicted;
- 8 features created regarding the products, promotions and campaigns considering the cycle used in prediction (Table 2);
- 8 features created regarding the products, promotions and campaigns considering the cycle to be predicted (the same shown in Table 2);
- 13 lag features (Table 3) considering the cycle used in prediction;
- 1 feature with the amount sold in the cycle used in prediction;

It is worth noting that all 14 additional features used for predicting future cycles will be available in the future, enabling their utilization in the models. At the time of making the prediction, all relevant information is known, including details such as the number of products in the portfolio, the specific products to be promoted, the corresponding discount amounts, information about cycle characteristics, and more.

For hyperparameter optimization, the data was divided into four training sets and four validation sets for each horizon. The validation sets had six cycles each with the most recent data as it is a forecasting problem. The training sets had different sizes for each horizon: 43 cycles for the next cycle predicted, 38 cycles for the 5 cycles ahead prediction and 33 cycles for the 10 cycles ahead prediction. Figure 2 shows how the data was split off.

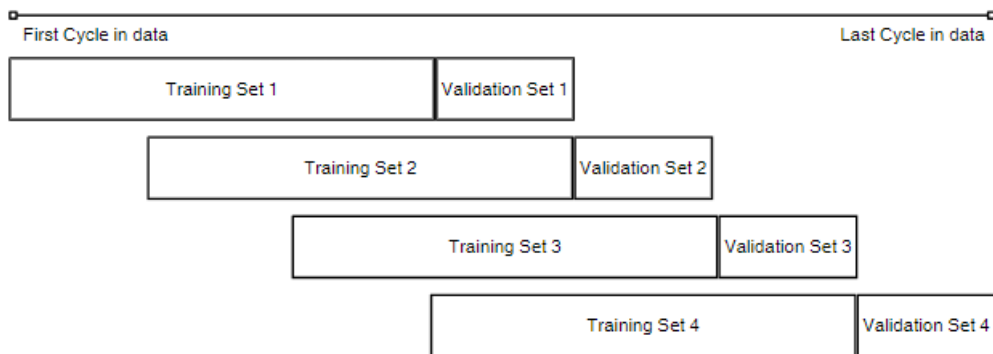


Figure 2. Training and Validation Sets for Hyperparameter Optimization.

In this step, optuna was used and the goal was to find the best hyperparameters that minimize the Weighted Absolute Percentage Error (WAPE). Its calculation is showed in

Equation (1). Since it is weighted, the forecast errors in products with more sales would be larger than in products with fewer sales. WAPE can range from 0% to any positive number in percent, the lower the better.

$$\frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|}, \quad (1)$$

where A_t and F_t are the observed and predicted values at instant t , respectively, and n is the maximum instant of time.

Five trials were made for each model and horizon. Table 5 shows the hyperparameters that were optimized and its ranges, as well as what each hyperparameter does. Unlike LightGBM and XGBoost, CatBoost uses a strategy called lossguide for the construction of trees, so it is not possible to specify the max_leaves hyperparameter for it.

Table 5. Range of hyperparameters to be optimized.

Hyperparameter	Description	Range
max_leaves / num_leaves	It limits the maximum amount of leaves a tree can have.	[20, 3000]
max_depth	Limits the maximum depth of decision trees.	[3, 12]
min_data_in_leaf / min_child_weight	Specifies the minimum number of samples required in a bin to be considered valid in histogram-based methods.	[100, 2000]
n_estimators / num_boost_rounds	Defines the maximum number of decision trees to be built.	[80, 300]
learning_rate	Controls the rate at which the model learns during training.	[0.01, 0.3]

Table 6 shows the final hyperparameters for CatBoost, Table 7 presents the hyperparameters used for LightGBM and Table 8 displays the XGBoost hyperparameters. The categorical features hyperparameter was also used in all models, so there was no need to encode the categories in the data preprocessing stage. All other hyperparameters were set as default.

Table 6. CatBoost hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
max_depth	9	11	9
min_data_in_leaf	243	1462	1982
n_estimators	140	130	247
learning_rate	0.116	0.179	0.238
random_state	42	42	42

Table 7. LightGBM hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
num_leaves	570	2275	2626
max_depth	6	10	9
min_data_in_leaf	707	937	205
n_estimators	229	185	219
learning_rate	0.190	0.221	0.164
random_state	42	42	42

Table 8. XGBoost hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
max_leaves	1322	2382	2131
max_depth	8	7	9
min_child_weight	1291	313	323
num_boost_rounds	211	147	277
learning_rate	0.130	0.103	0.208
seed	42	42	42
tree_method	hist	hist	hist

After the hyperparameter optimization, the final models were trained using data from 2019 to 2022. Six cycles from 2023 were used as test set. The final training sets and test sets were separated considering the cycle to be predicted.

4. Results and Discussions

The metric chosen to evaluate and compare the three models results through the different time horizons was WAPE, as mentioned in Section 1.

Table 9 presents the WAPE for each channel and for each time horizon. Overall, it can be observed that the predictions for the store channel tend to be more accurate compared to direct sales. Among the three models, XGBoost consistently achieved the lowest error across all prediction horizons for the store channel. LightGBM also demonstrated a strong performance in predicting for the store channel, while CatBoost showed a significantly higher error when predicting for the 10 cycles horizon compared to the other two models.

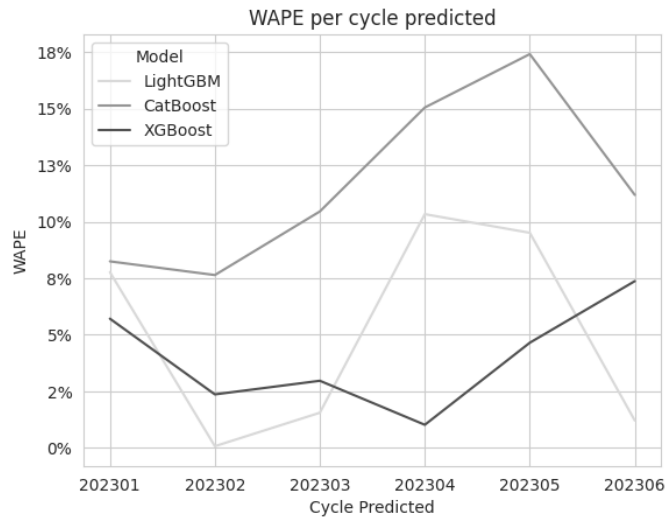
When examining the errors for direct sales, LightGBM yielded superior results in predicting 10 cycles ahead. On the other hand, XGBoost surpassed the other models in the remaining two time horizons. Regarding the range of error, XGBoost showed almost the same variation for both channels: 2.97% (stores) and 2.96% (direct sales). CatBoost had the highest error range in both channels (7.76% for stores and 6.73% for direct sales), while LightGBM had the lower error range in stores (2.31%) and a value of 4.97% in direct sales. CatBoost had the least acceptable results taking into account not just the channels but also the different horizons.

Table 9. WAPE per channel.

Model	Channel	Next cycle	5 cycles ahead	10 cycles ahead
CatBoost	Stores	3.49%	3.12%	10.88%
LightGBM	Stores	4.22%	2.77%	5.08%
XGBoost	Stores	3.07%	0.10%	2.62%
CatBoost	Direct Sales	17.37%	17.46%	10.73%
LightGBM	Direct Sales	10.77%	13.80%	8.83%
XGBoost	Direct Sales	8.75%	11.71%	9.91%

To discuss the results for each cycle in each one of the three horizons, Figures 3, 4 and 5 should be investigated. Figure 3 shows the cycles predicted considering the next period ahead, Figure 4 shows the cycles to be predicted in 5 periods ahead and Figure 5 has the 10 cycles ahead that were predicted. For each test set, as discussed in Section 3.5, there are six cycles. The results from cycles 202301 (first cycle of 2023) to 202306 (sixth cycle of 2023) were evaluated.

For the closest prediction (Figure 3), all errors are below 18%. While LightGBM's and XGBoost's errors are all lower than 11%, CatBoost shows peaks in certain cycles. Both LightGBM and XGBoost had 3 out of 6 cycles with the lowest error, whereas CatBoost presents higher errors in all cycles. Once more, both LightGBM and XGBoost had better results.

**Figure 3. WAPE per cycle predicted: next cycle.**

When predicting 5 cycles ahead, Figure 4 illustrates a notable peak in cycle 202304 across all models. Regarding the range of errors, when comparing the maximum and minimum WAPE for each model, all ranges fall between 13% and 18%. These results indicate no significant differences among the models in this particular scenario.

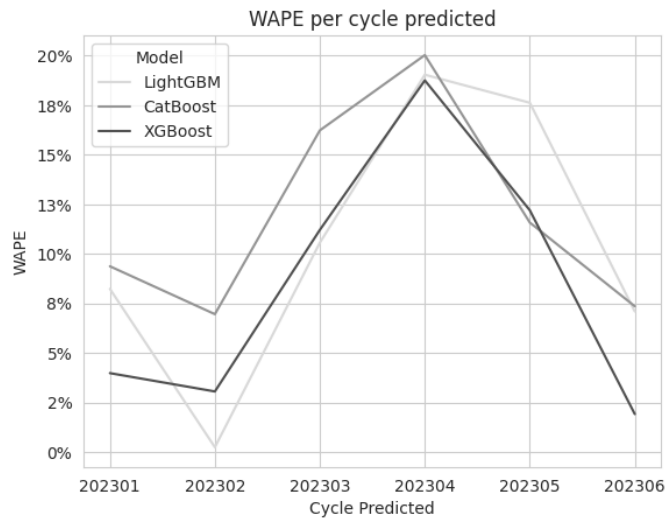


Figure 4. WAPE per cycle predicted: 5 cycles ahead.

For the long-term forecast, 10 cycles ahead were predicted (equivalent to eight months). It is possible to see the WAPE in the test set in Figure 5. All models shows a peak in the last cycle predicted, but LightGBM had a smaller error increase from one cycle to the next. CatBoost had the lower error in cycles 202303, 202304 and 202305. LightGBM was better in 202301 and 202306. XGBoost had a lower WAPE only in cycle 202302. In terms of error range, LightGBM has a range of 6.9%, followed by 10.1% in XGBoost and 11.2% in CatBoost.

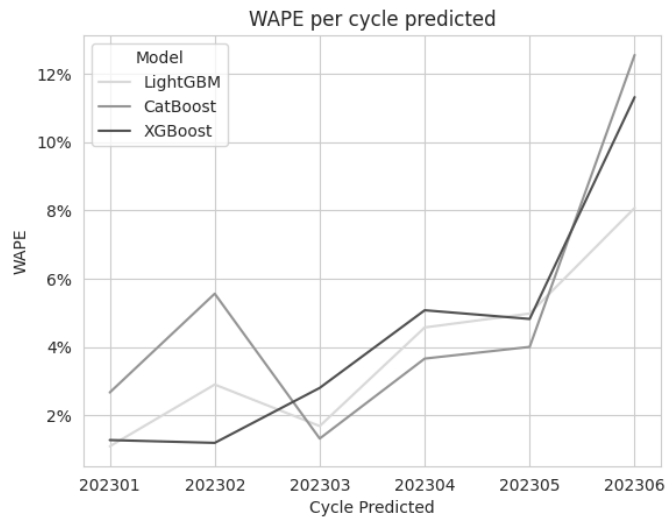


Figure 5. WAPE per cycle predicted: 10 cycles ahead.

As the goal of this paper is to compare three GB models over three time horizons, the model consistency considering the different scenarios is very important. The model that showed more stability between the analysis was LightGBM. XGBoost had a good overall performance as well, but Figure 5 shows that LightGBM is superior, especially because the biggest concern of this study is the long-term forecasting.

Since the feature engineering stage was a focus on the study, it is important to understand how the features had an impact on the most stable model performance. Figure 6 shows a feature importance chart for the LightGBM model predicting 10 cycles ahead. The features were divided between two big groups: original features and created features. Within the created features shown in the chart, three of them were lag features and the other three were some of the ones that were built upon information about products, promotions and campaigns. The names of the features will not be displayed due to confidentiality. It is evident that out of the top 10 most significant features for the model, 6 of them are derived from the applied methodology. This shows that the methodology used with a strong focus on feature engineering had a considerable impact on the results.

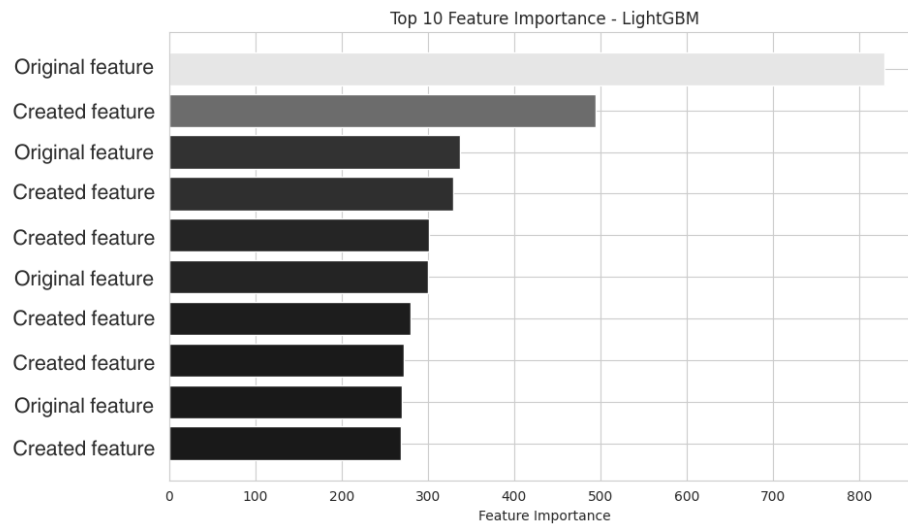


Figure 6. Feature Importance for LightGBM - 10 cycles ahead.

5. Conclusion

Demand forecasting is critical in various sectors and markets. For companies in retail, for example, knowing which and how many products are expected to be sold allow the business to be prepared to meet the customers needs. To forecast demand in a long-term scenario, it is even more challenging because of robustness [Zhou et al. 2022]. The aim of this research was to predict future demand using a real-world dataset in multiple time horizons, namely one period, five periods, and ten periods ahead. To achieve this goal, three Gradient Boosting algorithms (CatBoost, LightGBM, and XGBoost) were assessed, with a particular focus on feature engineering. The solution introduced in this paper indicated that LightGBM is able to provide good predictions in different steps ahead with consistency. XGBoost was also a remarkable alternative solution. In future work, enhancements could be derived from implementing the presented study across various retail datasets and also see how it behaves against other methodologies, such as ARIMA, Prophet and Deep Learning techniques.

References

- Andrade, L. and Cunha, C. B. (2022). Disaggregated retail forecasting: A gradient boosting approach. *Available at SSRN 4129889*.
- Baržić, M., Munitić, N.-F., Bronić, F., Jelić, L., and Lešić, V. (2022). Forecasting sales in retail with xgboost and iterated multi-step ahead method. In *2022 International Conference on Smart Systems and Technologies (SST)*, pages 153–158. IEEE.
- Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.
- Cox, J., Blackstone, J., Spencer, M., Production, A., and Society, I. C. (1995). *APICS Dictionary*. American Production and Inventory Control Society.
- Hasan, M. R., Kabir, M. A., Shuvro, R. A., and Das, P. (2022). A comparative study on forecasting of retail sales. *arXiv preprint arXiv:2203.06848*.
- Li, J. (2022). A feature engineering approach for tree-based machine learning sales forecast, optimized by a genetic algorithm based sales feature framework. In *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 133–139. IEEE.
- Lopes, G. (2022). The wisdom of crowds in forecasting at high-frequency for multiple time horizons: A case study of the brazilian retail sales. *Brazilian Review of Finance*, 20(2):77–115.
- Provost, F. and Fawcett, T. (2013a). Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59.
- Provost, F. and Fawcett, T. (2013b). *Data Science for Business*. O'Reilly.
- Schoenherr, T. and Speier-Pero, C. (2015). Data science, predictive analytics, and big data in supply chain management: Current state and future potential. *Journal of Business Logistics*, 36(1):120–132.
- Seyedan, M. and Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, 7(1):1–22.
- Spiliotis, E. et al. (2022). Decision trees for time-series forecasting. *Foresight: The International Journal of Applied Forecasting*, (64):30–44.
- Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430.
- Zhou, T., Zhu, J., Wang, X., Ma, Z., Wen, Q., Sun, L., and Jin, R. (2022). Tree-drnet: A robust deep model for long term time series forecasting. *arXiv preprint arXiv:2206.12106*.

APÊNDICE B



PREVISÃO DE DEMANDA DE LONGO PRAZO APLICADA A UMA EMPRESA DE VAREJO UTILIZANDO O PROPHET

LONG-TERM DEMAND FORECASTING APPLIED TO A RETAIL COMPANY USING PROPHET

PREVISIÓN DE LA DEMANDA A LARGO PLAZO APLICADA A UNA EMPRESA MINORISTA MEDIANTE PROPHET

Ana Clara Chaves Sousa¹, Yuri de Almeida Malheiros Barbosa², Telmo de Menezes e Silva Filho³ & Thais Gaudencio do Rêgo⁴

^{1 2 4} Universidade Federal da Paraíba (UFPB), Centro de Informática

⁴ University of Bristol, UK

¹ ana.chaves@academico.ufpb.br ² yuri@ci.ufpb.br ³ telmo.silvafilho@bristol.ac.uk ⁴ gaudenciothais@ci.ufpb.br

ARTIGO INFO.

Recebido:

Aprovado:

Disponibilizado:

PALAVRAS-CHAVE: Gestão da Cadeia de Abastecimento; Previsão de Demanda; Ciência de Dados; Inteligência Artificial; Aprendizado de Máquina.

KEYWORDS: Supply Chain Management; Demand Forecast; Data Science; Artificial intelligence; Machine Learning.

PALABRAS CLAVE: Gestión de la cadena de suministro; Pronóstico de Demanda; Ciencia de los datos; Inteligencia artificial; Aprendizaje automático.

*Autor Correspondente: Sousa, A. C. C.

RESUMO

A tomada de decisão baseada em dados se difundiu devido à alta disponibilidade de dados e da maior capacidade de processamento dos computadores. Para auxiliar na tomada de decisão, é possível extrair conhecimentos dos dados e fazer previsões através da Ciência de Dados. Dentro da Gestão da Cadeia de Abastecimento, uma área que está conectada à Engenharia de Produção, um desafio comum é fazer a previsão de demanda utilizando dados históricos. Para realizar essa previsão, técnicas de Inteligência Artificial têm sido amplamente utilizadas, especialmente o Aprendizado de Máquina. Neste estudo de caso, será feita a previsão de demanda de longo prazo (30 semanas) para dois canais de vendas diferentes de uma empresa do segmento da indústria e do varejo do mercado de Higiene Pessoal, Perfumaria e Cosméticos. Neste estudo, foi utilizado o algoritmo Prophet. Após implementada a metodologia, os resultados mostraram que, para o canal Loja, a semana 30 apresentou um WAPE de 4%. No canal Venda Direta, o erro na trigésima semana foi equivalente a 5%.

ABSTRACT

Data-driven decision making has become widespread due to the high availability of data and the increased processing

capacity of computers. To aid decision-making, it is possible to extract knowledge from data and make predictions through Data Science. Within Supply Chain Management, an area that is connected to Production Engineering, a common challenge is to forecast demand using historical data.

Artificial Intelligence techniques have been widely used to carry out this forecasting, especially Machine Learning. In this case study, long-term demand forecasting (30 weeks) will be carried out for two different sales channels of a company in the industrial and retail segments of the Cosmetic, Fragrance and Toiletry market. The Prophet algorithm was used in this study. After implementing the methodology, the results showed that, for the Store channel, week 30 had a WAPE of 4%. In the Direct Sales channel, the error in the thirtieth week was equivalent to 5%.

RESUMEN

La toma de decisiones basada en datos se ha generalizado debido a la gran disponibilidad de datos y a la mayor capacidad de procesamiento de los ordenadores. Para facilitar la toma de decisiones, es posible extraer conocimientos de los datos y hacer predicciones mediante la Ciencia de Datos. Dentro de la Gestión de la Cadena de Suministro, un área relacionada con la Ingeniería de Producción, un reto habitual es la previsión de la demanda a partir de datos históricos.

Para llevar a cabo esta previsión, se han utilizado ampliamente técnicas de Inteligencia Artificial, especialmente Machine Learning. En este caso de estudio, se realizará una previsión de la demanda a largo plazo (30 semanas) para dos canales de venta diferentes de una empresa de los segmentos industrial y minorista del mercado de Higiene Personal, Perfumería y Cosmética. En este estudio se utilizó el algoritmo Prophet. Tras aplicar la metodología, los resultados mostraron que, para el canal Tienda, la semana 30 tuvo un WAPE del 4%. En el canal de Venta Directa, el error en la semana 30 fue equivalente al 5%.

1. INTRODUÇÃO

A disponibilidade de grandes volumes e variabilidade de dados, juntamente com a ampliada capacidade de processamento computacional, propiciaram o que é denominado como tomada de decisão baseada em dados (do inglês, *data-driven decision making*) (Provost; Fawcett, 2013). *Data-driven decision making* é a prática de realizar escolhas fundamentadas na análise de dados, em contraposição à intuição (Provost; Fawcett, 2013).

A Ciência de Dados é um campo voltado para a coleta, preparação, análise, visualização, gestão e preservação de extensos conjuntos de dados (Saltz; Stanton, 2017). Foi concisamente delineada como a arte e ciência de adquirir conhecimento por meio de dados. Refere-se à maneira como os dados são empregados para extrair conhecimento (Ozdemir, 2016).

Pode-se afirmar que a Ciência de Dados compreende três fundamentos essenciais:

- Matemática e Estatística: Envolve a compreensão de equações, funções e fórmulas para realizar análises e conduzir modelagens matemáticas.
- Programação de Computadores: Diz respeito à habilidade de codificar em uma linguagem de programação para desenvolver soluções.
- Domínio do Negócio: Refere-se à compreensão do setor e do problema a ser resolvido, abrangendo áreas como medicina, finanças, indústria, entre outras. (Ozdemir, 2016)

Por meio da obtenção de informações derivadas de técnicas de Aprendizado de Máquina, torna-se viável realizar tomadas de decisão, efetuar previsões, compreender tanto o presente quanto o passado, e desenvolver novos produtos (Ozdemir, 2016). Dentre as várias esferas em que decisões fundamentadas em dados podem ser aplicadas, a Gestão da Cadeia de Abastecimento (do inglês, *Supply Chain Management* - SCM) é um exemplo notável. A abrangência da área de SCM compreende logística, transporte, gestão de operações, compras, engenharia, pesquisa e desenvolvimento (Schoenherr; Speier-Pero, 2015).

Dentro deste domínio, inúmeras oportunidades se apresentam, como a análise do comportamento do cliente, a avaliação de tendências e a previsão de demanda (Seyedan; Mafakheri, 2020). Schoenherr e Speier-Pero (2015) realizaram uma pesquisa com o propósito de enfatizar as vantagens derivadas da aplicação da análise preditiva na área, utilizando dados obtidos de profissionais de SCM. Alguns exemplos desses benefícios são: aprimoramento na tomada de decisões, aumento da visibilidade, aprimoramento no gerenciamento de riscos na cadeia de suprimentos, aperfeiçoamento na eficiência da cadeia de suprimentos, otimização dos custos na cadeia de suprimentos, aprimoramento de recursos para o planejamento de demanda, aprimoramento de recursos para o planejamento de vendas e operações.

Com base nisso, o tema deste trabalho é a previsão de demanda de longo prazo (30 semanas à frente) utilizando um algoritmo de Aprendizado de Máquina, um método de séries temporais chamado Prophet (Taylor; Letham, 2018). Serão utilizados dados de uma grande empresa de varejo e indústria do mercado de Higiene Pessoal, Perfumaria e Cosméticos (do inglês, *Cosmetic, Fragrance and Toiletry* - CFT). As previsões serão feitas para dois canais de

venda: Loja e Venda Direta. A fim de preservar a privacidade e integridade da empresa em questão, seu nome será mantido em sigilo, assim, será referida como Empresa X. Este trabalho é complementar a um estudo publicado pelos autores no Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2023) (Sousa et al., 2023). Enquanto o artigo já publicado mostra um comparativo entre algoritmos de Gradient Boosting em múltiplos horizontes temporais, este trabalho estuda o método Prophet para a previsão de longo prazo.

A estrutura deste estudo será dividida em seis seções, incluindo a Introdução. A Seção 2 fundamenta conceitos gerais. Na Seção 3, são apresentados alguns trabalhos relacionados. A Seção 4 detalha a metodologia empregada. Na Seção 5, os resultados são discutidos. Por fim, a Seção 6 apresenta as considerações finais.

2. FUNDAMENTAÇÃO TEÓRICA

Para contextualizar os conceitos gerais importantes para o entendimento deste estudo, esta seção foi dividida em três partes:

- 2.1. Aprendizado de Máquina: Definição de Aprendizado de Máquina e seus diferentes tipos, com exemplos.
- 2.2. Séries Temporais: Explicação de séries temporais.
- 2.3. Prophet: Contextualização do funcionamento do algoritmo utilizado no estudo, o Prophet do Facebook.

2.1. APRENDIZADO DE MÁQUINA

Aprendizado de Máquina (*Machine Learning*, em inglês) é um campo de pesquisa que está na interseção de Inteligência Artificial, Ciência da Computação e Estatística e é também conhecido como análise preditiva ou aprendizado estatístico. Está presente na vida cotidiana das mais diversas formas, como em recomendações em *streaming* sobre quais filmes ou séries assistir e reconhecimento facial. Os algoritmos de ML mais conhecidos são aqueles que automatizam os processos de tomada de decisão, encontrando padrões a partir de dados históricos conhecidos (Muller; Guido, 2018). Há muitos fatores que têm favorecido a expansão dos estudos na área de ML, como o desenvolvimento de algoritmos cada vez mais eficazes, a elevada capacidade dos recursos computacionais disponíveis atualmente e o grande volume de aplicações que são beneficiadas (Faceli, 2011).

Existem diversas definições para Aprendizado de Máquina na literatura, uma delas é “a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência” (Mitchell, 1997). Sendo assim, os computadores são programados para aprender com a experiência passada, empregando um princípio de inferência denominado indução. Conclusões genéricas são constatadas a partir de um conjunto de exemplos. Os algoritmos de ML aprendem a induzir uma função ou hipótese que seja capaz de resolver um problema, a partir de dados representativos de instâncias desse problema a ser resolvido (Faceli, 2011). A Inteligência Artificial não é a única área de pesquisa que é associada ao Aprendizado de

Máquina. Outros exemplos são: Probabilidade e Estatística, Neurociência, Teoria da Computação e Teoria da Informação (Faceli, 2011).

Ao aprender a partir dos dados de treinamento, o algoritmo de ML está procurando por uma hipótese que melhor se ajuste a esses dados, dentro de um espaço de hipóteses possíveis, sendo também necessário que essa hipótese seja capaz de descrever as relações entre os objetos. Um algoritmo de ML tem dois vieses: um viés de representação e outro de busca. As hipóteses que serão visitadas no espaço são restringidas pelo viés, ou seja, sem viés não há generalização e aprendizado (Faceli, 2011).

Os algoritmos de ML podem ser utilizados para tarefas de natureza diferentes que podem ser organizadas de acordo com alguns critérios. Um desses critérios é sobre o paradigma de aprendizado a ser adotado para executar uma tarefa. De acordo com ele, as tarefas podem ser divididas em preditivas e descritivas. Em modelos preditivos, o objetivo é encontrar uma função, a partir dos dados de treinamento, que possa ser utilizada para prever um rótulo ou valor que caracterize um novo exemplo, tomando como base seus atributos de entrada. Em modelos descritivos, o objetivo é explorar ou descrever um conjunto de dados (Faceli, 2011).

Existem diferentes tipos de Aprendizado de Máquina, dentre eles, o Aprendizado Supervisionado e o Não Supervisionado. Existe, ainda, uma combinação dos dois: o Aprendizado Semi-Supervisionado (Brownlee, 2016).

O Aprendizado Supervisionado é o mais comum, pois acontece quando existem variáveis de entrada (X) e uma variável de saída (Y), em que um algoritmo é utilizado para aprender a função de mapeamento entre eles. Resumidamente, o objetivo é aproximar a função quando há novos dados de entrada, de forma que a variável Y seja predita. O nome "Aprendizado Supervisionado" vem do contexto que um algoritmo irá aprender a partir de conjunto de dados de treinamento (Brownlee, 2016).

O Aprendizado Supervisionado, por sua vez, pode ser dividido em dois tipos de modelo:

- Classificação: A variável a ser prevista é uma categoria, podendo ser: um diagnóstico positivo (ou não) para uma doença, a espécie de uma planta, etc.
- Regressão: A variável a ser prevista é um valor, podendo ser: valor do aluguel de um apartamento em uma determinada área, temperatura, etc. (Brownlee, 2016)

O Aprendizado Não Supervisionado é quando temos disponíveis apenas os dados de entrada (X), sem uma variável Y correspondente. O objetivo é aprender sobre a similaridade e a estrutura dos dados de forma que seja possível entendê-los melhor. O algoritmo irá explorar os dados até descobrir padrões, como, por exemplo, agrupar clientes por comportamento de compra (Brownlee, 2016).

O Aprendizado Semi-Supervisionado acontece quando existe um conjunto de dados de entrada (X) e apenas uma parcela desses dados possui rótulo (Y). Um bom exemplo é um conjunto de imagens de animais, em que algumas estão rotuladas (cachorro, gato, coelho, etc) e o restante não possui rótulo algum (Brownlee, 2016).

A Seção 2.2 fundamenta os conceitos de séries temporais, que são importantes para melhor entendimento do método utilizado neste trabalho, o Prophet.

2.2. SÉRIES TEMPORAIS

A análise de séries temporais encontra aplicação em diversas áreas. No âmbito econômico, por exemplo, é possível monitorar o Produto Interno Bruto (PIB) e a oferta e demanda. Ademais, é amplamente empregada para rastrear fenômenos naturais, análise demográfica, estratégias de marketing, controle de qualidade e em contextos relacionados à saúde. Nesse sentido, a análise de séries temporais emerge como uma ferramenta valiosa para a ciência de dados (Sousa; Silva; AL., 2021).

A finalidade da análise de séries temporais consiste em extrair informações relevantes de conjuntos de dados organizados cronologicamente, diagnosticando comportamentos passados e prevendo comportamentos futuros (Nielsen, 2019). Uma Série Temporal frequentemente representa a observação de um processo subjacente, a partir do qual os valores são coletados por meio de medições realizadas em instantes de tempo específicos. Essa série pode ser univariada ou multivariada (Esling; Agon, 2012).

De maneira geral, uma série temporal pode ser categorizada em quatro componentes, os quais podem se manifestar de forma isolada ou combinada (Sousa; Silva; AL., 2021):

- **Tendência:** Refere-se à observação de um crescimento ou queda na evolução da variável ao longo do tempo.
- **Ciclo:** Consiste em movimentos ascendentes ou descendentes em uma linha temporal, ocorrendo ciclicamente dentro de uma tendência.
- **Sazonalidade:** Caracteriza-se por movimentos recorrentes na linha temporal, manifestando-se repetidamente em períodos específicos.
- **Irregularidade:** Corresponde ao ruído presente no modelo de série temporal. Qualquer aspecto que não represente uma tendência, um ciclo ou uma sazonalidade é considerado um erro aleatório, ou seja, uma irregularidade.

A análise de séries temporais é frequentemente conduzida por meio de métodos estatísticos convencionais e por algumas técnicas de Aprendizado de Máquina, tais como modelos de árvores e ajustes lineares (Nielsen, 2019).

A Seção 2.3 contextualiza o método utilizado neste estudo, o Prophet.

2.3. PROPHET

O Prophet emprega um modelo de série temporal decomponível, composto por três componentes principais: tendência, sazonalidade e feriados, que são combinados na seguinte equação (Taylor; Letham, 2018):

$$y(t) = g(t) + s(t) + \epsilon_t$$

onde $g(t)$ corresponde à função de tendência que modela as variações não periódicas no valor da série temporal, $s(t)$ representa as alterações periódicas, $h(t)$ representa os efeitos dos feriados e o termo de erro ϵ_t engloba quaisquer mudanças idiossincráticas que não são capturadas pelo modelo.

No Prophet, a tarefa de previsão é concebida como um exercício de ajuste de curva, distinguindo-se dos modelos de séries temporais, que consideram explicitamente a estrutura de dependência temporal nos dados (Taylor; Letham, 2018).

Em resumo, o Prophet é um modelo de regressão simples e modular, geralmente eficaz com parâmetros padrão. Ele oferece a capacidade de medir e monitorar a precisão das previsões, permitindo a identificação de melhorias incrementais. A combinação de modelos simples e ajustáveis, juntamente com um monitoramento de desempenho escalável, possibilita que um grande número de analistas realize previsões para diversas séries temporais. Por essa razão, é considerado um modelo de previsão em larga escala (Taylor; Letham, 2018).

3. TRABALHOS RELACIONADOS

A robustez representa um desafio significativo na previsão de demanda de longo prazo, conferindo complexidade substancial a essa tarefa, como destacado por Zhou et al. (2022). A detecção precisa da sazonalidade por parte do modelo e esta pode ser comprometida devido a padrões presentes em dados ruidosos, sendo esta a razão subjacente à complexidade inerente à obtenção de robustez nesse contexto, conforme elucidado por Wu et al. (2021).

Um projeto de pesquisa utilizou dados de vendas do Walmart para realizar uma análise comparativa de diversos modelos de previsão para os próximos 28 dias. O estudo investigou modelos tanto paramétricos quanto não paramétricos, incluindo, ARIMA (Média Móvel Integrada Autoregressiva), LightGBM e Prophet. A análise e comparação de desempenho revelaram que o ARIMA se destacou, apresentando o menor Erro Quadrático Médio (RMSE) registrado, fixado em 1,09 (HASAN et al., 2022).

Foi conduzido um estudo com o intuito de comparar o desempenho de diversos modelos em quatro diferentes horizontes de tempo: o dia subsequente, sete dias à frente, trinta dias à frente e noventa dias à frente. Os modelos tradicionais de previsão de séries temporais, como Naïve, HoltWinters, TBATS e SARIMA, foram avaliados em conjunto com modelos estabelecidos de aprendizado de máquina e aprendizado profundo, tais como MLP (do inglês, *Multilayer Perceptron*), LSTM (do inglês, *Long-Short Term Memory*), XGBoost e Prophet. Entre todos os modelos considerados, o Prophet demonstrou o menor erro no horizonte temporal mais distante, apresentando um Erro Quadrático Médio (RMSE) de 17,2 (Lopes, 2022).

Em uma pesquisa destinada a avaliar a confiabilidade da análise preditiva, o algoritmo XGBoost foi aplicado para gerar previsões diárias de volume de vendas ao longo de uma semana. O Erro Absoluto Médio (MAE) foi equivalente a 2,44 (Baržic et. al., 2022).

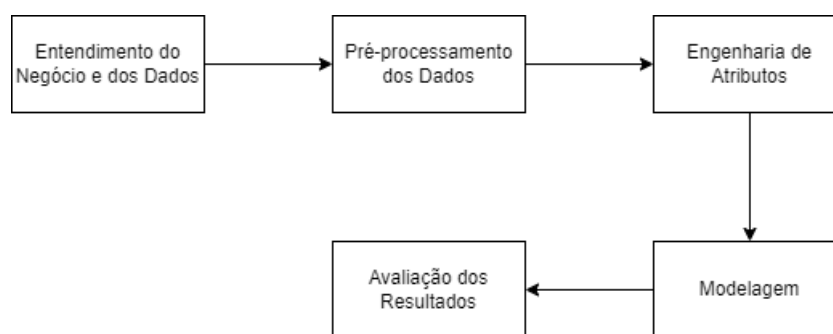
O ponto central de divergência entre a presente pesquisa e estudos anteriores reside no horizonte temporal de predição, sendo que o presente estudo apresenta uma previsão temporal mais longa que as pesquisas mencionadas.

4. METODOLOGIA

A abordagem metodológica adotada neste estudo fundamentou-se na Metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining*), que se traduz como Processo Padrão entre Indústrias para Mineração de Dados. Amplamente empregada em projetos de Ciência de Dados, essa metodologia é caracterizada por sua independência em relação a softwares ou técnicas específicas (Kelleher; Tierney, 2018).

Originalmente, a estrutura da metodologia CRISP-DM compreende seis fases distintas (entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação) (Kelleher; Tierney, 2018). Entretanto, adaptações foram realizadas para atender às exigências específicas deste trabalho. A Figura 1 representa visualmente a metodologia empregada, com as devidas modificações realizadas.

Figura 1. Metodologia geral aplicada baseada no CRISP-DM.



Fonte: Autores.

- **Entendimento do Negócio e dos Dados:** Na Seção 1, o problema a ser solucionado foi estabelecido em conformidade com o contexto empresarial. Subsequentemente, na Seção 3.1, os dados disponíveis para conduzir a previsão de demanda foram apresentados.
- **Pré-processamento dos Dados:** Nesta fase, efetuou-se a preparação completa dos dados, conforme detalhado na Seção 4.4.
- **Engenharia de Atributos:** A Seção 4.5 detalha esta etapa.
- **Modelagem:** Nesta fase, descrita na Seção 4.6, realiza-se a modelagem utilizando o Prophet.
- **Avaliação dos Resultados:** Por fim, conduz-se uma análise para observar os resultados, conforme a Seção 5.

As Seções 4.2 e 4.3 apresentam detalhes mais específicos acerca das ferramentas e do equipamento utilizados na realização deste estudo, englobando elementos como hardware, linguagens de programação, bibliotecas e ambiente em nuvem.

4.1. BASE DE DADOS

A base de dados empregada nesta pesquisa foi fornecida pela Empresa X no mercado de CFT, abrangendo 22 variáveis e 93,03M de instâncias. Os dados correspondem às vendas no período compreendido entre os anos de 2018 e 2023. Os dados de 2018 foram exclusivamente utilizados na criação de características de *lag* (atraso), enquanto os dados de 2023 foram reservados para a fase de teste. As variáveis brutas disponíveis estão detalhadas na Tabela 1.

Tabela 1. Variáveis disponíveis nos dados originais.

Variável	Descrição
Data de Venda	Data em que o produto foi vendido.
Ciclo	Período do ano, sendo que cada ano é composto por 17 ciclos.
Descrição do Ciclo	Definição do tipo de ciclo.
Data do início do ciclo	Data em que o ciclo foi iniciado.
Data do fim do ciclo	Data em que o ciclo termina.
Canal	Canal de vendas, podendo ser Venda Direta ou Loja.
UF	Unidade Federativa do Brasil.
Regional	Regional de campanha.
Categoria	Categoria do produto.
Subcategoria	Subcategoria do produto.
Marca	Marca do produto.
Código Produto	Código do Produto/SKU.
Descrição Produto	Descrição do Produto/SKU.
Volume	Variável Alvo. Quantidade de produtos vendidos.
Valor Praticado	Valor de venda com descontos.
Valor Tabela	Valor de venda sem descontos.
Valor de Desconto	Valor do desconto.
Porcentagem de Desconto	Porcentagem do desconto.

Preço Unitário	Preço unitário considerando os descontos.
Promoção	Indica se o produto está promocionado ou não.
Foco	Indica se o produto está em foco de campanha ou não.
Subfoco	Indica se o produto está em subfoco de campanha ou não.

Fonte: Sousa, A. et al. (2023)

4.2. HARDWARE

Para realizar os experimentos, o ambiente escolhido foi o *JupyterLab* na plataforma *workbench* do GCP (*Google Cloud Platform*). O hardware utilizado possui 16 vCPUs e 60 GB de RAM.

4.3. TECNOLOGIAS UTILIZADAS

A linguagem de programação empregada foi o Python 3.7 para pré-processamento dos dados, análises e construção do modelo, em conjunto com a linguagem de consulta estruturada SQL para importação, exportação e também para pré-processamento dos dados. No ambiente Python, foram utilizadas as seguintes bibliotecas:

- **pandas 1.3.5**, para manipulação dos dados por meio de *dataframes*;
- **numpy 1.21.6**, para execução de operações matemáticas e utilização de *arrays*;
- **matplotlib.pyplot 3.5.3**, para visualização dos dados por meio de gráficos;
- **google.cloud 3.3.3**, para conexão dos dados com o Google BigQuery;
- **sklearn 1.0.2**, para aplicação geral de técnicas de Aprendizado de Máquina;
- **prophet 1.1.1**, para modelagem por meio do método homônimo;
- **statsmodel 0.13.5**, para análise de séries temporais.

4.4. PRÉ-PROCESSAMENTO DOS DADOS

Nesta fase, os dados foram inicialmente formatados conforme as características individuais de cada variável. Após a formatação, procedeu-se à eliminação de dados nulos, resultando em uma redução de 0,22% na base de dados. Adicionalmente, realizou-se uma depuração para excluir produtos, categorias, subcategorias e marcas desativadas, considerando que não é pertinente prever a demanda de produtos que não integram mais o portfólio da empresa. Essa depuração resultou na remoção de 2,12% da base de dados.

Em seguida, os dados foram agrupados no nível temporal de semana, no nível de produto categoria/subcategoria/marca, no nível espacial de regional (sendo que são, no total, 6 regionais) e no nível de canal de vendas. Após o agrupamento, a base de dados apresentou aproximadamente 7,34M de instâncias.

4.5. ENGENHARIA DE ATRIBUTOS

Na fase de Engenharia de Atributos, diversas variáveis foram desenvolvidas para servirem como entrada no processo de modelagem preditiva. A Tabela 2 oferece uma descrição detalhada das variáveis criadas, levando em consideração os produtos e suas características, assim como promoções e campanhas.

Tabela 2. Atributos criados considerando produtos, campanhas e promoções.

Variáveis de origem	Variáveis criadas	Descrição
Produto, Ciclo, Categoria, Subcategoria, Marca	Quantidade de produtos no portfólio (1 variável)	Quantidade de produtos naquela categoria/subcategoria/marca.
Promoção, Produto, Ciclo, Categoria, Subcategoria, Marca, Estado	Produtos em promoção (1 variável)	Quantidade de produtos em promoção dentro da granularidade.
Porcentagem de Desconto, Ciclo, Produto, Categoria, Subcategoria, Marca, Estado	Quantidade de produtos sem desconto (1 variável criada)	Quantidade de produtos que estão fora de promoção.
Porcentagem de Desconto, Ciclo, Produto, Categoria, Subcategoria, Marca, Estado	Faixa de desconto (3 variáveis criadas)	Quantidade de produtos que pertencem a cada faixa de desconto (10%, 20%, 30%).
Foco, Ciclo, Produto, Categoria, Subcategoria, Marca, Estado	Produtos em foco (1 variável)	Quantidade de produtos em foco dentro da granularidade.
Subfoco, Ciclo, Produto, Categoria, Subcategoria, Marca, Estado	Produtos em subfoco (1 variável)	Quantidade de produtos em subfoco dentro da granularidade.

Fonte: Sousa, A. et al. (2023)

Na Tabela 3, foram mostradas as variáveis que calculam a fatia de mercado (do inglês, *market share*). O cálculo foi feito considerando as 52 semanas anteriores (duração de um ano) mais 30 semanas.

Tabela 3. Atributos criados calculando o *market share*.

Variáveis de origem	Variáveis criadas	Descrição
Categoria, Volume	<i>Market share</i> de volume da categoria (1 variável)	Volume de vendas de uma categoria em relação ao volume total no período do ano imediatamente anterior.
Categoria, Valor Tabela	<i>Market share</i> do valor tabela da categoria (1 variável)	Valor tabela de uma categoria em relação ao valor tabela total no período do ano imediatamente anterior.
Subcategoria, Volume	<i>Market share</i> de volume da subcategoria (1 variável)	Volume de vendas de uma subcategoria em relação ao volume total no período do ano imediatamente anterior.
Subcategoria, Valor Tabela	<i>Market share</i> do valor tabela da subcategoria (1 variável)	Valor tabela de uma subcategoria em relação ao valor tabela total no período do ano imediatamente anterior.

Marca, Volume	<i>Market share</i> de volume da marca (1 variável)	Volume de vendas de uma marca em relação ao volume total no período do ano imediatamente anterior.
Marca, Valor Tabela	<i>Market share</i> do valor tabela da marca (1 variável)	Valor tabela de uma marca em relação ao valor tabela total no período do ano imediatamente anterior.
Regional, Volume	<i>Market share</i> de volume da regional (1 variável)	Volume de vendas de uma regional em relação ao volume total no período do ano imediatamente anterior.
Regional, Valor Tabela	<i>Market share</i> do valor tabela da regional (1 variável)	Valor tabela de uma regional em relação ao valor tabela total no período do ano imediatamente anterior.

Fonte: Autores.

A Tabela 4 contém 21 variáveis de *lag* que foram criadas. As *lags* são as variáveis que permitem que o algoritmo de previsão considere a variável tempo, ou seja, elas extraem propriedades que podem ordenar os dados (Bergmeir; Benítez, 2012).

Na Tabela 5, são apresentadas as variáveis de ciclo que foram incorporadas ao método de séries temporais. Outro atributo vinculado às informações do ciclo, utilizado como entrada para o modelo, é a descrição do ciclo; contudo, essa variável não está listada na Tabela 5, uma vez que se trata de um elemento original dos dados.

Tabela 4. Atributos de *lag* construídos.

Variáveis de origem	Variáveis criadas	Descrição
Categoria, Volume	<i>Market share</i> de volume da categoria (1 variável)	Volume de vendas de uma categoria em relação ao volume total no período do ano imediatamente anterior.
Categoria, Valor Tabela	<i>Market share</i> do valor tabela da categoria (1 variável)	Valor tabela de uma categoria em relação ao valor tabela total no período do ano imediatamente anterior.
Subcategoria, Volume	<i>Market share</i> de volume da subcategoria (1 variável)	Volume de vendas de uma subcategoria em relação ao volume total no período do ano imediatamente anterior.
Subcategoria, Valor Tabela	<i>Market share</i> do valor tabela da subcategoria (1 variável)	Valor tabela de uma subcategoria em relação ao valor tabela total no período do ano imediatamente anterior.
Marca, Volume	<i>Market share</i> de volume da marca (1 variável)	Volume de vendas de uma marca em relação ao volume total no período do ano imediatamente anterior.
Marca, Valor Tabela	<i>Market share</i> do valor tabela da marca (1 variável)	Valor tabela de uma marca em relação ao valor tabela total no período do ano imediatamente anterior.
Regional, Volume	<i>Market share</i> de volume da regional (1 variável)	Volume de vendas de uma regional em relação ao volume total no período do ano imediatamente anterior.

Regional, Valor Tabela	<i>Market share</i> do valor tabela da regional (1 variável)	Valor tabela de uma regional em relação ao valor tabela total no período do ano imediatamente anterior.
---------------------------	--	---

Fonte: Autores.

Tabela 5. Atributos construídos com as características dos ciclos.

Variáveis de origem	Variáveis criadas	Descrição
Data de início de ciclo, Data de fim de ciclo	Duração do ciclo (1 variável)	Duração do ciclo daquela semana em dias, comumente 21 dias.
Ciclo	Ano do ciclo, Número do ciclo (2 variáveis)	Informação sobre o ciclo daquela semana.

Fonte: Autores.

Também foram gerados atributos automáticos por meio do método *PolynomialFeatures* da biblioteca *scikit-learn*, com o parâmetro *degree* estabelecido como 2. Algumas dessas variáveis são destacadas na Tabela 6. As variáveis de origem utilizadas para a criação desses atributos automáticos compreenderam: quantidade de produtos no portfólio, quantidade de produtos em promoção, quantidade de produtos sem desconto, quantidade de produtos com 10% de desconto, quantidade de produtos com 20% de desconto, quantidade de produtos com 30% de desconto, quantidade de produtos em destaque e quantidade de produtos em subfoco.

Tabela 6. Atributos automáticos construídos utilizando a biblioteca *scikit-learn*.

Variáveis de origem	Variáveis criadas	Descrição
Quantidade de produtos em portfólio	Quadrado de produtos em portfólio (1 variável)	Quantidade de produtos do portfólio ao quadrado.
Quantidade de produtos em promoção	Quadrado de produtos em promoção (1 variável)	Quantidade de produtos em promoção ao quadrado.
Quantidade de produtos sem desconto	Quadrado de produtos fora da promoção (1 variável)	Quantidade de produtos fora da promoção ao quadrado.
Quantidade de produtos nas faixas de 10%, 20%, 30% de desconto	Quadrado de produtos na faixa de 10%, 20%, 30% de desconto (3 variáveis)	Quantidade de produtos na faixa de 10%, 20%, 30% de desconto ao quadrado.
Quantidade de produtos em foco	Quadrado de produtos em foco (1 variável)	Quantidade de produtos em foco ao quadrado.
Quantidade de produtos em subfoco	Quadrado de produtos em subfoco (1 variável)	Quantidade de produtos em subfoco ao quadrado.

Fonte: Autores.

Adicionalmente às variáveis apresentadas na Tabela 6, foram geradas mais 28 variáveis automáticas, utilizando as mesmas características e o parâmetro previamente mencionado do método *PolynomialFeatures*. Essas 28 variáveis consistem nos produtos de cada uma dessas variáveis em pares, como exemplificado pela multiplicação da quantidade de produtos no portfólio pela quantidade de produtos em destaque. Dessa forma, o total resultante foi de 36 atributos automáticos construídos.

4.6. MODELAGEM

Foi implementado um modelo para cada série temporal, englobando cada categoria, subcategoria, marca, região e canal, totalizando 1.556 modelos. Os atributos de entrada são:

- 7 variáveis da granularidade da previsão (semana, ciclo, regional, canal, categoria, subcategoria e marca);
- 1 variável original contendo a descrição do ciclo;
- 8 variáveis criadas considerando produtos, campanhas e promoções (Tabela 2);
- 8 variáveis de *market share* construídas (Tabela 3);
- 13 variáveis de *lag* (Tabela 4);
- 3 variáveis construídas considerando informações de ciclo (Tabela 5);
- 6 variáveis criadas de maneira automática (Tabela 6).

Foi feito um treinamento para cada modelo com os dados de 2019 até 2022, utilizando os hiperparâmetros conforme configuração padrão da biblioteca do Prophet. Os dados do ano de 2023 foram utilizados como teste. O conjunto de teste tem o tamanho de trinta semanas e a performance do modelo foi avaliada utilizando esses resultados.

A métrica utilizada para a avaliação dos resultados foi o erro percentual absoluto ponderado (do inglês, *Weighted Mean Absolute Percentage Error* - WAPE). Devido à ponderação, os erros de previsão em produtos com maior volume de vendas seriam superiores aos erros em produtos com menor volume de vendas. O WAPE pode variar de 0% a qualquer número positivo em porcentagem, sendo desejável um valor menor para uma melhor precisão. O cálculo do WAPE é mostrado na equação abaixo:

$$WAPE = \frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|}$$

onde A_t é o valor observado no instante t , F_t é o valor predito no instante t e n é o instante máximo de tempo.

5. RESULTADOS E DISCUSSÃO

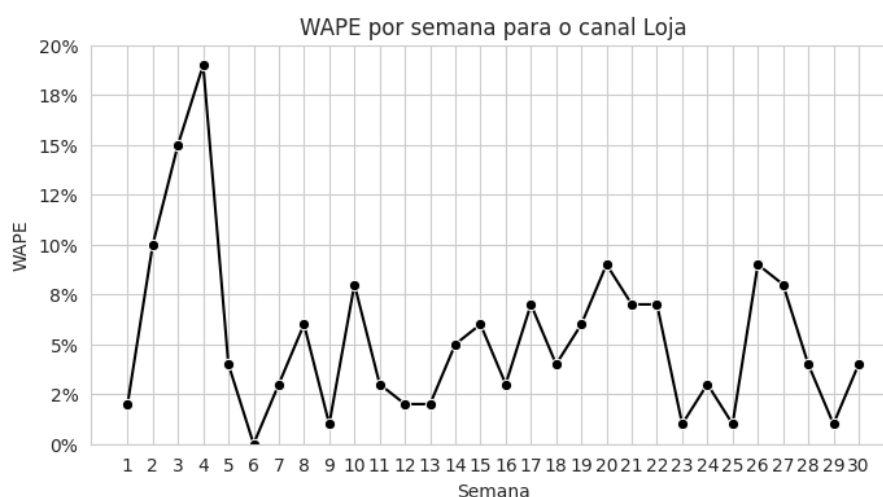
Nesta etapa, os resultados serão avaliados utilizando a métrica de erro WAPE, mencionada anteriormente na Seção 3.6. O volume realizado e o volume predito não serão informados por motivos de confidencialidade. Como a previsão do Prophet foi de até 30 semanas, os primeiros

resultados a serem analisados é o quanto o erro pode variar ao longo dessas semanas para os canais Loja e Venda Direta.

Em seguida, serão avaliados os erros na semana 30 (longo prazo) para as 3 categorias, subcategorias e marcas mais vendidas, para cada um dos dois canais de venda. Novamente, por questão de confidencialidade, os nomes reais das categorias, subcategorias e marcas não serão identificados. Sendo assim, as categorias serão nomeadas como C1, C2 e C3, as subcategorias como S1, S2 e S3 e as marcas como M1, M2 e M3.

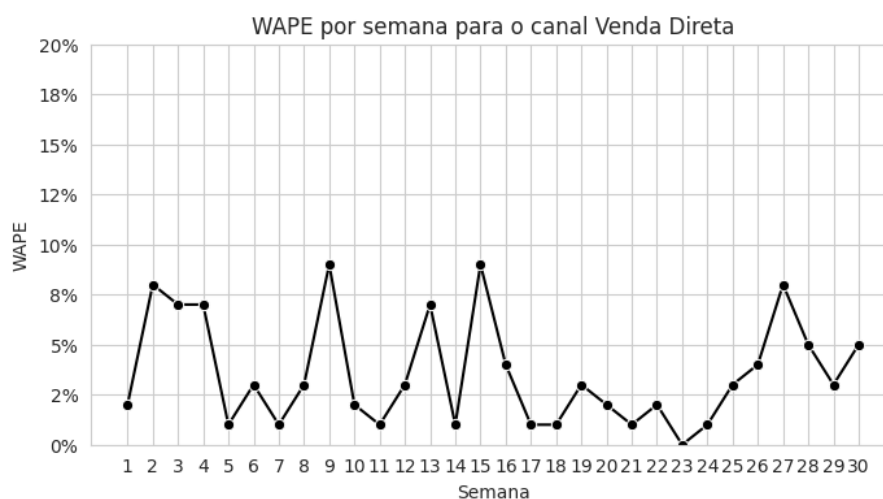
As Figuras 2 e 3 ilustram o WAPE ao longo das 30 semanas de previsão para o canal Loja e para o canal Venda Direta, respectivamente. No canal Loja (Figura 2), os erros em todas as semanas estão abaixo de 10%, com exceção das semanas 2, 3 e 4 que apresentaram os erros de 10%, 15% e 19%, respectivamente. Para o canal Venda Direta, todos os erros, sem exceção, estão abaixo dos 10%. Isso mostra que os resultados para o canal Venda Direta são melhores.

Figura 2. Erro de previsão no horizonte de 30 semanas para o canal Loja.



Fonte: Autores.

Figura 3. Erro de previsão no horizonte de 30 semanas para o canal Venda Direta.



Fonte: Autores.

A Tabela 7 exibe os erros para as 3 categorias mais vendidas na trigésima semana. É possível notar que todos os erros estão abaixo de 10% e a categoria que apresentou os menores erros para ambos os canais foi a C2.

A Tabela 8 mostra os erros para as 3 subcategorias mais vendidas na última semana de predição. Diferente do que foi observado na Tabela 7, quando o recorte de subcategoria é observado, a subcategoria S3 apresenta erros entre 10% e 12%. A segunda subcategoria mais vendida (S2) apresentou o menor WAPE, tanto em Loja, quanto em Venda Direta, assim como aconteceu com a categoria C2.

Tabela 7. Erro de predição para as 3 categorias mais vendidas nos dois canais de venda na semana 30.

Canal	Categoria	WAPE
Loja	C1	9,36%
Venda Direta	C1	5,20%
Loja	C2	2,19%
Venda Direta	C2	1,43%
Loja	C3	6,81%
Venda Direta	C3	9,13%

Fonte: Autores.

Tabela 8. Erro de predição para as 3 subcategorias mais vendidas nos dois canais de venda na semana 30.

Canal	Categoria	WAPE
Loja	S1	7,89%
Venda Direta	S1	2,65%
Loja	S2	0,70%
Venda Direta	S2	1,82%
Loja	S3	11,71%
Venda Direta	S3	10,35%

Fonte: Autores.

A Tabela 9 ilustra o WAPE para as 3 marcas com maior volume de vendas no horizonte de predição mais distante. Nessa visão, o Prophet apresentou menor estabilidade do que nos recortes de categoria e subcategoria, com todos os erros entre 5% e 16%.

Tabela 9. Erro de predição para as 3 marcas mais vendidas nos dois canais de venda.

Canal	Categoria	WAPE
Loja	M1	11,60%
Venda Direta	M1	10,00%
Loja	M2	5,03%
Venda Direta	M2	15,13%
Loja	M3	7,46%
Venda Direta	M3	9,90%

Fonte: Autores.

De maneira geral, pode-se dizer que o modelo apresenta previsões satisfatórias no longo prazo (30 semanas, aproximadamente 7 meses). Na visão semana a semana, um pequeno pico no canal Loja foi identificado nas semanas 3 e 4, mas a semana 30 apresentou um erro igual a 4%. No canal Venda Direta, o WAPE na última semana de predição foi equivalente a 5%. Ao analisar os erros utilizando as características dos produtos, apesar de ter sido identificado alguns erros acima de 10%, o maior erro encontrado foi de 15,13% para a marca M2 no canal Venda Direta.

6. CONSIDERAÇÕES FINAIS

Na área da Gestão da Cadeia de Abastecimento, que está associada à Engenharia de Produção, um desafio frequente é realizar previsões de demanda com base em dados históricos. Essa tarefa não é trivial e, especialmente em previsão de demanda de longo prazo, é ainda mais complexo devido à robustez (Zhou et al., 2022). Este estudo teve como objetivo fazer a previsão de demanda utilizando dados reais de uma empresa parceira em um período de 30 semanas à frente utilizando o algoritmo Prophet. Após a implementação da metodologia em dois canais de venda distintos, pôde-se observar que os resultados foram satisfatórios. O WAPE do canal Loja foi de apenas 5% e, no canal Venda Direta, foi ainda menor: 4%. Em futuras pesquisas, melhorias podem ser obtidas ao aplicar o estudo apresentado em conjuntos de dados distintos, proporcionando uma validação adicional. Além disso, a realização de experimentos com outros algoritmos, incluindo técnicas de aprendizagem profunda, pode enriquecer as análises e a compreensão do problema abordado.

6. REFERÊNCIAS

Bartkus, V., Mannor, M., & Campbell, J. (2022). Fast and rigorous: Configurational determinants of strategic decision-making balance. *Long Range Planning*, 55(3), 102142.

Baržic, M., Munitic, N.F., Bronic, F., Jelic, L., & Lešic, V. (2022). Forecasting Sales in Retail with XGBoost and Iterated Multi-step Ahead Method. In *2022 International Conference on Smart Systems and Technologies (SST)* (pp. 153–158).

- Bergmeir, C., & Benítez, J. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192–213.
- Brownlee, J. (2016). *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*. Machine Learning Mastery.
- Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1), 1–34.
- Faceli, K. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC.
- Hasan, M., Kabir, M., Shuvro, R., & Das, P. (2022). A Comparative Study on Forecasting of Retail Sales. *arXiv preprint arXiv:2203.06848*.
- Kelleher, J., & Tierney, B. (2018). *Data Science*. MIT Press.
- Lopes, G. (2022). The wisdom of crowds in forecasting at high-frequency for multiple time horizons: A case study of the Brazilian retail sales. *Brazilian Review of Finance*, 20(2), 77–115.
- Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.
- Muller, A., & Guido, S. (2018). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated.
- Nielsen, A. (2019). *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly Media.
- Sousa, A., Silva, C., & J.S.F. D. et al. (2021). *Análise de séries temporais*. SAGAH.
- Norvig, P., & Russell, S. (2013). *Inteligência Artificial*. ELSEVIER EDITORA.
- Ozdemir, S. (2016). *Principles of Data Science*. Packt Publishing.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business*. O'Reilly.
- Saltz, J., & Stanton, J. (2017). *An Introduction to Data Science*. SAGE Publications.
- Santos, M. (2021). *Introdução à inteligência artificial*. Editora Saraiva.
- Schoenherr, T., & Speier-Pero, C. (2015). Data science, predictive analytics, and big data in supply chain management: Current state and future potential. *Journal of Business Logistics*, 36(1), 120–132.
- Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, 7(1), 1–22.
- Sousa, A., Rego, T., Barbosa, Y., Menezes, T. (2023). Comparing Gradient Boosting Algorithms to Forecast Sales in Retail. In *Anais do XX Encontro Nacional de Inteligência Artificial e Computacional* (pp. 596–609).
- Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 22419–22430.
- Zhou, T., Zhu, J., Wang, X., Ma, Z., Wen, Q., Sun, L., & Jin, R. (2022). TreeDRNet: A Robust Deep Model for Long Term Time Series Forecasting. *arXiv preprint arXiv:2206.12106*.