



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Dissertação de Mestrado

Um modelo híbrido para séries temporais hierárquicas com múltipla sazonalidade

por

Gustavo Martins Venancio Pires

Brasília, 13 de Março de 2023

Um modelo híbrido para séries temporais hierárquicas com múltipla sazonalidade

por

Gustavo Martins Venancio Pires

Dissertação apresentada ao Departamento de Estatística da Universidade de Brasília, como requisito parcial para obtenção do título de Mestre em Estatística.

Orientador: Prof. Dr. José Augusto Fiorucci

Brasília, 13 de Março de 2023

Dissertação submetida ao Programa de Pós-Graduação em Estatística do Departamento de Estatística da Universidade de Brasília como parte dos requisitos para a obtenção do título de Mestre em Estatística.

Texto aprovado por:

Prof. Dr. José Augusto Fiorucci
Orientador, EST/UnB

Prof. Dr. Diego Carvalho do Nascimento
Membro externo, UDA

Prof. Dr. Paulo Henrique Ferreira
Membro externo, UFBA

Prof. Dr. Eduardo Yoshio Nakano
Suplente, EST/UnB

Estatísticos, assim como os artistas, têm o mau hábito de se apaixonar por seus modelos.

(George E. P. Box)

Para todos que me apoiaram ao longo desta caminhada.

Agradecimentos

Agradeço aos meus pais por me incentivarem a ingressar no mestrado em Estatística, à minha namorada por me incentivar todos os dias e me tornar uma pessoa melhor, aos meus amigos e familiares por sempre estarem ao meu lado nos momentos difíceis.

Um agradecimento especial ao meu orientador José Augusto Fiorucci por todo acompanhamento durante o projeto, pela disponibilidade em me atender sempre que possível e por todas as conversas que tivemos durante esse período.

Agradeço aos meus professores e colegas que me acompanharam e fizeram parte da construção do meu conhecimento durante essa jornada.

Agradeço à Universidade de Brasília, lugar onde conheci as pessoas mais brilhantes na minha vida e aos funcionários do Departamento de Estatística que sempre executaram um trabalho impecável.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Esta dissertação propõe um modelo híbrido capaz de realizar previsões de séries temporais hierárquicas com múltiplas sazonalidades. Essa metodologia híbrida consiste em utilizar um modelo de *Machine Learning* que possua variáveis contendo metodologias estatísticas de séries temporais para gerar previsões coesas. Essa metodologia foi aplicada no banco de dados da competição *M5 - Forecasting (2020)* disponibilizada pelo *Kaggle*, em que o objetivo era prever com maior acurácia a venda diária de 3.409 produtos distribuídos em 5 níveis de hierarquia por 28 dias. Durante o trabalho foram comparadas 5 abordagens diferentes e o modelo de *Light Gradient Boosting Machine* (LGBM) contendo uma variável baseada na metodologia estatística TBATS (Trigonometric seasonality, Box-Cox transformation ARMA errors, Trend and Seasonal components) chegou obter um ganho de acurácia de 27% em comparação com os modelos de LGBM sem a variável em questão. Esse modelo teria obtido a 318ª colocação na competição, ficando entre os top 6% competidores.

Palavras-chave: Séries temporais hierárquicas, Sazonalidade múltipla, Sazonalidade complexa, LGBM, TBATS, Big Data Time Series

Abstract

This Master's Thesis proposes a hybrid model capable of forecasting hierarchical time series with multiple seasonality. This hybrid methodology consists of using a *Machine Learning* model that has variables containing time series statistical methodologies to generate cohesive forecasts. This methodology was applied to the *M5 - Forecasting (2020)* competition available through *Kaggle*, in which the objective was to more accurately predict the daily sale of 3,409 products distributed in 5 levels of hierarchy by 28 days. During the dissertation, 5 different approaches were compared and the *Light Gradient Boosting Machine* (LGBM) model containing a variable based on the TBATS (Trigonometric seasonity, Box-Cox transformation ARMA errors, Trend and Seasonal components) obtained an accuracy gain of 27% compared to the LGBM models without the variable mentioned. This model would have obtained the 318th place in the competition, being among the top 6% competitors.

Keywords: Hierarchical Time Series, Multiple Seasonality, Complex Seasonality, LGBM, TBATS, Big Data Time Series

Sumário

1	Introdução	1
2	Revisão bibliográfica	3
2.1	Trabalhos Correlatos	3
2.2	Séries Temporais	6
2.3	Modelos de séries temporais	7
2.3.1	Modelos auto-regressivos	7
2.3.2	Modelos de médias móveis	8
2.3.3	Modelos auto-regressivos e de médias móveis	9
2.4	TBATS	9
2.5	Séries Temporais Hierárquicas ou Agrupadas	14
2.5.1	Métodos de Resolução e Aplicação	16
2.6	Light Gradient Boosting Machine	18
2.7	Métricas de avaliação do modelo	21
2.7.1	Weighted Root Mean Squared Scaled Error (WRMSSE)	21
2.7.2	Root Mean Squared Error (RMSE)	22
2.7.3	Mean Absolute Error (MAE)	22
2.7.4	Mean Absolute Percentage Error (MAPE)	23
3	Banco de dados	24

3.1	Análise Exploratória	26
3.1.1	Vendas Totais	27
3.1.2	Vendas por Estado	29
3.1.3	Vendas por Loja	29
3.1.4	Vendas por Categoria e Estado	30
3.1.5	Vendas por Departamento e Estado	31
3.1.6	Constatações	32
4	Metodologia	33
4.1	Construção de variáveis estatísticas para Machine Learning	33
4.2	Aplicação do conceito de construção da variável estatística	36
5	Resultados	38
5.1	Modelo TBATS	39
5.2	Modelo LGBM sem a variável estatística composta pelo TBATS	42
5.3	Modelos LGBM com a variável estatística composta pelo TBATS	44
5.3.1	LGBM com primeiro ajuste do TBATS até o dia 750	45
5.3.2	LGBM com primeiro ajuste do TBATS até o dia 365	48
5.3.3	LGBM com primeiro ajuste do TBATS até o dia 1.500	51
5.4	Comparação de Resultados	52
6	Conclusão	55
	Referências Bibliográficas	57

Lista de Tabelas

3.1	Relações e descrições de variáveis utilizadas no treinamento do modelo	26
4.1	Exemplo de construção da variável h_1 em estrutura de dados hierarquizados . .	36
5.1	Medidas de erro do modelo TBATS conforme hierarquia	41
5.2	Medida WRMSSE do modelo TBATS conforme o intervalo	41
5.3	Medidas de erro do modelo LGBM conforme Hierarquia	43
5.4	Medida WRMSSE do modelo LGBM conforme o intervalo	44
5.5	Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 750 conforme hierarquia	46
5.6	Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 750 conforme o intervalo	47
5.7	Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 365 conforme Hierarquia	49
5.8	Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 365 conforme o intervalo	50
5.9	Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 1.500 conforme Hierarquia	51
5.10	Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 1.500 conforme o intervalo	52

5.11 Métricas do WRMSSE para todos os modelos treinados 53

Lista de Figuras

2.1	Série temporal da quantidade de mortes mensais por doenças de câncer no Reino Unido (Diggle, 1990)	6
2.2	Série temporal do Consumo de Energia em kWh pelos Estados Unidos de 1973 a 2010 (Hyndman et al., 2022)	10
2.3	Exemplo de Hierarquia de nível 3	15
2.4	Representação de uma estrutura agrupada	16
2.5	Algoritmos inclusos no processo EFB	21
3.1	Organograma do banco de dados	25
3.2	Série diária da quantidade de produtos vendidos	27
3.3	Decomposição da série do total de vendas	28
3.4	Séries de vendas diárias por Estado	29
3.5	Séries de vendas mensais por loja e Estado	30
3.6	Séries de vendas mensais por Categoria e Estado	31
3.7	Séries de vendas mensais por Departamento e Estado	32
4.1	Exemplo de construção de variável	35
4.2	Processo de construção da variável h_1	37
5.1	Análise residual do modelo TBATS	40
5.2	Curva de aprendizado do LGBM treinado sem a variável estatística	43

5.3	Curva de aprendizado do LGBM com primeiro ajuste do TBATS até o dia 750 .	46
5.4	Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 750	48
5.5	Curva de aprendizado do LGBM treinado com a variável estatística até o dia 365	49
5.6	Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 365	50
5.7	Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 1.500	52
5.8	Comparativo entre previsões durante o período de validação da competição . .	54

Capítulo 1

Introdução

Séries Temporais hierárquicas representam valores coletados num espaço temporal que podem ser agregados ou desagregados em estruturas de hierarquia como: informações geográficas, características endógenas e outras opções associadas a ordenação. Esses tipos de série geralmente estão relacionados a problemas como: a previsão de demanda por energia numa cidade, em que também há o interesse em analisar suas previsões por região, bairro, rua, prédio e demais granularidades. Outra característica dessas séries está associada aos dados de alta-frequência que possuem um período longo para análise e muitas observações distribuídas pelo tempo. Como solução para séries desse tipo, modelos de *Machine Learning* vêm sendo propostos como uma alternativa aos modelos estatísticos, mas sua superioridade permanece em questão (Spiliotis et al., 2020).

Além de sua aplicabilidade em numerosas situações, muitas competições de previsão de séries temporais vêm adotando problemas de séries de alta frequência hierarquicamente estruturadas, o que contribuiu para o avanço no debate sobre o problema. Tem sido comum encontrar soluções vencedoras dessas competições que utilizem modelos como o eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGBM), Redes Neurais (NN) e misturas de técnicas de computação intensiva (Makridakis, Spiliotis e Assimakopoulos, 2020), devido ao grande volume de dados envolvido. A qualidade das variáveis compõe um papel crucial na

capacidade de obter uma acurácia satisfatória nesses modelos. Isso suplica a necessidade de realizar um tratamento adequado às variáveis que já estão presentes no banco de dados e incorporar informações relevantes ao problema. Atualmente existem pacotes, tanto em **R** quanto em **Python**, que extraem fatores importantes das séries como: **tsfeatures** (Hyndman et al., 2019) e **TSFEL** (Barandas et al., 2020), porém ambos não possuem a alternativa de adicionar elementos baseados em modelos estatísticos.

Para este trabalho propomos uma maneira de construir variáveis que utilize técnicas estatísticas de séries temporais em bancos de dados orientados ao treinamento de modelos de *Machine Learning*. Temos como objetivo avaliar a sua relevância e medir sua contribuição no poder preditivo do modelo treinado. Este trabalho foi motivado pela competição *M5 - Forecasting (2020)* disponibilizada pelo *Kaggle*, em que o objetivo era prever com maior acurácia a venda diária de 3.409 produtos distribuídos em 5 níveis de hierarquia por 28 dias com uma premiação de \$50.000.

A seguir será realizada uma revisão bibliográfica abordando trabalhos correlatos e conceitos básicos para melhor entendimento do trabalho. No capítulo 3 é feita uma explicação do banco de dados e da metodologia adotada para solução do problema seguido de uma análise exploratória. No capítulo seguinte são apresentados os resultados e, por fim, são feitos os comentários finais na Conclusão.

Capítulo 2

Revisão bibliográfica

2.1 Trabalhos Correlatos

Um grande desafio das séries temporais hierárquicas diz respeito às gerações de previsões coesas conforme a hierarquia. As abordagens mais conhecidas dentro da área são o *bottom-up* e *top-down*, apresentados posteriormente. Em Gross e Sohl, 1990 são discutidas as diretrizes sobre como selecionar um método e sob quais condições um determinado procedimento é aplicável. Foi visto que na prática muitas empresas combinam esses métodos (resultando no chamado *middle-out*) em que as previsões são obtidas para cada série em um nível intermediário da hierarquia, e então a agregação é usada para obter previsões em níveis mais altos e a desagregação é usada para obter previsões em níveis mais baixos.

Tanto pelo *bottom-up* quanto pelo *top-down* há incertezas sobre a perda de informação ao agregar previsões para níveis seguintes de estrutura, o que dificulta obter previsões para qualquer um desses métodos. Em Hyndman et al., 2011 foi proposta uma abordagem baseada em prever independentemente todas as séries em todos os níveis de hierarquia e utilizar um modelo de regressão para combinar todos os resultados de maneira otimizada. Foi visto que os resultados das previsões somaram-se adequadamente pelas hierarquias, não possuem viés e possuem variância mínima sob algumas suposições simples. Essa abordagem utiliza um estimador de

mínimos quadrados que requer uma estimativa da matriz de variância-covariância dos erros de previsão.

Wickramasuriya, Athanasopoulos, Hyndman et al., 2015 demonstram que essa matriz é incoerente e impossível de estimar na prática por problemas de identificabilidade e apresentam uma nova abordagem chamada *Minimum Trace* (MinT), que propõe realizar previsões minimizando o erro quadrático médio de previsões consistentes através de toda a série temporal sob a suposição de que não há viés. O problema de minimização possui uma forma fechada de resolução e os resultados apresentados sugeriram que há coerência, falta de viés e variância mínima entre todas combinações de previsão.

Aliado ao problema das agregações de previsões, também pode-se encontrar desafios quanto à sazonalidade presente nos dados. Geralmente, versões sazonais do ARIMA e os modelos ETS são designados para ciclos sazonais mais curtos, como 12 meses para dados mensais ou 4 trimestres para dados trimestrais (Hyndman e Athanasopoulos, 2018), e podem não se ajustar bem em casos que possuem mais de um tipo de sazonalidade. De Livera, Hyndman e Snyder, 2011 propõem uma maneira de estimar séries temporais de sazonalidades múltiplas por meio de um modelo de espaço de estados (TBATS) incorporando transformações de Box-Cox, representações de Fourier com coeficientes variáveis com o tempo e uma correção utilizando o processo ARMA que será abordado com maiores detalhes no capítulo de Metodologia.

Caso o banco de dados seja extenso e possua muitas hierarquias, o processo MinT se tornará custoso computacionalmente por ter que realizar previsões individuais para cada série. Dessa forma surge a necessidade de obter acurácia satisfatória para o modelo com menor custo computacional e pouco tempo de processamento. Uma alternativa para esse tipo de problema é adotar soluções de *Machine Learning* que possam acelerar este processo sem perda de eficácia. Phumchusri e Ungtrakul, 2020 encontraram, para um estudo de caso, que modelos de *Machine Learning* apresentam performance melhores que os modelos de séries temporais designados para sazonalidade complexa como o BATS e TBATS.

Um dos modelos que têm chamado atenção é o *Light Gradient Boosting Machine* (LGBM),

algoritmo desenvolvido por Ke et al., 2017, que é uma forma eficiente e distribuída de executar o convencional *Gradient Boosting Decision Tree* (GBDT) aumentando em mais de 20 vezes a velocidade de treinamento e alcançando praticamente a mesma acurácia que um processo convencional do GBDT. O LGBM pode ser aplicado para problemas de séries temporais, regressão, classificação binária e múltipla, entropia e LambdaRank. Ao comparar o LGBM com outros algoritmos do tipo *Boosting* como o *XGBoost* e *Catboost* em um banco de dados de larga escala, foi visto que o LGBM apresenta uma velocidade maior que qualquer outro método *Boosting* e maior acurácia utilizando a mesma restrição de tempo e otimização (Al Daoud, 2019).

Para modelos da classe GBDT há a necessidade de construir variáveis dentro do banco de dados que auxiliem a explicar a variável resposta. No contexto de séries temporais atualmente existem pacotes, como o **tsfeatures** (Hyndman et al., 2019) para o *software* R, que extrai informações como: entropia, parâmetros das decomposições STL, auto-correlações, heterogeneidade e demais outras; O **TSEFEL** (Barandas et al., 2020) que é um pacote em Python no qual computa mais de 60 tipos de variáveis as quais separam em domínios temporais (auto-correlação, média absoluta das diferenças, mediana entre as diferenças, entropia e demais outras), estatísticos (amplitude interquartílica, desvio médio absoluto, Variância, etc) e espectrais (média absoluta de *Wavelet*, variância de *Wavelet*, distância espectral e outros). Porém, nenhum deles aborda uma maneira de construção de variáveis que parte de metodologias de ajuste de modelos estatísticos.

As soluções não se limitam somente aos modelos usuais de *Machine Learning*, Mancuso, Piccialli e Sudoso, 2021 propõem uma abordagem para séries temporais hierárquicas utilizando Redes Neurais (NN), na qual é explorada a habilidade de uma rede extrair informações capturando a estrutura da hierarquia ao utilizar variáveis explicativas. Sua abordagem é validada para três bancos de dados reais (inclusive o utilizado para esta dissertação) em que obteve performance melhor que os modelos Naive Bayes, ARIMA e ETS.

2.2 Séries Temporais

Uma série temporal é um conjunto de observações obtidas sequencialmente ao longo do tempo em que a ordem dos dados é fundamental para análise. Sua classificação pode ser contínua, quando as observações são obtidas a qualquer intervalo de tempo, ou discreta, quando são extraídas em intervalos de tempo equidistantes. A principal característica das séries temporais é a existência de dependência entre observações vizinhas, o que limita a utilização de teorias que assumam dados independentes.

Podemos descrever as séries temporais como uma realização de um processo estocástico representando uma família de variáveis aleatórias no formato $Y = \{Y(t), t \in T\}$ em que denotamos uma série temporal como $\{Y_t\}$ em que cada observação Y representa o valor da variável aleatória em estudo no momento t . Os principais objetivos ao analisar uma série temporal passam por: investigar o mecanismo gerador da série temporal, realizar previsões de valores futuros da série, descrever o comportamento da série ou procurar periodicidades relevantes nos dados (Morettin e Tolo, 2006). A Figura 1 apresenta o gráfico com uma série da quantidade de mortes mensais por doenças de câncer no Reino Unido entre 1974 e 1979.

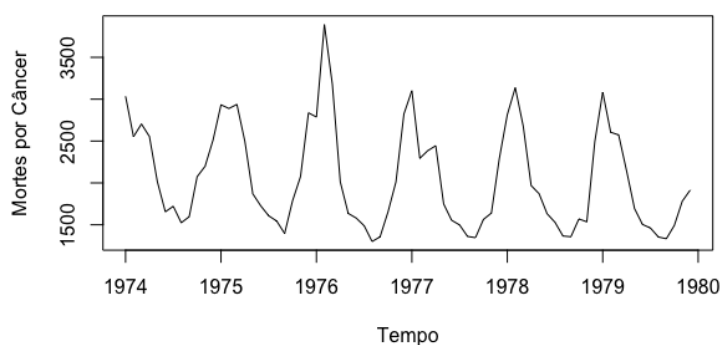


Figura 2.1: Série temporal da quantidade de mortes mensais por doenças de câncer no Reino Unido (Diggle, 1990)

Pela Figura 1 observamos um comportamento cíclico e que tende a se repetir em um deter-

minado período de tempo; Este fator é determinado como sazonalidade. Outro fator importante é relacionado ao comportamento principal da série, ou seja, se ela possui uma propensão a seguir um caminho de crescimento ou decréscimo. Neste caso, definimos esse fenômeno como tendência da série. Além dos dois fenômenos citados, também existe o fator aleatório dentro de cada processo, o qual denominamos como ruído. Com base nos três fenômenos descritos definimos um modelo de decomposição aditivo como:

$$Y_t = T_t + S_t + \alpha_t \quad (2.1)$$

Sendo os termos T_t , S_t e α_t os componentes de tendência, sazonalidade e ruído, respectivamente. Entre as formas de estimação de tendência, estão: i) ajustar um polinômio por mínimos quadrados, ii) suavização por médias móveis, medianas móveis ou lowess. Quanto à sazonalidade, existem diversas formas de estimar a sazonalidade sendo que as mais usuais são: Sazonalidade Determinística - método de regressão; e Sazonalidade Estocástica - método das médias móveis (Morettin e Tolo, 2006).

2.3 Modelos de séries temporais

Em séries temporais existem diversos modelos apropriados para cada característica de problema e comportamento de série. A seguir serão introduzidos os modelos da família ARMA que são os mais utilizados no quesito de divulgação e introdução a técnicas mais específicas.

2.3.1 Modelos auto-regressivos

Um modelo auto-regressivo de ordem p possui a série histórica Y_t descrita pelos parâmetros ϕ , seus valores passados e pelo ruído aleatório. Denota-se este modelo na ordem p como AR(p)

em (2) ou com o operador auto-regressivo estacionário em (2) e (3):

$$\tilde{Y}_t = \phi_1 \tilde{Y}_{t-1} + \phi_2 \tilde{Y}_{t-2} + \cdots + \phi_p \tilde{Y}_{t-p} + \alpha_t, \quad (2.2)$$

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \quad (2.3)$$

Pela Equação (3) podemos reescrever um processo AR(1) da seguinte forma:

$$\tilde{Y}_t = \phi \tilde{Y}_{t-1} + \alpha_t,$$

$$\tilde{Y}_t = \alpha_t + \phi \alpha_{t-1} + \phi^2 \alpha_{t-2} + \cdots = \sum_{j=0}^{\infty} \phi^j \alpha_{t-j},$$

$$\psi(B) = \sum_{j=0}^{\infty} \phi^j \alpha_{t-j} = (1 - \phi B)^{-1}$$

O processo será estacionário se $\psi(B)$ convergir para $|B| \leq 1$, ou seja, devemos ter as raízes de $|\phi| < 1$. Além dessa condição, define-se a estacionariedade se as autocovariâncias são independentes.

2.3.2 Modelos de médias móveis

Um processo é denominado como médias móveis de ordem q , denotado como MA(q) se sua equação é a seguinte:

$$Y_t = \mu + \alpha_t - \theta_1 \alpha_{t-1} - \cdots - \theta_q \alpha_{t-q} \quad (2.4)$$

Realizando a transformação $\tilde{Y}_t = Y_t - \mu$ temos:

$$\tilde{Y}_t = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q) \alpha_t = \theta(B) \alpha_t$$

Para o exemplo mais simples de médias móveis escrevemos MA(1):

$$\begin{aligned}\tilde{Y}_t &= \alpha_t - \theta\alpha_{t-1} \\ \tilde{Y}_t &= (1 - \theta B)\alpha_t\end{aligned}$$

De modo que $\theta(B) = 1 - \theta B$ e por ser sempre finito, o processo é sempre estacionário. O mesmo vale para qualquer ordem q do processo MA(q).

2.3.3 Modelos auto-regressivos e de médias móveis

Os modelos auto-regressivos e de médias móveis são construídos a partir da combinação da equação (2) dos modelos auto-regressivos e da equação (4) dos modelos de médias móveis. Sua notação também consiste da combinação de suas iniciais formando o modelo ARMA(p, q) e sua forma segue:

$$\tilde{Y}_t = \phi_1\tilde{Y}_{t-1} + \phi_2\tilde{Y}_{t-2} + \dots + \phi_p\tilde{Y}_{t-p} + \alpha_t - \theta_1\alpha_{t-1} - \dots - \theta_q\alpha_{t-q}$$

Aplicando os operadores auto-regressivos e de médias móveis, reescreve-se a equação acima da seguinte maneira:

$$\phi(B)\tilde{Y}_t = \theta(B)\alpha_t$$

2.4 TBATS

Para o exemplo da Figura 2.1, percebemos um padrão sazonal quadrimestral relativamente simples. Porém, para casos de séries temporais com uma frequência maior de observações é comum encontrar padrões sazonais mais complicados, como o caso da Figura 2.2, em que existe mais de um tipo de sazonalidade (mensal e anual). Para estes casos definimos a sazonalidade como múltipla e isso requer um ajuste na equação (1) tratando o termo da sazonalidade como

$S_t = \sum_{n=1}^T s_n$ para uma quantidade T de sazonalidades.

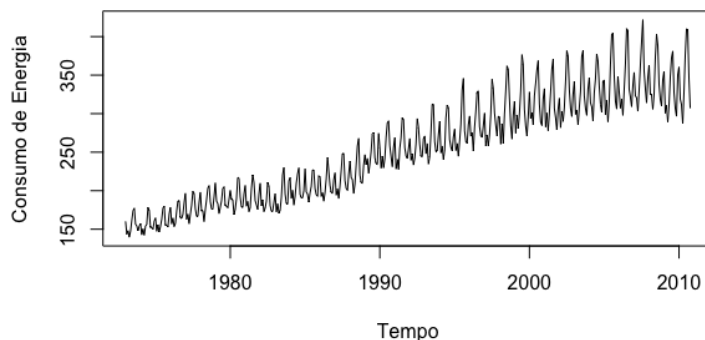


Figura 2.2: Série temporal do Consumo de Energia em kWh pelos Estados Unidos de 1973 a 2010 (Hyndman et al., 2022)

Esse padrão sazonal merece uma atenção especial pois podemos nos deparar com situações em que a sazonalidade passa a ter um ciclo não inteiro, como o ciclo anual da Figura 2.2, que representa $365,25/7 = 52,17$ semanas. A maioria dos métodos para modelagem de sazonalidade é incapaz de lidar com esse tipo de complexidade, sendo que a maioria assume um período inteiro dentro de seu modelo.

O TBATS (*Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components*) é um modelo desenvolvido por De Livera, Hyndman e Snyder, 2011, capaz de trabalhar com séries temporais de sazonalidade complexa, que utiliza uma combinação de transformações de Box-Cox, séries de Fourier e correções de resíduos por modelos ARMA. O modelo é construído a partir do BATS, proposto pelos mesmos autores, e segue a seguinte estrutura:

$$y_t = \text{série original no tempo } t$$

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^{(\omega)} - 1}{\omega}, & \omega \neq 0 \\ \log(y_t), & \omega = 0 \end{cases}$$

sendo $y_t^{(\omega)}$ a representação da transformação de Box-Cox sobre a série original com o parâmetro ω . Essa transformação tem como objetivo estabilizar a variância dentro da série e tornar os dados mais semelhantes à uma distribuição conhecida. Ao aplicar o modelo de Holt-Winters aditivo com Damped sobre a série transformada, temos a equação:

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t \quad (2.5)$$

sendo l_t e b_t os componentes de nível e tendência, respectivamente. $s_{t-m_i}^{(i)}$ é a i -ésima componente sazonal no tempo t , sendo m_1, \dots, m_t os respectivos períodos sazonais e d_t sendo o termo de erro que apresenta uma memória temporal sendo um processo ARMA(p, q) utilizado para modelar possíveis autocorrelações que geralmente sobram nos resíduos dos modelos de alinhamento exponencial. Cada termo da soma é representado por:

$$\begin{aligned} l_t &= l_{t-1} + \phi b_{t-1} + \alpha d_t \\ b_t &= \phi b_{t-1} + \beta d_t \\ s_t^{(i)} &= s_{t-m_i}^{(i)} + \gamma_i d_t \\ d_t &= \sum_{j=1}^p \varphi_j d_{t-j} + \sum_{k=1}^q \theta_k \epsilon_{t-k} + \epsilon_k \end{aligned}$$

O termo $s_{t-m_i}^{(i)}$ possui a limitação de trabalhar com períodos sazonais m_i completos, ou seja, caso a sazonalidade seja complexa seria necessário realizar uma aproximação em m_i para trabalhar com esse modelo. De Livera, Hyndman e Snyder, 2011 introduzem uma nova representação desse componente baseado nas decomposições em séries de Fourier. A ideia principal dos autores é decompor a sazonalidade em uma combinação de séries harmônicas, sendo cada componente sazonal escrita como a transformação de Fourier:

$$s_t = a_0 + a_1 \text{sen}\left(\frac{2\pi t}{m}\right) + a_2 \text{cos}\left(\frac{2\pi t}{m}\right) + a_3 \text{sen}\left(\frac{4\pi t}{m}\right) + a_4 \text{cos}\left(\frac{4\pi t}{m}\right) + \dots$$

Essa componente trigonométrica permite modelar séries com diversas sazonalidades, assim como séries com temporalidades de frequência não inteira. Então no modelo TBATS reescrevemos o componente sazonal da Equação (5) como uma série de Fourier para k_i sendo o número de séries harmônicas utilizadas para modelar cada ciclo sazonal. Já os outros dois termos são relacionados às equações harmônicas, mas escritas como um processo dinâmico:

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos(\lambda_j^{(i)}) + s_{j,t-1}^{*(i)} \text{sen}(\lambda_j^{(i)}) + \gamma_1^{(i)} d_t$$

$$s_{j,t}^{*(i)} = -s_{j,t-1} \text{sen}(\lambda_j^{(i)}) + s_{j,t-1}^{*(i)} \cos(\lambda_j^{(i)}) + \gamma_2^{(i)} d_t$$

Consideramos γ_1 e γ_2 como parâmetros de suavização e $\lambda_j^{(i)}$ sendo igual a $\frac{2\pi j}{m_i}$. O nível estocástico da i -ésima componente sazonal é descrito por $s_{j,t}^{(i)}$ e o "crescimento" sazonal no nível da i -ésima componente, que é necessária para descrever a mudança da componente sazonal pelo tempo, será $s_{j,t}^{*(i)}$.

Com isso, obtém-se o modelo final TBATS e a sua notação dada por $\text{TBATS}(\omega, \{p, q\}, \phi, \{m_1, k_1\}, \dots, \{m_T, k_T\})$, em que ω está relacionado a transformação de Box-Cox, os termos p, q fazem referência ao modelo $\text{ARMA}(p, q)$ utilizado para os erros, ϕ está relacionado ao parâmetro de Damped e as combinações m_i, k_i fazem referência a cada ciclo sazonal de tamanho m_i , demandando uma quantidade k_i de séries harmônicas.

A estimação dos parâmetros em modelos lineares de espaço pode ser resolvido por mínimos quadrados ou por verossimilhança gaussiana, mas para o contexto do modelo TBATS há a necessidade de estimar os parâmetros de transformação de Box-Cox (ω) e os coeficientes ARMA $\{p, q\}$. Os estados iniciais dos modelos de espaço de estado são geralmente tratados como vetores aleatórios. Dados os valores experimentais dos parâmetros desconhecidos, as distribuições conjuntas de estados estacionários são derivados e, em seguida, atribuídos aos estados iniciais associados.

A verossimilhança condicional da série observada $y = (y_1, y_2, \dots, y_n)$ é derivada da premissa que $\epsilon_t \sim N(0, \sigma^2)$. Isso implica que a densidade da série após a transformação de Fourier é $y_t^{(\omega)} \sim N(\omega x_{t-1}, \sigma^2)$ e é descrita da seguinte forma:

$$\begin{aligned} p(y_t^{(\omega)} | x_0, \vartheta, \sigma^2) &= \prod_{t=1}^n p(y_t^{(\omega)} | x_{t-1}, \vartheta, \sigma^2) \\ &= \prod_{t=1}^n p(\epsilon_t) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=1}^n \epsilon_t\right) \end{aligned}$$

Em que $\vartheta = (\omega, \phi, p, q)$ é o vetor contendo os parâmetros de Box-Cox, alisamento exponencial e os coeficientes ARMA. Sendo assim, a densidade da série original utilizando o jacobiano da transformação de Box-Cox é:

$$\begin{aligned} p(y_t | x_0, \vartheta, \sigma^2) &= p(y_t^{(\omega)} | x_0, \vartheta, \sigma^2) \left| \det \left(\frac{\partial y_t^{(\omega)}}{\partial y} \right) \right| \\ &= p(y_t^{(\omega)} | x_0, \vartheta, \sigma^2) \prod_{t=1}^n y_t^{\omega-1} \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=1}^n \epsilon_t^2\right) \prod_{t=1}^n y_t^{\omega-1} \end{aligned}$$

Assim a log-verossimilhança é obtida por:

$$\ell(x_0, \vartheta, \sigma^2) = \frac{-n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^n \epsilon_t^2 + (\omega - 1) \sum_{t=1}^n \log y_t \quad (2.6)$$

A estimativa de máxima verossimilhança da variância do erro é obtida igualando a derivada parcial em relação a σ^2 a zero, resultando em:

$$\hat{\sigma}^2 = n^{-1} \sum_{t=1}^n \epsilon_t^2 \quad (2.7)$$

Substituindo (7) em (6), multiplicando por -2 e omitindo os termos constantes temos:

$$\ell(x_0, \vartheta) = n \log \left(\sum_{t=1}^n \epsilon_t^2 \right) - 2(\omega - 1) \sum_{t=1}^n \log y_t \quad (2.8)$$

Para obter estimativas de verossimilhança deve-se minimizar (8) e resolver o problema de otimização numérica computacionalmente. Durante as estimativas realizadas neste trabalho foi utilizada a função *tbats* do pacote **forecast** (Hyndman et al., 2020) do *software* R.

Este trabalho focou em analisar um ramo das séries temporais conhecidas como hierárquicas ou agrupadas, as quais serão apresentadas a seguir em conjunto com suas metodologias mais conhecidas de resolução.

2.5 Séries Temporais Hierárquicas ou Agrupadas

As séries temporais podem ser desagregadas por diversos fatores de interesse. Como por exemplo, o número de chuteiras vendidas por uma marca específica pode ser desagregado por tipo de solado: borracha, meias travas e travas longas, sendo que cada tipo de solado ainda pode ter mais categorias específicas como: travas de alumínio, travas de borracha, travas mistas, entre outros. Essas categorias citadas ainda podem formar um novo tipo de grupo e assim a coleção segue uma agregação hierárquica.

A Figura 2.3 mostra uma estrutura hierárquica de nível 3, em que no topo da hierarquia (círculo roxo) é determinado o total da série e a parte mais agregada de todo o banco de dados. A notação para a observação t no total da série fica determinada por y_t para $t = 1, \dots, T$. Partindo para o próximo nível, o total é dividido em duas séries que são divididas novamente em três e duas novas séries de hierarquia mais baixa. Abaixo do nível mais alto, são empregadas notações do tipo $y_{j,t}$ correspondendo à observação t do nível j . Para o exemplo da Figura 2.3, $y_{B,t}$ corresponde a t -ésima observação do primeiro nível, assim como $y_{BB,t}$ denota a t -ésima observação da série correspondente ao segundo nível.

Para este exemplo são observados 8 tipos de séries que podem ser combinadas para me-

lhora interpretação do pesquisador, se for necessário. Por formalização, para cada tempo t as observações do nível mais baixo até o nível mais alto somadas devem ser iguais ao total:

$$y_t = y_{AA,t} + y_{AB,t} + y_{AC,t} + y_{BA,t} + y_{BB,t} \text{ e assim por diante.}$$

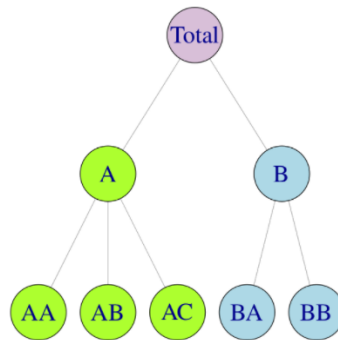


Figura 2.3: Exemplo de Hierarquia de nível 3
 Fonte: Hyndman e Athanasopoulos, 2018

Séries temporais hierárquicas geralmente são providas de divisões geográficas (Hyndman e Athanasopoulos, 2018). No exemplo das chuteiras vendidas, o total pode ser desagregado por país, região, estado, município e assim por diante. A marca pode desagregar também por tipo de produto e informação geográfica provocando uma divisão hierárquica mais ramificada. Para estes casos são denominados como séries temporais agrupadas.

As séries temporais agrupadas não necessariamente se desagregam de uma única maneira hierárquica. No exemplo das chuteiras podemos desagregar todos os níveis geográficos por propósito de uso do produto (profissional, semi-profissional e amador) para que seja possível avaliar o ritmo de venda para nichos de jogadores em certas regiões do país em questão.

A Figura 2.4 mostra uma estrutura agrupada de nível 3, sendo o total representado pelo círculo roxo. Esse total pode ser desagregado por atributos (A,B), formando a série $y_{A,t}$ e $y_{B,t}$, ou pelos atributos (X,Y), formando $y_{X,t}$ e $y_{Y,t}$ que também são desagregados em outros atributos. As notações são semelhantes às apresentadas para séries hierárquicas e seguem o mesmo raciocínio. Uma série temporal agrupada nem sempre impõe uma hierarquia entre as classes, de tal forma que não existe apenas uma maneira como a série se agrupa.

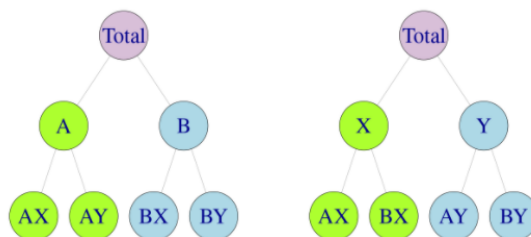


Figura 2.4: Representação de uma estrutura agrupada
Fonte: Hyndman e Athanasopoulos, 2018

2.5.1 Métodos de Resolução e Aplicação

Um dos grandes desafios que podem ocorrer nessa área é modelar casos com poucas observações para níveis mais altos ou mais baixos de hierarquia, conforme o problema. Com isso, serão apresentadas algumas técnicas utilizadas para problemas como esse. Mais especificamente serão explicadas as soluções: *bottom-up* e *top-down*.

- *Bottom-up*

É um método simples que consiste em primeiramente gerar previsões para níveis inferiores de hierarquia e consequentemente somar para os próximos níveis em sequência para gerar uma previsão geral. Como exemplo para a Figura 2.3 é gerado a previsão dos próximos h períodos do último nível como: $\hat{y}_{AA,h}, \hat{y}_{AB,h}, \hat{y}_{AC,h}, \hat{y}_{BA,h}, \hat{y}_{BB,h}$. Somando então as previsões dos níveis mais baixos, podemos obter previsões para os níveis intermediários e também para os mais altos:

$$\begin{aligned}\tilde{y}_h &= \hat{y}_{AA,h} + \hat{y}_{AB,h} + \hat{y}_{AC,h} + \hat{y}_{BA,h} + \hat{y}_{BB,h} \\ \tilde{y}_{A,h} &= \hat{y}_{AA,h} + \hat{y}_{AB,h} + \hat{y}_{AC,h} \\ \tilde{y}_{B,h} &= \hat{y}_{BA,h} + \hat{y}_{BB,h}\end{aligned}$$

Uma vantagem desse método é que, por estar realizando previsões dos níveis mais baixos da estrutura hierárquica, não há perda de informação pelo método de agregação. No entanto, é

comum que os níveis mais baixos sejam formados por séries temporais intermitentes e inflacionadas de zero, o que pode inviabilizar o emprego de técnicas convencionais.

- *Top-down*

Esse tipo de abordagem é adotada para estruturas muito restritas de agregação que, ao contrário da metodologia anterior, realiza a previsão para o nível mais alto da hierarquia e depois é realizada a desagregação até o nível mais baixo da série.

Sejam p_1, p_2, \dots, p_m as proporções de desagregação que ditarão como a previsão total será distribuída para cada nível subsequente, a previsão para os níveis mais baixo seguindo a Figura 2.3 será encontrada por: $\tilde{y}_{AA,t} = p_1 \hat{y}_t$, $\tilde{y}_{AB,t} = p_2 \hat{y}_t$, $\tilde{y}_{AC,t} = p_3 \hat{y}_t$, $\tilde{y}_{BA,t} = p_4 \hat{y}_t$, $\tilde{y}_{BB,t} = p_5 \hat{y}_t$.

A definição dessas proporções pode ser realizada de maneira arbitrária, conforme o problema e a desagregação dos tipos de hierarquia. Como exemplo, foi visto em Gross e Sohl, 1990 que as proporções médias e proporções das médias históricas performaram bem no processo de previsão da quantidade de itens em linhas de produção. Essas proporções médias são definidas por:

$$p_j = \frac{1}{T} \sum_{t=1}^T \frac{y_{j,t}}{y_t}$$

Para $j = 1, \dots, m$ em que p_j representa a média das proporções históricas do nível j da hierarquia durante o período $t = 1, \dots, T$ sobre o total agregado y_t . Já as proporções das médias históricas são escritas como:

$$p_j = \frac{\sum_{t=1}^T \frac{y_{j,t}}{T}}{\sum_{t=1}^T \frac{y_t}{T}}$$

Em que cada proporção p_j captura a média histórica dos valores da série do nível j em relação ao valor total agregado de y_t .

Esse método também é relativamente simples por precisar apenas de uma modelagem para o topo da hierarquia e uma estratégia de alocação desses valores preditos para os níveis seguin-

tes. Uma desvantagem desse método é a perda de informação devido à agregação. Utilizando essa metodologia não é possível captar características individuais de cada série como: eventos especiais, dinâmica da série e sazonalidade das próprias proporções adotadas.

2.6 Light Gradient Boosting Machine

O *Light Gradient Boosting Machine* (LGBM, s.d.) é um algoritmo com estrutura de *Gradient Boosting Decision Tree* que realiza o aprendizado baseado em árvores de decisão. O LGBM foi desenvolvido para ser um método distributivo e eficiente com as seguintes vantagens: tempo reduzido de treinamento, maior eficiência, menor uso de memória, melhor acurácia, capacidade em lidar com dados de larga escala e realizar o aprendizado de forma paralela e distributiva.

O *Gradient Boosting Decision Tree* (GBDT) é um algoritmo popular no campo de *Machine Learning* devido a sua interpretabilidade. Esse modelo é iterativo e se resume a combinar árvores de decisão que minimizem os seus erros residuais, porém o maior custo do GBDT está no processo de aprendizado das árvores de decisão, mais precisamente onde encontrar os pontos ótimos de separação dentro de cada variável no qual minimize o erro residual. Com o crescimento do banco de dados de treinamento (em termos de observações e variáveis) a eficiência e escalabilidade do algoritmo tende a decrescer cada vez mais.

Para solucionar esse problema existem algumas implementações efetivas na literatura como o *XGBoost*, *Catboost* e *pGBRT*. Cada metodologia se destaca pela maneira de realizar a separação otimizada das árvores de decisão e obter o menor erro residual. O LGBM propõe solucionar os problemas de otimização e escalabilidade utilizando duas técnicas: *Gradient-based-One-Side-Sampling* (GOSS) e *Exclusive Feature Bundling* (EFB)

O GOSS é uma técnica de amostragem que encontra uma boa forma de reduzir o número de observações no banco de dados e manter a acurácia do modelo. O erro residual, ou gradiente, fornece informação valiosa para o treinamento do modelo, pois as observações associadas a pequenos gradientes possuem um erro de treinamento pequeno (Ke et al., 2017). Sendo assim,

uma primeira ideia seria eliminar as partes que possuem maiores gradientes e trabalhar com partes do banco com erros menores de treinamento, mas isso poderia ocasionar um certo viés ao modelo, além de modificar a distribuição do banco de dados afetando a acurácia do modelo treinado.

A fim de captar uma maior variabilidade dentro dos dados para que o modelo seja mais assertivo, o GOSS propõe manter essas instâncias de dados com altos gradientes e obter uma amostra nas partes com gradientes mais baixos. Primeiramente, são ordenadas as instâncias com maiores gradientes numa ordem decrescente, depois seleciona-se as $\alpha\%$ maiores instâncias de maiores gradientes e realiza-se uma amostra aleatória da outra parte do banco ($1 - \alpha\%$) com menores gradientes. Depois são determinados os pontos de separação e conseqüentemente realizadas as partições das árvores de acordo com o ganho estimado de variância sobre os dados selecionados. Seja $\tilde{V}_j(d)$ o ganho de variância estimada:

$$\tilde{V}_j(d) = \frac{1}{n} \left(\frac{(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i)^2}{n_r^j(d)} \right)$$

Em que $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$, g_i é referente aos gradientes da função perda em relação ao *output* do modelo e o coeficiente $\frac{1-a}{b}$ é usado para normalizar a soma de gradientes sobre B em relação ao tamanho de A^c . Dessa forma o custo computacional é menor em comparação com os outros algoritmos de *Boosting* que utilizam uma definição mais acurada de variância sobre o banco de dados completo. Segundo Ke et al., 2017, esse processo não perde muito em acurácia e performa melhor do que apenas realizar uma amostra aleatória no banco de dados.

O EFB é uma técnica desenvolvida a fim de diminuir a dimensão do banco de dados utilizado para o treinamento sem perda excessiva de informação. Normalmente, os dados utilizados para esses tipos de modelo possuem um grande número de variáveis com informações exclusivas, isto é, não há valores que não sejam iguais a zero simultaneamente, como geralmente as

variáveis categóricas são transformadas em *dummies* ou pelo processo de *One-Hot-Encoding*. Portanto, o EFB procura agrupar essas variáveis em uma única *feature* para que a complexidade da geração dos histogramas no processo de geração das árvores passe a ser menor por trabalhar com uma quantidade menor de *features* em comparação ao banco original. De toda forma, há dois pontos de atenção para essa técnica: quais *features* devem ser misturadas e como construir essa mistura.

O Algoritmo 3 da Figura 2.5, retirada do artigo de Ke et al., 2017, mostra como é realizado esse agrupamento das variáveis de informação exclusiva. Primeiramente, é construído um grafo com arestas ponderadas entre elas, conforme o total de conflitos entre cada *feature*. Depois, estes grafos são ordenados de forma decrescente e consequentemente assinalados para um agrupamento existente conforme a quantidade de conflitos (controlado por K) ou cria-se um novo grupo caso não encontre conflitos suficientes dentro do agrupamento analisado.

Outro ponto de atenção é sobre como construir esses agrupamentos de forma que possa reduzir a complexidade do processo de treinamento. Para assegurar que os valores originais das variáveis fossem identificados dentro de cada agrupamento, são realizados deslocamentos dentro dos valores originais das variáveis, como por exemplo: A variável A possui valores entre [0,10) e a variável B de [0,20), para que os valores da *feature* B sejam reconhecidos dentro do agrupamento é realizado um deslocamento de 10 unidades fazendo com que seu novo intervalo seja de [10,30); Após isso é seguro agrupar as duas variáveis e utilizar um intervalo de [0,30) para repor as duas variáveis originais. O Algoritmo 4 da Figura 2.5 detalha como é feito o processo.

Após esses procedimentos no banco de dados, é realizado o processo usual de *Gradient Boosting* onde é determinado uma função perda, ou Gradiente conforme os autores, e o objetivo é combinar árvores de decisão iterativamente que consigam minimizar esta função conforme a estratégia de validação do modelo adotada pelo usuário. Com essas transformações nos dados espera-se encontrar uma acurácia semelhante aos demais algoritmos em um tempo reduzido de treinamento.

<p>Algorithm 3: Greedy Bundling</p> <p>Input: F: features, K: max conflict count</p> <p>Construct graph G</p> <p>searchOrder $\leftarrow G.sortByDegree()$</p> <p>bundles $\leftarrow \{\}$, bundlesConflict $\leftarrow \{\}$</p> <p>for i in searchOrder do</p> <p style="padding-left: 20px;">needNew \leftarrow True</p> <p style="padding-left: 20px;">for $j = 1$ to len(bundles) do</p> <p style="padding-left: 40px;">cnt \leftarrow ConflictCnt(bundles[j], $F[i]$)</p> <p style="padding-left: 40px;">if cnt + bundlesConflict[j] $\leq K$ then</p> <p style="padding-left: 60px;">bundles[j].add($F[i]$), needNew \leftarrow False</p> <p style="padding-left: 60px;">break</p> <p style="padding-left: 20px;">if needNew then</p> <p style="padding-left: 40px;">Add $F[i]$ as a new bundle to bundles</p> <p>Output: bundles</p>	<p>Algorithm 4: Merge Exclusive Features</p> <p>Input: numData: number of data</p> <p>Input: F: One bundle of exclusive features</p> <p>binRanges $\leftarrow \{0\}$, totalBin $\leftarrow 0$</p> <p>for f in F do</p> <p style="padding-left: 20px;">totalBin += f.numBin</p> <p style="padding-left: 20px;">binRanges.append(totalBin)</p> <p>newBin \leftarrow new Bin(numData)</p> <p>for $i = 1$ to numData do</p> <p style="padding-left: 20px;">newBin[i] $\leftarrow 0$</p> <p style="padding-left: 20px;">for $j = 1$ to len(F) do</p> <p style="padding-left: 40px;">if $F[j].bin[i] \neq 0$ then</p> <p style="padding-left: 60px;">newBin[i] $\leftarrow F[j].bin[i] + binRanges[j]$</p> <p>Output: newBin, binRanges</p>
---	--

Figura 2.5: Algoritmos incluídos no processo EFB
 Fonte: Ke et al., 2017

2.7 Métricas de avaliação do modelo

2.7.1 Weighted Root Mean Squared Scaled Error (WRMSSE)

O WRMSSE é uma métrica de avaliação voltada para os casos de séries temporais hierárquicas ou agrupadas. Ela tem como ponto principal relacionar o peso da hierarquia de cada série ao erro de previsão realizada pelo modelo. Makridakis, Spiliotis e Assimakopoulos, 2020 cita que essa métrica se destaca por: ser independente de escala o que possibilita comparar séries com diferentes hierarquias, não depender de divisões com valores iguais a zero ou muito próximos a ele e penalizar erros positivos/negativos assim como altos/baixos valores de previsões igualmente. Para efeitos de comparação, quanto menor o valor do WRMSSE melhor o ajuste do modelo. Seu cálculo é dado por:

$$WRMSSE = \sum_{i=1}^I w_i * RMSSE$$

$$RMSSE = \sqrt{\frac{1}{h} \frac{\sum_{i=n+1}^{n+h} (y_i - \hat{y}_i)^2}{\frac{1}{n-1} \sum_{i=2}^n (y_i - y_{i-1})^2}}$$

A comissão organizadora da competição não divulga o cálculo dos pesos w_i para não influenciar os modelos desenvolvidos pelos participantes.

2.7.2 Root Mean Squared Error (RMSE)

O RMSE é uma forma de medir o tamanho do erro quadrático médio de suas previsões. O seu cálculo é dado pela raiz quadrada da média das diferenças quadráticas entre os valores atuais e preditos. O RMSE tem como característica dar um peso muito alto para grandes discrepâncias, sendo totalmente indicado para casos onde grandes erros são indesejáveis. O RMSE é calculado da seguinte forma:

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

2.7.3 Mean Absolute Error (MAE)

O MAE é uma métrica que avalia a magnitude média dos erros em suas previsões independente do sinal. Ela é calculada a partir da média entre a diferença absoluta dos valores reais e preditos onde todas as observações possuem pesos iguais. Ela pode variar de zero a infinito e quanto menor o valor dessa métrica menor o erro relativo aos valores originais. O MAE é calculado por:

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

2.7.4 Mean Absolute Percentage Error (MAPE)

O MAPE é uma medida relativa muito semelhante ao MAE e por ser uma medida expressa em porcentagem, que é independente de escala, pode ser utilizada para comparar previsões em diferentes hierarquias ou grupos. Essa é uma medida fácil de ser explicada, mas se torna impossível de ser calculada para valores originais iguais a zero. O seu cálculo é dado por:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Capítulo 3

Banco de dados

O banco de dados deste trabalho foi cedido pela rede de supermercados Walmart e utilizado pela competição *M5 Forecasting - Accuracy* disponibilizada no site *Kaggle*². Este banco é referente às quantidades de vendas diárias de 3.409 produtos pelos Estados Unidos entre 2011 e 2016 e a competição era vencida por quem desenvolvesse um modelo que obtivesse o menor valor da métrica WRMSSE durante o período indicado.

Mais precisamente, esses dados envolvem produtos que são classificados em: três categorias (*Hobbies, Foods e Household*), sete departamentos nos quais os três tipos de produtos são desagregados e são vendidos através de dez lojas localizadas em três Estados (Califórnia, Texas e Wisconsin). Sendo assim, a informação mais desagregada (unidade de produto vendida) pode ser agrupada tanto por localização (Loja e Estado), como por informação do produto (Categoria e Departamento). A Figura 3.1 ajuda a entender a estrutura hierárquica dos dados.

Em conjunto aos dados sobre vendas, foi disponibilizado um banco de dados com informações de calendário contendo: data, dia da semana, se havia algum evento relevante no dia (ex: Dia dos Pais, Natal, Dia do Trabalho), o tipo do evento (ex: feriado nacional, celebração religiosa, final de um evento esportivo) e se era dia de recebimento de auxílio do governo americano. Outra fonte de dados fornecida foi a relação dos preços históricos de prateleira de todos

²<https://www.kaggle.com/competitions/m5-forecasting-accuracy>

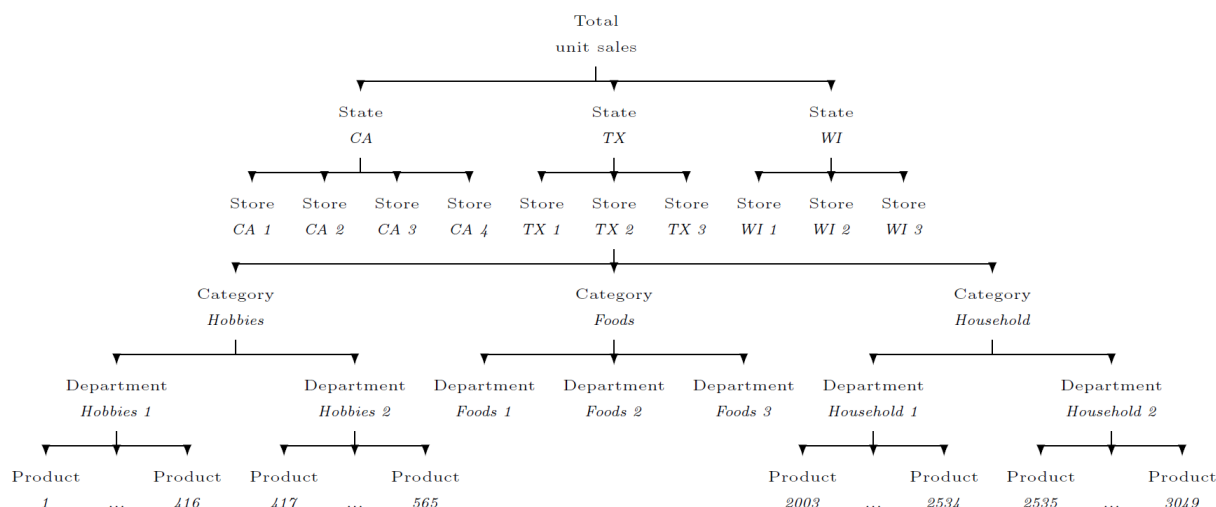


Figura 3.1: Organograma do banco de dados
 Fonte: Makridakis, Spiliotis e Assimakopoulos, 2020

os produtos durante o período de 2011 a 2016.

Sendo assim, foi feito o cruzamento entre esses dados de tal forma que possibilitou construir novas variáveis para fazer parte do banco de dados para modelagem. Por meio das unidades de produtos vendidas foram construídas colunas contendo *lags* de 28 a 35 dias em torno dos itens vendidos, o motivo para ser a partir de 28 dias é por representar o tamanho da janela de previsão adotada pela competição e estar até os 35 dias foi devido aos limites computacionais da máquina utilizada nesse trabalho. Foi realizado o cálculo de médias móveis sobre esses *lags* com períodos semanais, mensais e semestrais. Pelo histórico de preços foram obtidos os valores mínimos, máximos, médios, seus desvios e quantidades de preços distintos durante o intervalo de estudo para cada produto. Com as informações das datas e calendário foram construídas variáveis auxiliares quanto ao período do ano, véspera de evento relevante e algumas indicadores relacionadas aos auxílios por parte do governo americano. Finalizamos a construção do banco de dados de modelagem com a variável que chamamos de h_1 . Ela corresponde à construção da variável contendo uma modelagem estatística com previsões no horizonte de 1 dia, a qual será melhor detalhada na Tabela 3.1.

Tabela 3.1: Relações e descrições de variáveis utilizadas no treinamento do modelo

Relação de Variáveis	Descrição
id	identificadora
item_id	produto
dept_id	departamento
cat_id	categoria
store_id	loja
state_id	estado
date	data
wm_yr_wk	semestre/ano/semana
event_name	nome do evento
event_type	tipo de evento
snap_state	indicadora de auxílio
sell_price	preço de venda
day	dia
month	mês
year	ano
week	semana do ano
weekday	dia da semana
max_price	preço máximo
min_price	preço mínimo
mean_price	preço médio
dev_price	desvio-padrão do preço
dif_price	Quantidade de preços diferentes por produto
lag_k	lag de ordem k sobre as unidades vendidas por produto
rolling_mean_j_k	média móvel nos últimos j dias sobre os lags de ordem k
h_1	previsão um passo a frente do modelo estatístico escolhido

3.1 Análise Exploratória

A análise exploratória do banco de dados pode ser realizada por diversas perspectivas. Por possuir uma estrutura hierárquica, foi visto que cada ramificação possuía características intrínsecas como: volumes discrepantes de vendas de acordo com a hierarquia, sazonalidades diferentes em algumas lojas e tendências específicas em alguns departamentos. Primeiramente, serão analisados os diferentes cenários do comportamento da variável resposta e, posteriormente, seus atributos gerais para efeitos de modelagem.

3.1.1 Vendas Totais

A série do total de unidades vendidas, que representa um resultado agregado da variável resposta no maior nível de hierarquia, apresentada na Figura 3.2, possui uma tendência de crescimento pelo período em estudo com resultados que sugerem um aumento nos períodos seguintes. É possível perceber que há diferentes sazonalidades dentro da série (principalmente semanal e anual) e quedas bruscas apresentadas quase sempre ao final de cada ano. Os ciclos repetitivos da série podem ser explicados pelos hábitos de compras mensais e semanais dos clientes, enquanto que os resultados zerados ao fim do ano ocorrem exatamente nos dias de Natal quando o supermercado não funciona.

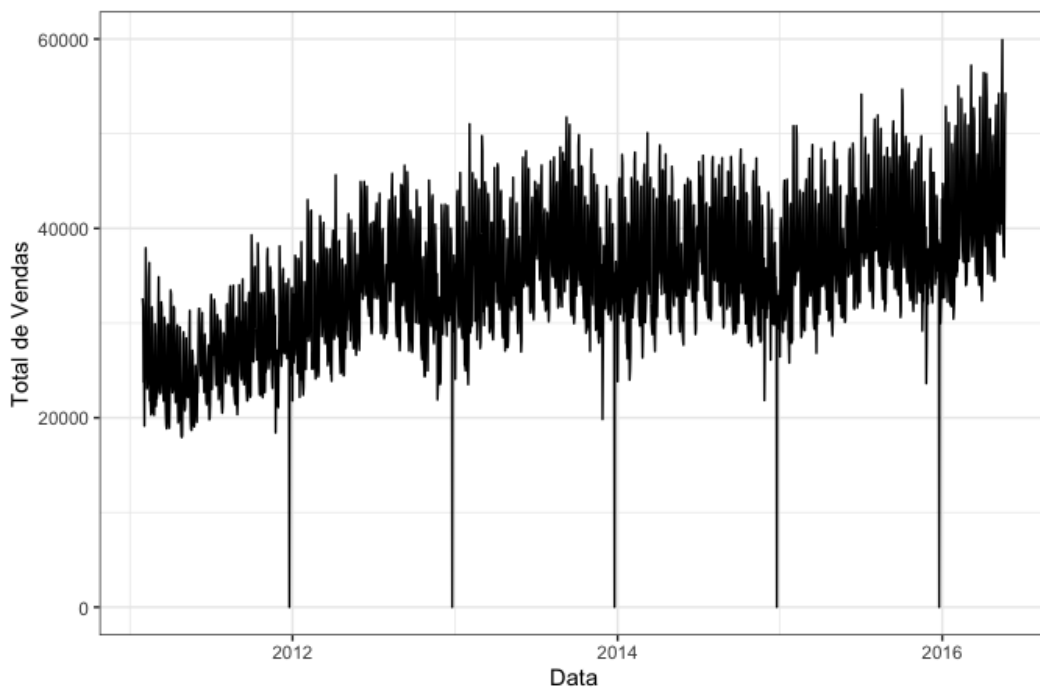


Figura 3.2: Série diária da quantidade de produtos vendidos

A Figura 3.3 representa a decomposição da série do total de unidades vendidas, de acordo com as suas características. O primeiro gráfico representa a série total, visto na Figura 3.2, o segundo mostra uma estimativa da tendência, o terceiro é a estimativa da sazonalidade de frequência semanal, o quarto apresenta a sazonalidade com frequência mensal, o quinto exibe a

sazonalidade anual e, por último, são apresentados os fatores aleatórios da série.

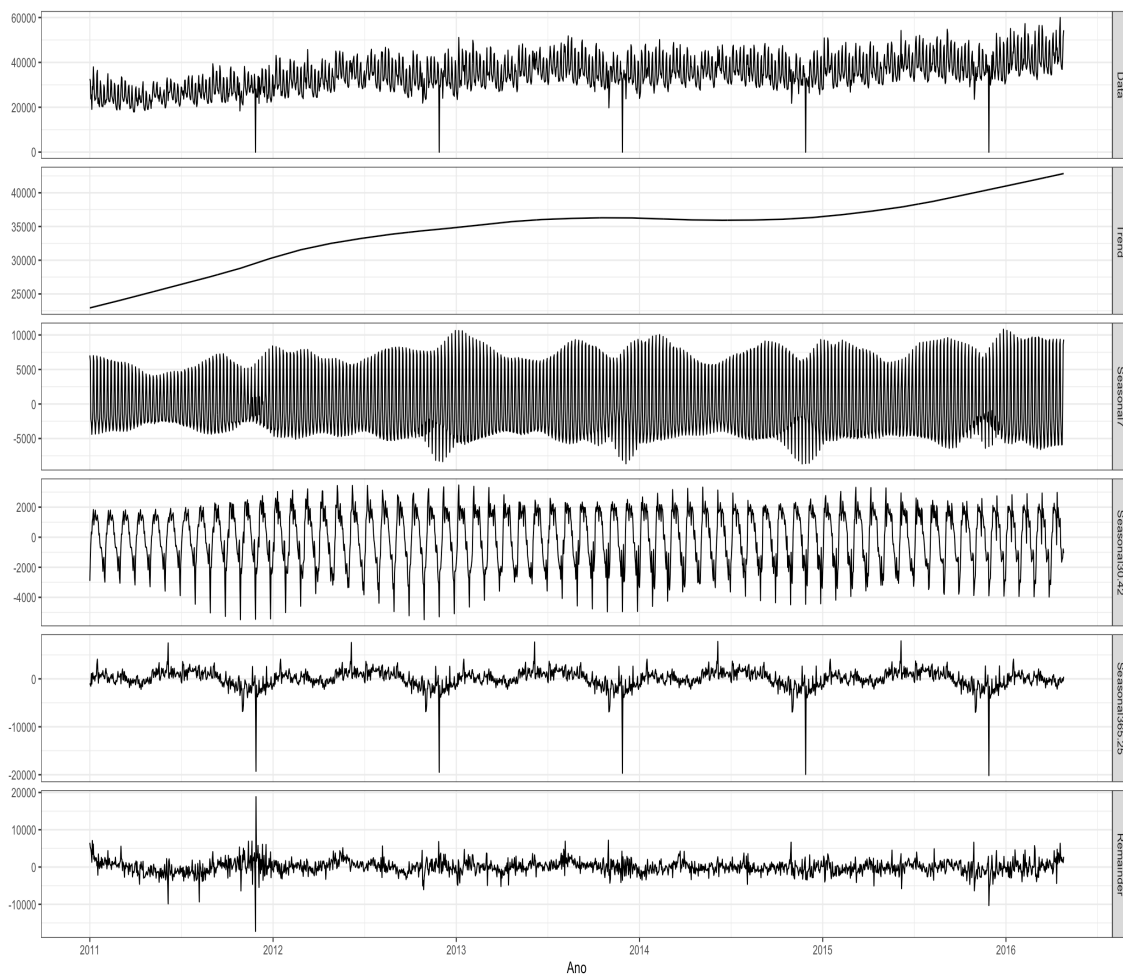


Figura 3.3: Decomposição da série do total de vendas

O gráfico de tendência presente na decomposição aponta que o ano de 2016 aparenta um crescimento de vendas maior que os anos observados anteriormente. Também fica claro que há três sazonalidades bem definidas (semanal, mensal e anual). Além disso há a necessidade de se precaver ao realizar previsões em algumas datas festivas, principalmente em relação ao Natal, no qual o padrão de vendas se modifica bastante.

3.1.2 Vendas por Estado

Analisando os mesmos dados, mas para o próximo nível de hierarquia, fica evidente na Figura 3.4 que no estado da Califórnia há uma maior quantidade de vendas no geral, enquanto os estados de Wisconsin e Texas possuem frequências de vendas reduzidas, mas bem semelhantes durante todo o período. Nenhuma das três séries estaduais observadas aparentam grandes discrepâncias da série agregada no sentido de possuírem uma tendência de crescimento e períodos sazonais bem claros.

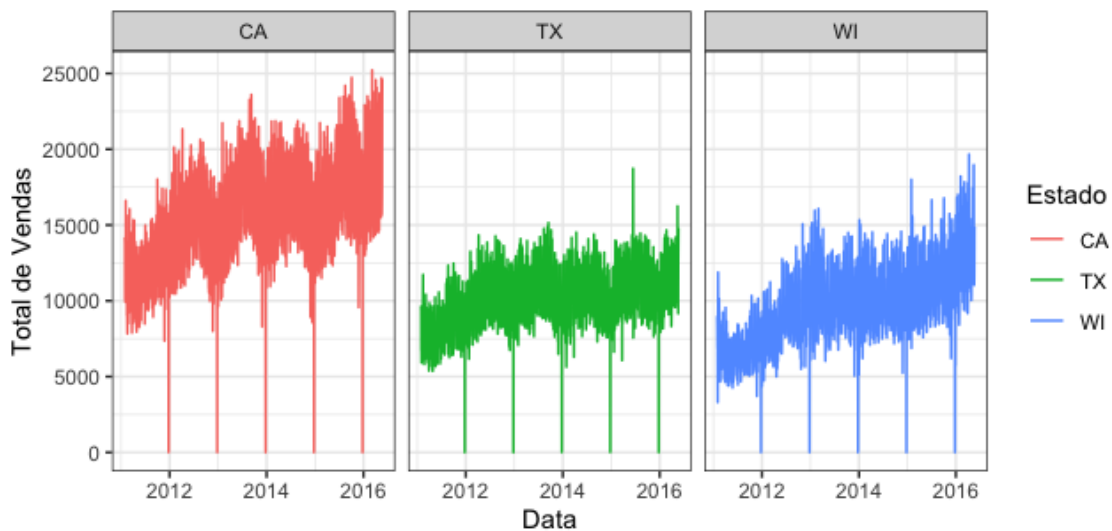


Figura 3.4: Séries de vendas diárias por Estado

3.1.3 Vendas por Loja

O próximo nível da hierarquia é composto por dez lojas, sendo quatro na Califórnia, três no Texas e três no Wisconsin. Podemos perceber na Figura 3.5 que o nível de vendas pelas lojas do Texas permanecem bem semelhantes durante o período em estudo, com destaque para a loja TX_3 que conseguiu elevar seus patamares de venda da TX_1, loja com menor estágio de consumo, para a TX_2, estabelecimento com maior quantidade média de produtos vendidos por mês dentro do Estado. Para as lojas do Wisconsin foi visto um curioso salto nas séries de WI_1

e WI_2 em 2012, enquanto que a WI_3 teve uma queda brusca no nível de produtos vendidos ao longo da série e aparenta estar numa tendência de diminuição. As lojas da Califórnia são bem separadas no quesito volume de vendas, porém nota-se que a loja CA_2 tem um declínio em 2015, para depois crescer para patamares acima do que o habitual.

A loja que chama mais atenção com seu volume de vendas é a CA_3, que a partir do ano de 2012 obteve uma quantidade mensal de vendas variando entre 15.000 e 20.000 produtos vendidos. As demais lojas não se separam muito em níveis de vendas durante os anos em estudo, sendo que esses valores mensais variaram entre 5.000 a 15.000 unidades de produtos comercializadas com algumas lojas apresentando aumentos ou quedas significativas durante o período.

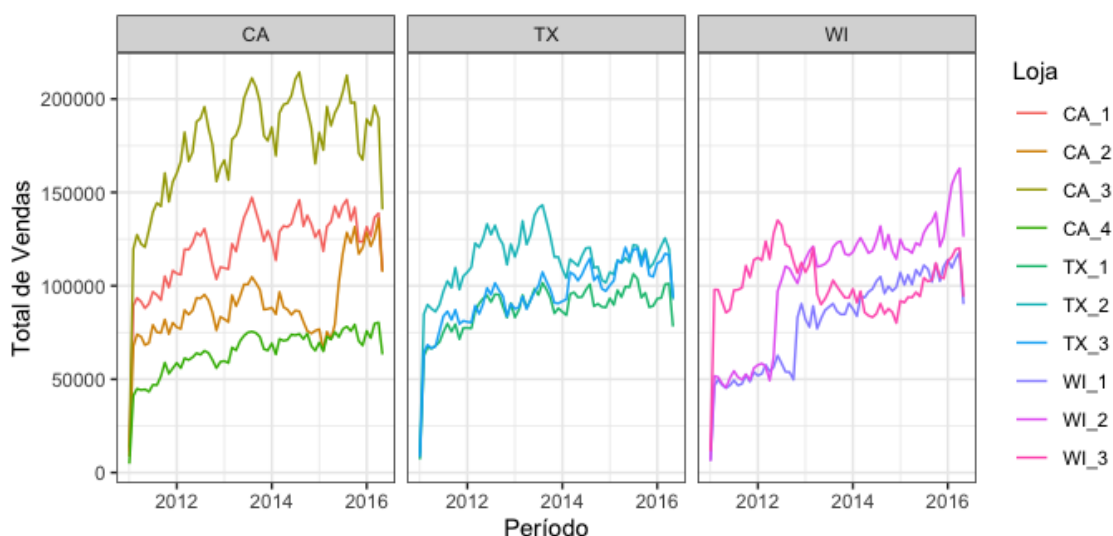


Figura 3.5: Séries de vendas mensais por loja e Estado

3.1.4 Vendas por Categoria e Estado

Analisando as séries por categoria do produto (*Hobbies*, *Foods* e *Household*) foi visto na Figura 3.6 que a categoria *Foods* apresenta maior quantitativo de vendas entre os três estados, seguido dos itens de *Household* e *Hobbies*. As categorias de produtos também aparentam possuir uma tendência de aumento na quantidade de vendas e sazonalidades durante o período em

estudo. Para o próximo nível de hierarquia (departamento), as séries se tornam muito específicas e não há razão para explicarmos uma a uma. Mas, obviamente, há departamentos dentro da categoria *Foods*, por exemplo, que se sobressaem dos demais como foi visto nos eventos segregados em lojas, em que uma loja dentro de cada estado tinha um nível de vendas superior as outras.

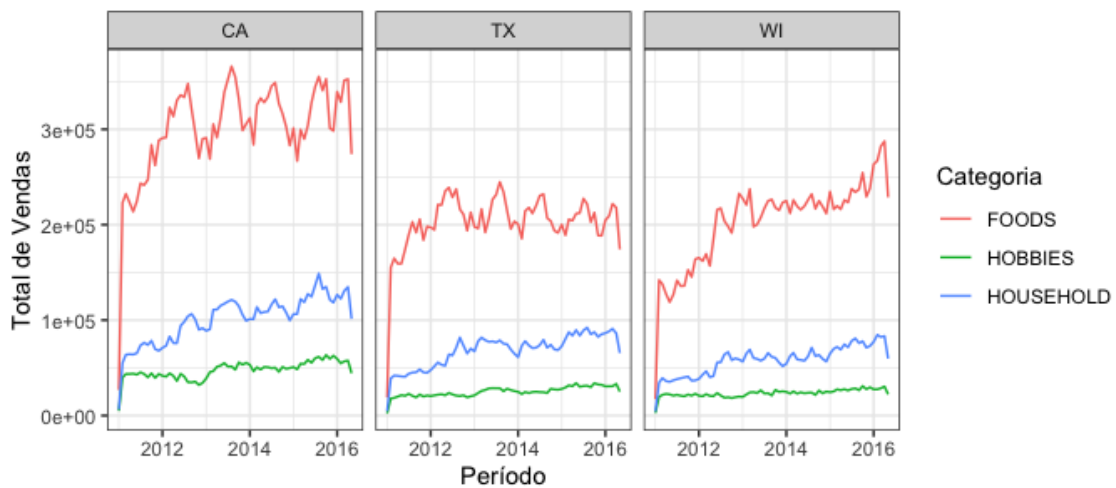


Figura 3.6: Séries de vendas mensais por Categoria e Estado

3.1.5 Vendas por Departamento e Estado

Analisando o comportamento das vendas entre os departamentos na Figura 3.7 fica evidente que para os três Estados que o FOODS_3 é aquele que possui o maior destaque. Seguido pelo HOUSEHOLD_1, com exceção do Estado do Wisconsin, que a partir de 2014 teve um aumento substancial em suas quantidades vendidas. Como ponto de menor relevância, o departamento HOBBIES_2 foi unânime entre as 3 regiões estudadas. As demais séries são equiparadas no quesito quantitativo dentro de cada estado.

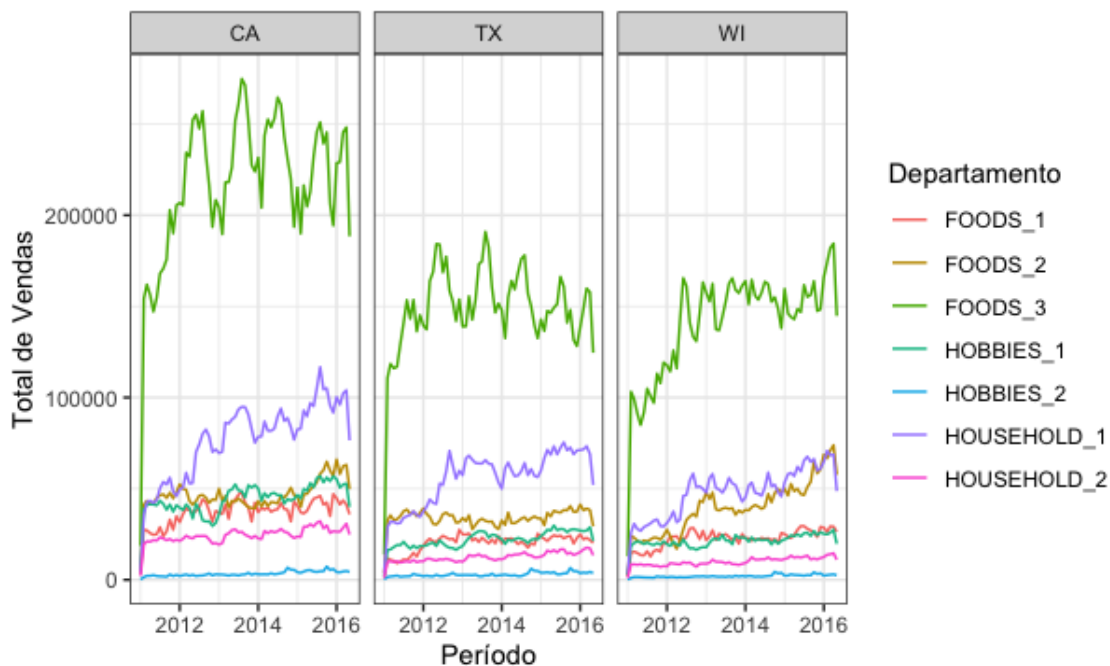


Figura 3.7: Séries de vendas mensais por Departamento e Estado

3.1.6 Constatações

Em resumo, pela análise exploratória que foi possível realizar acerca da variável resposta pudemos identificar padrões de sazonalidade semanal, mensal e anual dentro da série diária de vendas agregadas, assim como uma tendência crescente da série. Em algumas datas comemorativas o padrão da série se modifica, como foi visto para os dias de natal em que os mercados ficaram fechados.

Foi possível avaliar que o Estado com maior volume de vendas é a Califórnia, enquanto que os demais estados possuem frequências de vendas reduzidas, mas bem semelhantes a série total durante o período. Também foi visto que algumas lojas se destacam pelo poder de recuperação, como foi o caso da CA_3, ou por tendência de queda, por exemplo: WI_3. A categoria de produto que possui maior patamar de mercadorias vendidas é a *Foods* seguido de *Household* e *Hobbies*, o que ocasiona o Departamento FOODS_3 a ter o maior destaque de vendas, enquanto HOBBIES_2 teve a menor relevância entre os três Estados.

Capítulo 4

Metodologia

Conforme apresentação prévia na Tabela 1, foi realizado um procedimento de construção de variáveis estatísticas a fim de auxiliar o poder preditivo do modelo. A seguir será apresentado todo o processo de construção da variável e posteriormente como o conceito foi aplicado ao banco de dados utilizado.

4.1 Construção de variáveis estatísticas para Machine Learning

Um dos processos na elaboração de um modelo de *Machine Learning* passa pela construção de um banco de dados com informações relevantes que auxiliem a explicar a variável resposta. No contexto de séries temporais é comum adicionar variáveis relacionadas ao tempo, como: data do evento, dia da semana, mês e ano em questão. Ao redor dessas informações também é possível extrair mais elementos como a temporada do ano, eventos relevantes e feriados que geralmente são fatores importantes para a análise.

Em conjunto às informações temporais, podemos incluir informações estatísticas que também auxiliem dentro do modelo. Analisando o conteúdo dos pacotes **tsfeatures** (Hyndman et al., 2019), **TSELM** (Barandas et al., 2020), **tsfresh** (Christ et al., 2018), **FATS** (Nun et al., 2015) e as demais referências na bibliografia percebemos uma lacuna quanto à criação de variáveis

que utilizem modelos de séries temporais como o ARMA e o TBATS.

A metodologia adotada nesse trabalho utiliza o conceito de janelas deslizantes e consiste em criar uma coluna no banco de dados com os valores preditos pela metodologia estatística aplicada no intervalo temporal escolhido e, a partir desse período, gerar uma nova estimativa do modelo para que possam ser feitas novas previsões. A seguir é listado o passo a passo para construção da *feature* em um cenário que envolva uma separação temporal na série entre treinamento, validação e teste no banco de dados.

1. Seja y_i a série temporal de interesse, vamos dividi-la em 3 partes: treinamento ($i = 1, \dots, t$), validação ($i = t+1, \dots, v$) e teste ($i = v+1, \dots, n$), sendo elas ordenadas de maneira temporal ($i = 1, \dots, n$);
2. Estima-se um modelo estatístico para uma janela do banco de dados de treinamento (em que $i = 1, \dots, q \forall q \leq t$) e utiliza-se este modelo para prever valores h passos à frente;
3. Seja c_i a coluna desejada, seu preenchimento será realizado a partir dos valores preditos pelo modelo ajustado na seguinte forma: $c_i = \hat{y}_i$. Ou seja, os valores previstos pela metodologia no dia i corresponde ao valor na coluna c atrelada ao dia i . Considera-se c_i nulo quando i for menor ou igual a q , tamanho da janela inicial de ajuste do primeiro modelo estatístico.
4. Após uma quantidade k de previsões, reajustamos a janela temporal com os valores originais da série de interesse para reestimar o modelo e preencher por outras k observações. E assim repetir o processo de reestimação até o final do período de treinamento.
5. A partir do período de validação/teste, em que consideramos não ter os verdadeiros valores da série para simular a aplicação do modelo em um novo cenário, utilizamos das previsões geradas pelo modelo de *Machine Learning* para reestimar o modelo estatístico até a data limite de treinamento/validação e preencher a variável com valores das previsões realizadas pelo modelo estatístico.

§4.1. Construção de variáveis estatísticas para Machine Learning

A seguir é apresentada uma ilustração do preenchimento da coluna adotando a metodologia explicada acima para uma série contendo 15 dias e um horizonte de previsão de apenas 1 dia.

	Treino									Validação			Teste		
	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15
Passo 1	Estimação do Modelo Estatístico			Previsão											
Passo 2	Utilização do Modelo estimado no passo 1 com o valor original do dia 4				Previsão										
Passo 3	Utilização do Modelo estimado no passo 1 com o valor original do dia 4 e 5					Previsão									
Passo 4	Estimação de um novo Modelo Estatístico						Previsão								
Passo 5	Utilização do Modelo estimado no passo 4 com o valor original do dia 7							Previsão							
Passo 6	Utilização do Modelo estimado no passo 4 com o valor original do dia 7 e 8								Previsão						
Passo 7	Estimação de um novo Modelo Estatístico									Previsão					
Passo 8	Utilização do Modelo do passo 7 adicionando o valor estimado no dia 10 para gerar previsão do dia 11										Previsão				
Passo 9	Utilização do Modelo do passo 7 adicionando o valor estimado no dia 11 para gerar previsão do dia 12											Previsão			
Passo 10	Estimação de um novo Modelo Estatístico												Previsão		
Passo 11	Utilização do Modelo do passo 10 adicionando o valor estimado no dia 13 para gerar previsão do dia 14													Previsão	
Passo 12	Utilização do Modelo do passo 10 adicionando o valor estimado no dia 14 para gerar previsão do dia 15														Previsão
Coluna Criada	Nulo	Nulo	Nulo	Valor Predito em d4	Valor Predito em d5	Valor Predito em d6	Valor Predito em d7	Valor Predito em d8	Valor Predito em d9	Valor Predito em d10	Valor Predito em d11	Valor Predito em d12	Valor Predito em d13	Valor Predito em d14	Valor Predito em d15

Figura 4.1: Exemplo de construção de variável

Pelo processo apresentado na Figura 4.1, primeiramente treinamos o modelo entre os dias d1 e d3, para realizar uma primeira previsão em d4. Somente a partir dessa previsão que a coluna é preenchida, como fica evidenciado na última linha da Figura 4.1 'Coluna Criada'. No segundo passo utilizamos o mesmo modelo estimado, porém com o valor original da variável no dia 4 para realizar a previsão no dia 5 e assim seguimos este padrão com uma reestimação do modelo a cada 3 dias.

Atenta-se ao fato que para o período de validação e teste não aproveitamos os valores originais da série para realizar as previsões, pois ao utilizarmos o modelo em demais datas futuras não teremos a informação dos valores reais como no período de treinamento. Para este período em que não enxergamos os verdadeiros valores, alimentamos o modelo estatístico com as previsões obtidas pelo modelo de *Machine Learning*. Seja uma série de duas hierarquias no formato $y_t = y_{A,t} + y_{B,t}$, a aplicação da metodologia apresentada na Figura 4.1 no banco de dados ficaria

da forma apresentada na Tabela 4.1.

Tabela 4.1: Exemplo de construção da variável h_1 em estrutura de dados hierarquizados

y_t	Dia	h_1
$y_{A,1}$	d1	Nulo
$y_{B,1}$	d1	Nulo
$y_{A,2}$	d2	Nulo
$y_{B,2}$	d2	Nulo
$y_{A,3}$	d3	\hat{y}_3
$y_{B,3}$	d3	\hat{y}_3
$y_{A,4}$	d4	\hat{y}_4
$y_{B,4}$	d4	\hat{y}_4
$y_{A,5}$	d5	\hat{y}_5
$y_{B,5}$	d5	\hat{y}_5
\vdots	\vdots	\vdots
$y_{A,15}$	d8	\hat{y}_{15}
$y_{B,15}$	d8	\hat{y}_{15}

A proposta para este trabalho é utilizar o modelo TBATS em níveis mais altos da estrutura hierárquica, os quais o modelo pode apresentar um bom desempenho e, em seguida, compilar a projeção dos níveis mais altos gerados pelo TBATS com o comportamento dos níveis mais baixos através de um algoritmo de *Machine Learning*, que no caso foi escolhido o LGBM, mas que também poderia ser qualquer outro.

4.2 Aplicação do conceito de construção da variável estatística

Para dar início ao processo de construção da variável estatística definimos utilizar como base para as nossas previsões o modelo TBATS. Avaliando o formato da série no nível mais agregado (Figura 3.2) podemos perceber que se trata de uma série de alta frequência com múltiplas sazonalidades envolvidas, o que capacita o modelo TBATS para representar o comportamento das vendas no alto da hierarquia. Optamos por utilizar um horizonte de previsão mínimo ($h = 1$ dia) por ter uma menor incerteza no processo de previsão.

Em um segundo momento é definida a janela de ajuste do primeiro modelo TBATS. Durante

§4.2. Aplicação do conceito de construção da variável estatística

a seção dos resultados testamos o poder dessa variável em três cenários, mas a seguir ilustraremos o caso para uma aplicação do TBATS nos primeiros 750 dias. Para este caso, somente a partir do 750º dia a variável é preenchida com as previsões do modelo TBATS ajustado e seus parâmetros são atualizados a cada 30 dias por um novo modelo calculado para que os valores se mantenham fidedignos.

Essa variável é de suma importância para o trabalho pois a intenção de incluí-la dentro do modelo LGBM foi adicionar um possível fator sazonal, no qual espera-se que o TBATS possa explicar e contribuir positivamente para a acurácia do modelo. Tivemos como objetivo avaliar a sua importância, medir o seu poder de previsão e propor uma nova técnica para construção de variáveis aplicada a problemas desse tipo. A Figura 4.2 ilustra como foi o processo de construção da variável h_1 na prática.

	Treino									Validação			Teste		
	Dia 1	...	Dia 750	Dia 751	...	Dia 780	Dia 781	...	Dia 1913	Dia 1914	...	Dia 1941	Dia 1942	...	Dia 1969
Passo 1	Estimação do Modelo Estatístico			Previsão											
Passo 2	Utilização do Modelo estimado no passo 1 com o valor original do dia 751			Previsão											
⋮	⋮			⋮											
Passo 30	Utilização do Modelo estimado no passo 1 com o valor original até o dia 779			Previsão											
Passo 31	Estimação de um novo Modelo Estatístico						Previsão								
Passo 32	Utilização do Modelo estimado no passo 31 com o valor original do dia 781						Previsão								
⋮	⋮						⋮								
Passo 1.163	Utilização do Modelo estimado no passo 1.140 com o valor original até o dia 1912									Previsão					
Passo 1.164	Estimação de um novo Modelo Estatístico										Previsão				
Passo 1.165	Utilização do Modelo do passo 1.164 adicionando o valor estimado no dia 1.914 para gerar previsão do dia 1.915										Previsão				
⋮	⋮										⋮				
Passo 1.191	Utilização do Modelo do passo 1.190 adicionando o valor estimado no dia 1.940 para gerar previsão do dia 1.941										Previsão				
Passo 1.192	Estimação de um novo Modelo Estatístico												Previsão		
Passo 1.193	Utilização do Modelo do passo 1.192 adicionando o valor estimado no dia 1.942 para gerar previsão do dia 1.943												Previsão		
⋮	⋮												⋮		
Passo 1.221	Utilização do Modelo do passo 1.220 adicionando o valor estimado no dia 1.968 para gerar previsão do dia 1.969														Previsão
Coluna Criada	Nulo	Nulo	Nulo	Valor Predito em d751	...	Valor Predito em d780	Valor Predito em d781	...	Valor Predito em d1913	Valor Predito em d1914	...	Valor Predito em d1941	Valor Predito em d1942	...	Valor Predito em d1969

Figura 4.2: Processo de construção da variável h_1

Capítulo 5

Resultados

O objeto de modelagem é a quantidade de vendas distribuída no tempo conforme o seu nível de hierarquia e o objetivo principal é realizar previsões acuradas conforme as diversas perspectivas observadas no capítulo de análise exploratória. Existem diversas formas de abordar esse problema, seja por remoção de tendência e sazonalidade para modelagem da série estacionária, por aplicação do TBATS, que performa muito bem em cenários de múltiplas sazonalidades ou pelo treinamento de um modelo de *Machine Learning* que geralmente se sobressai em previsões com quantidades massivas de dados (Makridakis, Spiliotis e Assimakopoulos, 2020).

Para efeitos de *benchmark* serão feitas cinco abordagens e, posteriormente, comparações pelo mesmo cenário temporal. O primeiro procedimento será por meio do ajuste de um TBATS na série do total de unidades vendidas desagregando suas previsões utilizando a metodologia *top-down*, posteriormente será realizado um LGBM ao nível de produto e agregação de resultados via *bottom-up*. Depois, serão exibidos outros três modelos LGBM com os mesmos parâmetros do anterior com adição da variável h_1 , variando a janela temporal do primeiro ajuste do TBATS.

Os intervalos temporais serão os mesmos utilizados na competição do *Kaggle* que de 29/01/2011 a 24/04/2016 compreendeu-se como o período de treinamento do modelo, o que configuram 1.913 dias. Desde a data de abertura da competição até o envio dos modelos finais o *Kaggle*

disponibilizava um *ranking* público em que eram enviadas previsões para os dias 25/04/2016 a 22/05/2016 (dia 1.914 até o dia 1.941) e os modelos eram avaliados conforme o WRMSSE (*Weighted Root Mean Squared Scaled Error*). Para este intervalo, todos os competidores possuíam os valores reais e podiam enviar os resultados de seus modelos para compararem as respostas.

O vencedor da competição seria aquele que tivesse o menor WRMSSE para o *ranking* privado, valendo para os dias 23/05/2016 a 19/06/2016, 27 dias seguintes ao período de avaliação pública na plataforma. Para este intervalo os competidores não obtinham os valores reais de vendas e a métrica era calculada a partir das previsões enviadas pelo *site*. A seguir serão debatidos os resultados e as estratégias utilizadas para os cinco modelos treinados.

5.1 Modelo TBATS

Para realizar o treinamento do modelo TBATS adotou-se a estratégia de agregação das unidades vendidas por data para modelagem a nível mais alto de hierarquia. A partir deste modelo treinado que foi adotada a abordagem *top-down* utilizando proporções históricas para avaliação de resultados granulares. Como foi descrito na metodologia, este método sofre de possíveis perdas de captação de características individuais, como possíveis tendências de queda ou poderes de recuperação, que foram vistos no capítulo de estatísticas exploratórias, entretanto o objetivo deste modelo é servir como comparação aos outros seguintes.

O modelo que melhor se ajustou aos dados durante o período de treinamento foi um TBATS de ordem $(1, 2, 2, 0, \{7, 3\}, \{30.42, 7\}, \{365.25, 5\})$ em que o parâmetro ω referente à transformação de Box-Cox é igual a 1, p e q são iguais a 2 e representam o componente do erro modelado por um processo ARMA(p,q), o parâmetro ϕ referente à tendência de Damped equivalente a zero e os primeiros valores que estão entre colchetes representam os períodos sazonais utilizados no modelo, ou seja, ciclos semanais, mensais e anuais enquanto o outro parâmetro corresponde ao número de termos de Fourier utilizados para cada sazonalidade.

Analisando os resíduos do modelo, o primeiro ponto a chamar a atenção, presente na Figura 5.1, são os picos negativos aos finais de cada ano que também foram vistos na Figura 3.2, os quais são referentes aos dias de natal. Além desses pontos, também são vistos picos positivos em torno da metade de cada ano, em que são referentes aos dias anteriores ao feriado de 4 de Julho nos Estados Unidos, fato também visto para os dias anteriores ao *Thanksgiving* que geralmente ocorre nos meses de novembro. Ao avaliar o gráfico entre as autocorrelações dos resíduos podemos perceber a relação entre eles ciclicamente como são os casos dos *lags* de 7 e 15 dias que se sobressaíram em relação aos demais. Ao investigar o histograma dos erros, em conjunto com a sua densidade, percebemos um indício de que os resíduos estão em torno de zero com sua variância controlada.

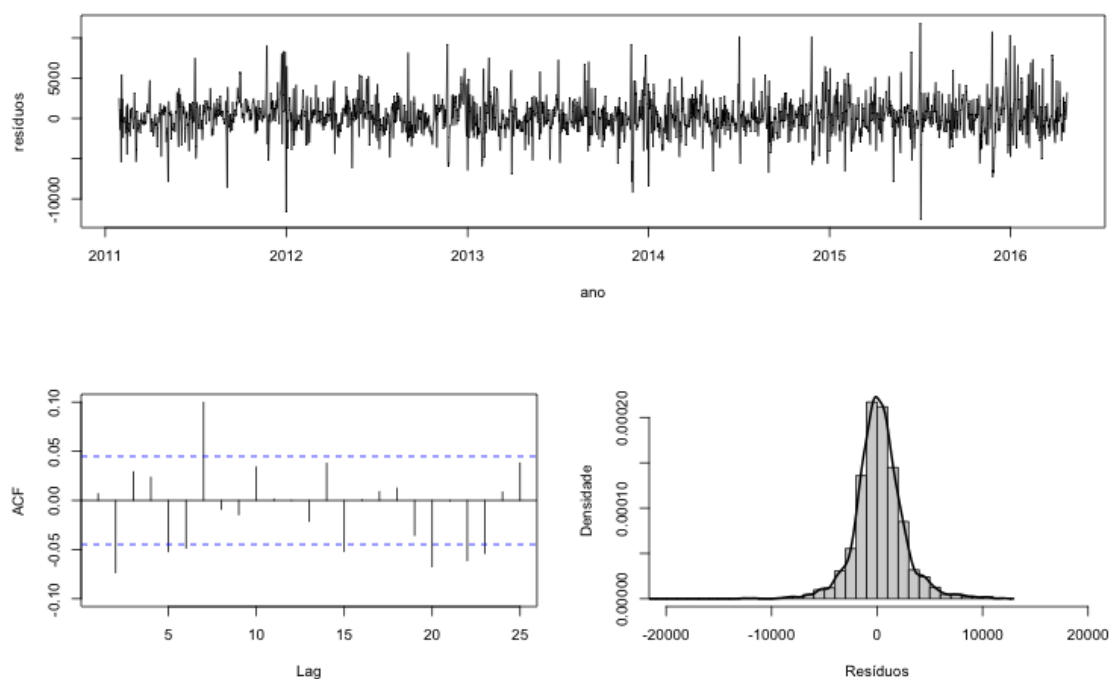


Figura 5.1: Análise residual do modelo TBATS

A Figura 5.8 compara os valores da série no mais alto nível de hierarquia (total de vendas) entre os dias 1.914 até 1.941 e os valores preditos pelo modelo TBATS. Avaliando o poder de

predição graficamente, fica evidente que entre os primeiros 10 dias os valores previstos pelo modelo são bem semelhantes às quantidades reais, porém para os próximos dias as previsões se descolam mais, principalmente entre os dias 1.930 e 1.935. Traduzindo para métricas, a Tabela 5.1 mostra o erro quadrático médio igual a 3.189,8, o seu erro absoluto médio foi de 2.464,4 e o erro percentual absoluto médio ficou em 5,31%. Tanto o RMSE quanto o MAE penalizam erros em escalas maiores, por isso quanto menor for o nível de hierarquia menores são os seus valores, mas em termos percentuais, fica claro pelo MAPE que as discrepâncias nos níveis de lojas e departamentos foram superiores em relação ao total.

Tabela 5.1: Medidas de erro do modelo TBATS conforme hierarquia

Nível de Hierarquia	RMSE	MAE	MAPE
Total	3.189,8	2.464,4	0,053
Estados	1.714,5	1.233,2	0,084
Loja	854,9	626,7	0,13
Categoria	1.380,4	901,5	0,07
Departamento	827,5	564,8	0,12
Produto	2,64	1,21	-

Por não possuir os valores verdadeiros para os dias 1.942 até 1.969, a única maneira de avaliar o modelo é por meio da plataforma *Kaggle* em que enviamos as estimativas a nível de produto para os períodos de avaliação e nos é retornado o valor do WRMSSE. Para o TBATS modelado desagregando por *top-down* encontramos um WRMSSE para o primeiro intervalo de 0,948 e para o segundo o valor foi igual a 0,894 (Tabela 5.2).

Tabela 5.2: Medida WRMSSE do modelo TBATS conforme o intervalo

Intervalo	WRMSSE
Score Público ($d_{1914} - d_{1941}$)	0,948
Score Privado ($d_{1942} - d_{1969}$)	0,894

5.2 Modelo LGBM sem a variável estatística composta pelo TBATS

O processo para treinar o modelo LGBM iniciou-se na construção do banco de dados de modelagem. A competição fornecia três principais fontes de dados: quantidade de vendas por produto entre 2011 e 2016, informações de calendário e preço de prateleira dos produtos. Ao realizar o cruzamento entre as bases obtemos as características de hierarquia de cada produto, as informações relacionadas às datas, a quantidade vendida e seu determinado preço. Em conjunto a esses dados construímos variáveis contendo *lags* e médias móveis das quantidades vendidas, informações adicionais de calendário e de preços históricos. A relação das variáveis criadas para o treinamento do modelo, com exceção da variável estatística, se encontra na Tabela 3.1.

Após realizar as transformações citadas tornamos o banco de dados em uma dimensão de cerca de 47 milhões de linhas, em que cada produto aparecia 1.941 vezes variando apenas a quantidade de produtos vendida por dia, e 80 colunas criadas para que possamos treinar o modelo. Assim, construímos um ambiente LGBM para ser treinado no nível mais baixo da hierarquia e utilizar uma abordagem *bottom-up* para compilação dos resultados. O LGBM possuía um parâmetro limite de 7.500 *rounds*, um *early stopping* de 200 rodadas e a métrica de avaliação utilizada foi o RMSE para encontrar o melhor modelo. Todo esse processo foi executado pelo R com auxílio do pacote *lightgbm* (Wang, Zhang e Zhao, 2017).

Entre os processos de preparação do banco, exclusão de variáveis que possuíam valores unitários ou com baixa variância, calibração dos parâmetros do LGBM e treinamento do modelo o R conseguiu executar essa tarefa em cerca de 7,5 horas (Utilizando uma máquina *Apple MacBook Pro* com processador M1 e uma memória RAM de 16GB). O modelo obteve um RMSE de 2,15 na sua 7500^a rodada de execução do algoritmo, mas a tendência da curva apresentada na Figura 5.2 sugere que esse número possa cair ainda mais.

Analisando a Figura 5.8 é possível analisar que, semelhantemente ao primeiro modelo citado, os 5 primeiros dias de previsão estão muito condizentes com os valores preditos. Mas, ao contrário do que foi visto para o TBATS, as previsões entre os dias 1.920 e 1.930 ficaram

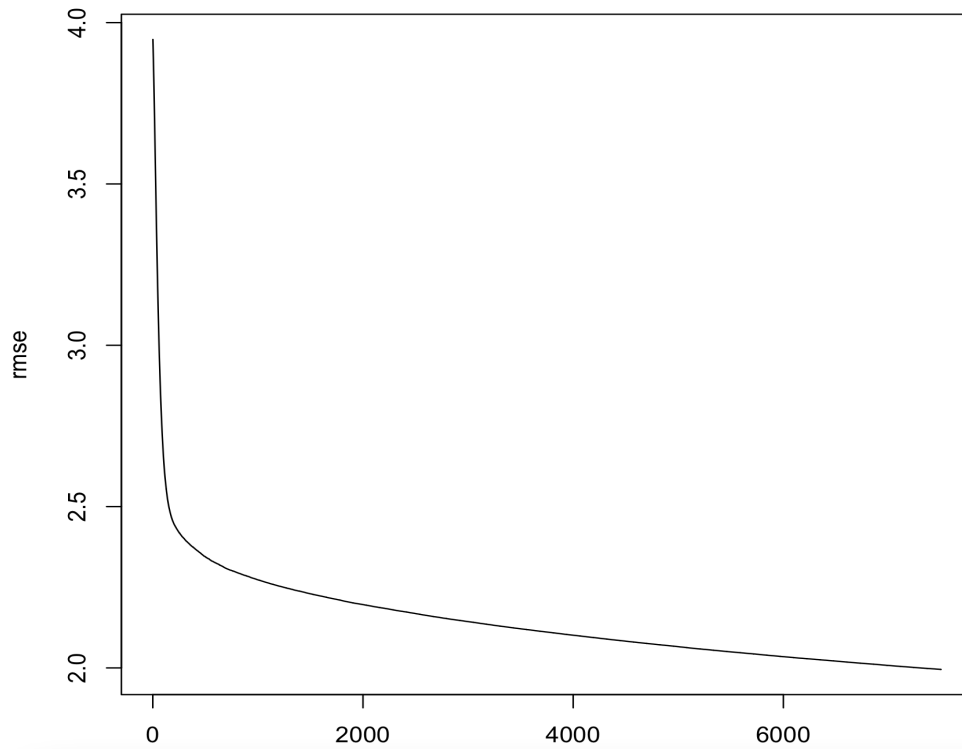


Figura 5.2: Curva de aprendizado do LGBM treinado sem a variável estatística

mais distantes dos totais originais, mas o modelo LGBM conseguiu diminuir a diferença vista no pico dos valores encontrados entre os dias 1.930 e 1.935. As três métricas encontradas para o modelo LGBM foram superiores na hierarquia total em comparação ao TBATS, mas para os demais níveis foram encontrados valores inferiores. Em termos percentuais, todos os valores da métrica MAPE, presentes na Tabela 5.3, ficaram abaixo dos 10% e nenhuma delas se sobressaiu como foi visto no TBATS.

Tabela 5.3: Medidas de erro do modelo LGBM conforme Hierarquia

Nível de Hierarquia	RMSE	MAE	MAPE
Total	3.528,51	3.079,06	0,068
Estados	1.385,08	1.115,74	0,073
Loja	514,05	382,64	0,082
Categoria	1.757,5	1905,1	0,06
Departamento	834,58	499,76	0,083
Produto	2,15	1,05	-

O WRMSSE obtido para o score público foi de 0,725 e para o score privado foi igual a 0,713, observados na Tabela 5.4. Isso significou uma melhora de mais de 20% na métrica de avaliação para ambos os *scores* em comparação com o modelo anterior.

Tabela 5.4: Medida WRMSSE do modelo LGBM conforme o intervalo

Intervalo	WRMSSE
Score Público ($d_{1914} - d_{1941}$)	0,725
Score Privado ($d_{1942} - d_{1969}$)	0,713

5.3 Modelos LGBM com a variável estatística composta pelo TBATS

O LGBM treinado para esta seção partiu dos mesmos parâmetros, intervalos temporais e metodologia de construção de variáveis do modelo anterior, mas com a adição da variável estatística construída a partir do TBATS. O processo de construção da variável para um modelo TBATS variou entre três cenários: i) do dia 1 ao 365; ii) entre os dias 1 e 750 e iii) do dia 1 ao dia 1.500. Ilustrando o procedimento para um primeiro ajuste do TBATS à série de vendas gerais, obtemos a variável h_1 da seguinte forma:

1. Suponha um primeiro intervalo de ajuste do TBATS para os primeiros 750 dias. É gerado de um modelo TBATS entre os dias 1 e 750 para o maior nível de hierarquia
2. Previsão de um passo à frente (nesse caso para o dia 751) para o total de quantidade vendidas e preenchimento da variável h_1 para o dia 751.
3. Utilização do modelo obtido em (1), com os valores reais dos dias 1 a 751, para prever o total vendido no dia 752 e, assim por diante, até o dia 1.913, que era o limite para o período de treinamento. Adotamos uma atualização dos parâmetros do TBATS a cada 30 dias de previsão, ou seja, um novo modelo TBATS era estimado conforme realizadas 30 novas previsões. Para este exemplo, um novo modelo TBATS foi estimado para os dias 1 a 780, ao gerar a previsão do dia 781.

4. Para a janela de validação (dia 1.914 até 1.941), consideramos que não temos os valores reais para compor a série utilizada pelo TBATS para gerar previsões durante esse intervalo. Dessa forma, alimentamos a série do total de vendas com a previsão gerada pelo modelo LGBM, como por exemplo: o valor de h_1 para o dia 1.914 é realizado ainda utilizando a previsão da série original com valores verdadeiros até o dia 1.913. Já para o dia 1.915 utilizamos o modelo LGBM treinado para gerar uma estimativa somente para o dia 1.914 e adicionamos esse valor à série de vendas totais utilizada no TBATS para gerar o valor de h_1 para o dia 1.915 e seguir esses passos até o dia 1.941
5. Por fim, repete-se o processo para a janela de teste (dia 1.942 até 1.969).

Ao final do processo de elaboração da variável, o banco de dados teve uma dimensão de iguais 47 milhões de linhas (registro de vendas por produto) e 81 colunas criadas (features adotadas) para o treinamento do modelo. Os parâmetros adotados para treinar o LGBM permaneceram em 7.500 rounds, 200 rodadas de *early stopping* e o RMSE como métrica de avaliação de performance durante o treinamento. Todo esse procedimento leva em torno de 8 horas para ser executado, um adicional de 30 minutos em relação à modelagem passada, devido ao desenvolvimento da variável estatística.

5.3.1 LGBM com primeiro ajuste do TBATS até o dia 750

O LGBM obteve um RMSE de 1,99 dentro do conjunto de treinamento e chegou a esse valor na 7.500^a interação, assim como o primeiro LGBM apresentado. Pela Figura 5.3 fica evidente que esse valor pode abaixar, mas não drasticamente de acordo com o decaimento lento a partir da 5000^a interação.

Avaliando a Figura 5.8, que compara os valores originais aos valores preditos, durante a janela de validação, vemos uma maior aderência nos 10 primeiros dias. As maiores distorções ficaram entre os dias 1.925 e 1.935 com um maior pico encontrado no dia 1.934, assim como foi visto para os outros dois modelos analisados. Analisando as três métricas na Tabela 5.5,

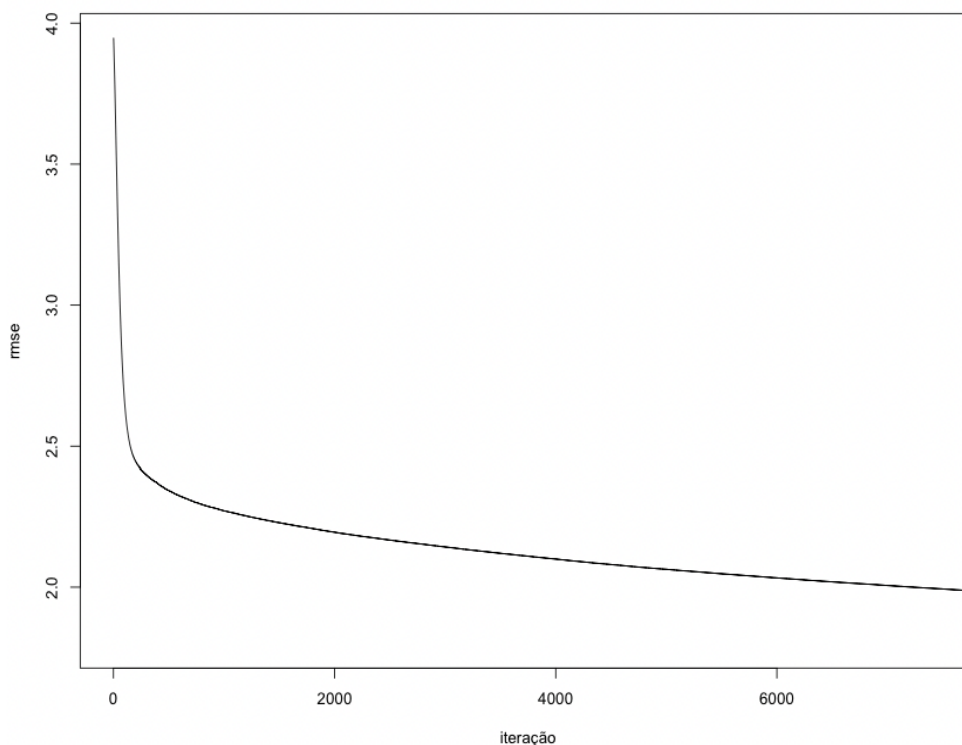


Figura 5.3: Curva de aprendizado do LGBM com primeiro ajuste do TBATS até o dia 750

vemos que esse modelo obteve os menores valores em todas as medidas e hierarquias vistas. Para a hierarquia geral encontramos um RMSE de 2.282,68, o seu MAE foi de 1.419,29 e o erro percentual absoluto médio ficou em 3,03%, enquanto que a nível de produto essas medidas não ficaram distantes do modelo anterior.

Tabela 5.5: Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 750 conforme hierarquia

Nível de Hierarquia	RMSE	MAE	MAPE
Total	2.282,68	1.419,29	0,0303
Estados	1.057,92	795,15	0,053
Loja	437,80	316,44	0,068
Categoria	1.141,98	667,16	0,054
Departamento	611,27	364,26	0,070
Produto	1,99	1,02	-

Para este modelo foram encontrados WRMSSE de 0,645 e 0,666 nos *scores* público e pri-

§5.3. Modelos LGBM com a variável estatística composta pelo TBATS

vado, conforme visto na Tabela 5.6, o que representa uma melhora de 10% e 6,5% em relação ao LGBM anterior. Chama a atenção o fato de que, diferentemente dos outros modelos, o valor da métrica utilizada na competição aumentou em 0,02 do *score* público para o privado, enquanto que, para o TBATS, a redução foi de 0,05 e para o outro LGBM foi de 0,01.

Tabela 5.6: Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 750 conforme o intervalo

Intervalo	WRMSSE
Score Público ($d_{1914} - d_{1941}$)	0,645
Score Privado ($d_{1942} - d_{1969}$)	0,666

Além do impacto visto nas reduções das métricas, o efeito da variável associada ao TBATS (h_1) é visto no Gráfico 5.4 que se refere à importância das variáveis presentes no modelo. Essa importância é obtida pela ponderação da probabilidade da árvore atingir um nó (ponto de partição). A probabilidade do nó pode ser calculada pelo número de amostras que chegam ao nó, dividido pelo número total de amostras utilizadas durante o processo de treinamento. Quanto maior o valor, mais relevante é a variável durante a composição das estimativas de produtos vendidos por dia.

Entre as 81 variáveis utilizadas para o treinamento, ela foi a 19ª mais relevante dentro do mecanismo do LGBM. Entre as mais importantes, vemos que as variáveis criadas a nível de produto, como aquelas com padrão 'rolling_mean_X_Y', as quais representam a média móvel nos últimos X dias para o lag de Y dias acerca da quantidade de produtos vendidos, preço do produto e a semana do ano se sobressaíram em relação às demais. Mas vale ressaltar que variáveis que aplicam o mesmo conceito, mas em períodos diferentes ficaram abaixo de h_1 . Mesmo sendo uma variável que representa uma previsão de uma série no nível total e o LGBM ser aplicado a nível de produto foi possível perceber o ganho que ela trouxe ao modelo.

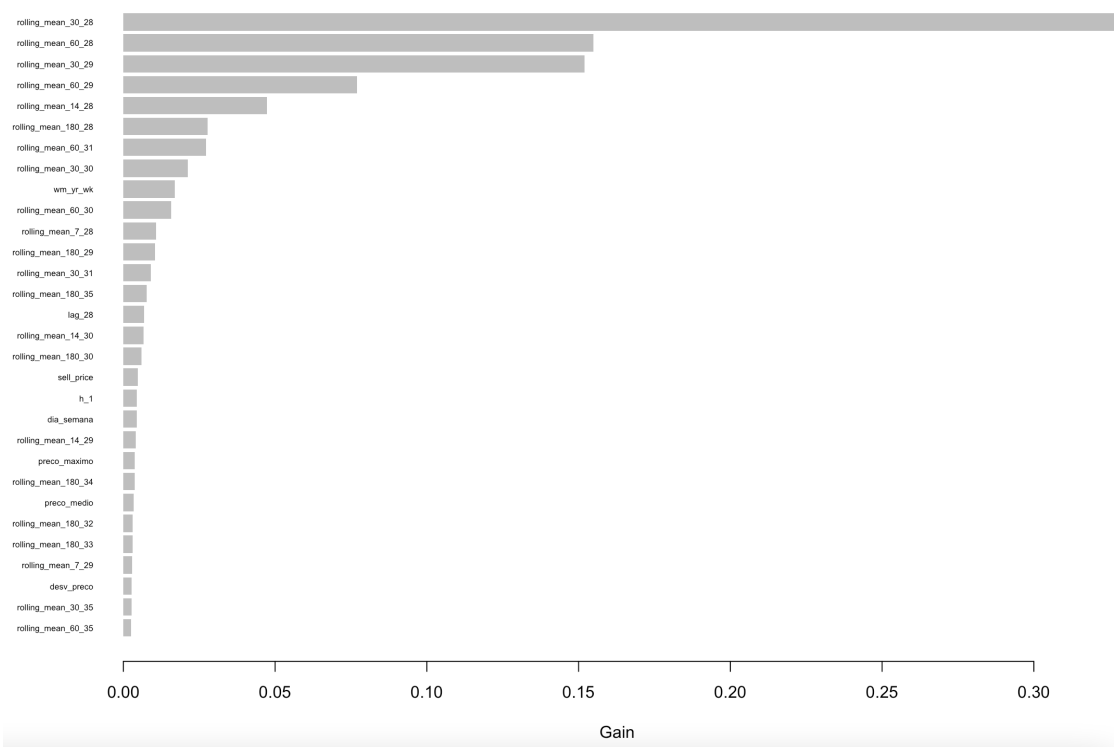


Figura 5.4: Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 750

5.3.2 LGBM com primeiro ajuste do TBATS até o dia 365

Modificando a janela do primeiro ajuste do TBATS para os primeiros 365 dias do período de treinamento como base da construção da variável estatística, o modelo LGBM obteve um RMSE de 2,15 no conjunto de treinamento e esse valor foi encontrado na 7.500ª interação. Pela curva de aprendizado na Figura 5.5 fica perceptível o decréscimo moderado pelas últimas rodadas do modelo, o que não sugere um decaimento maior já visto durante o processo.

A Figura 5.8 mostra um bom ajuste do modelo entre os primeiros 15 dias, mas encontra-se uma discrepância superior entre os dias 1.920 e 1.925 em comparação ao modelo anterior. A maior discrepância também se encontra em torno dos dias 1.930 e 1.935. Um fato que chama a atenção ao analisar as métricas de performance é que, nos níveis de hierarquia Total, Estados, Loja e Produto, este modelo obteve valores superiores aos resultados descritos na Tabela 5.5,

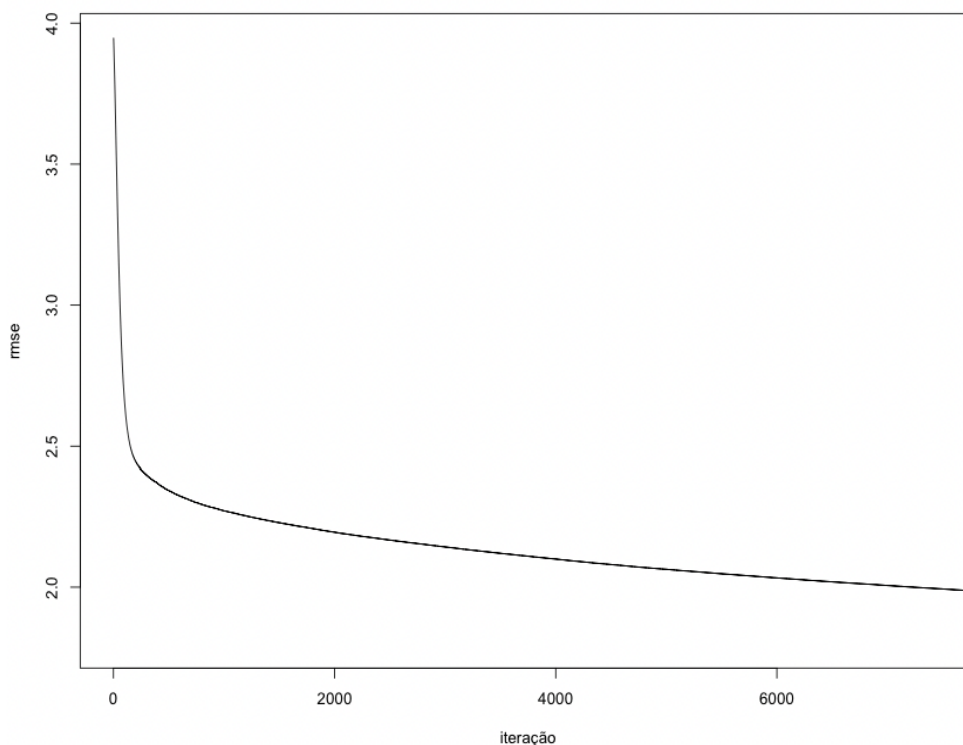


Figura 5.5: Curva de aprendizado do LGBM treinado com a variável estatística até o dia 365

enquanto que, para a hierarquia de Categoria e Departamento, os resultados das métricas do RMSE e MAE foram inferiores.

Tabela 5.7: Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 365 conforme Hierarquia

Nível de Hierarquia	RMSE	MAE	MAPE
Total	2.330,96	1.616,80	0,0356
Estados	1.066,83	807,24	0,0547
Loja	438,95	327,66	0,072
Categoria	1.071,32	678,06	0,055
Departamento	601,81	372,00	0,0719
Produto	2,15	1,06	-

Em relação ao WRMSSE, métrica utilizada na competição, as escalas foram muito semelhantes, com suas diferenças sendo apenas no *score* privado, com um valor superior a 0,001 em relação ao modelo anterior, conforme a Tabela 5.8.

Tabela 5.8: Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 365 conforme o intervalo

Intervalo	WRMSSE
Score Público ($d_{1914} - d_{1941}$)	0,645
Score Privado ($d_{1942} - d_{1969}$)	0,667

Em termos de importância para o modelo LGBM, a variável h_1 foi a 26^a entre as 81 utilizadas para treinar o modelo, conforme a Figura 5.6. Uma queda em relação ao modelo anterior, mas ainda assim com maior poder de predição em comparação a outras que foram construídas a nível de produto.

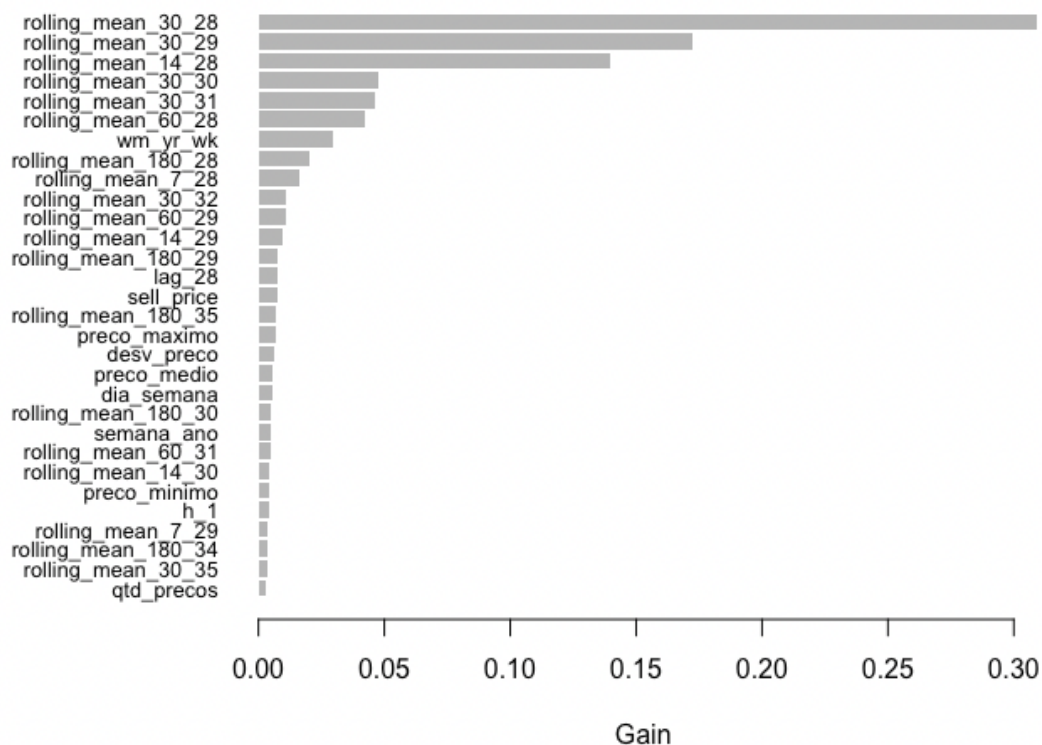


Figura 5.6: Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 365

5.3.3 LGBM com primeiro ajuste do TBATS até o dia 1.500

Aumentando a janela do primeiro ajuste do TBATS para 1.500 dias dentro do período de treinamento obtém-se um RMSE de 2,16 durante o período de treinamento na 7.500ª interação. Sua curva de aprendizado foi bastante semelhante às duas apresentadas nos modelos dessa classe, ou seja, o decréscimo da curva de aprendizado aparentava não diminuir tão bruscamente. Pela Figura 5.8 avaliamos que a principal diferença nas estimativas se encontram entre os dias 1.920 a 1.930, em que as estimativas se encontram abaixo dos valores originais da série, enquanto que nos outros dois modelos encontrava-se a previsão acima da linha real de quantidades vendidas.

Essa discrepância vista na Figura 5.8 fica evidenciada em termos das medidas de erro apresentadas. Foi visto que os valores para todas as métricas desse modelo ultrapassaram as medidas encontradas para os outros modelos contendo a variável h_1 .

Tabela 5.9: Medidas de erro do modelo LGBM com primeiro ajuste do TBATS até o dia 1.500 conforme Hierarquia

Nível de Hierarquia	RMSE	MAE	MAPE
Total	2.851,72	2.256,69	0,0488
Estados	1.201,65	904,69	0,0596
Loja	464,55	328,73	0,070
Categoria	1.490,74	958,18	0,059
Departamento	745,59	455,61	0,0808
Produto	2,16	1,05	-

O *score* público, o qual retrata as comparações realizadas acima, condiz com a Tabela 5.9 e Figura 5.8, sendo o maior WRMSSE visto entre os três modelos dessa modalidade testada. Mas, ainda assim, obteve um valor abaixo do primeiro LGBM treinado sem a variável h_1 e do TBATS. Porém, o que chama a atenção é o fato que esse modelo seria o vencedor numa possível competição com os outros quatro avaliados por obter o menor WRMSSE, conforme a Tabela 5.10.

Quanto à sua relevância, a variável h_1 foi a 18ª mais importante para o modelo LGBM, dentro das 81 colunas utilizadas para treinamento do modelo vista na Figura 5.7.

Tabela 5.10: Medida WRMSSE do modelo LGBM com primeiro ajuste do TBATS até o dia 1.500 conforme o intervalo

Intervalo	WRMSSE
Score Público ($d_{1914} - d_{1941}$)	0,681
Score Privado ($d_{1942} - d_{1969}$)	0,648

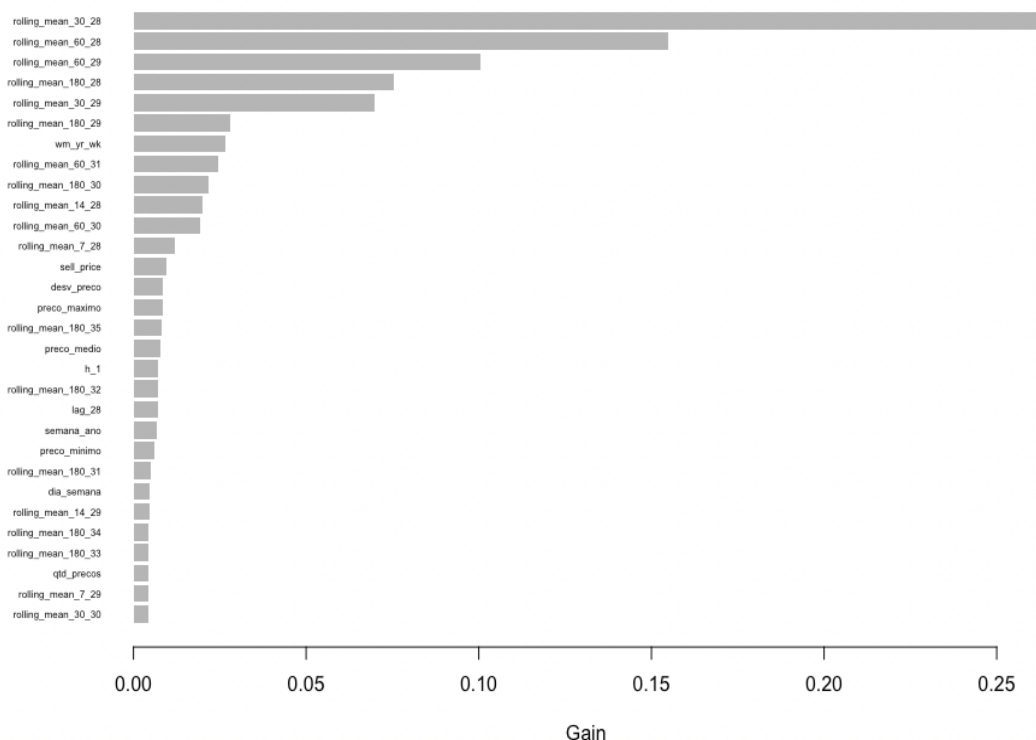


Figura 5.7: Variáveis mais relevantes para o modelo LGBM com primeiro ajuste do TBATS até o dia 1.500

5.4 Comparação de Resultados

Diversas outras técnicas e abordagens poderiam ser testadas para esse problema, mas por meio dos cinco modelos foi possível mesclar a teoria estatística para solução de problemas envolvendo séries temporais de múltipla sazonalidade, soluções de *Machine Learning* e dois formatos de combinação de resultados para previsões de séries hierárquicas.

Com base nas informações que são fornecidas pela competição, podemos comparar as previsões de cada modelo nos primeiros 28 dias após o período de treinamento (dia 1.914 até dia

1.941). A Figura 5.8 apresenta todas as previsões durante esse período. Pelo gráfico fica evidente que as previsões dos modelos se assemelham durante os 10 primeiros dias, mas durante os dias restantes as linhas dos modelos contendo a variável TBATS no LGBM tendem a estar abaixo da linha da série original.

O dia 1.934, presente na Figura 5.8, chamou a atenção devido ao seu grande pico de unidades vendidas e por todos os modelos subestimarem suas previsões. Já no dia 1.927 ocorreu o contrário, todos os modelos superestimaram suas previsões em valores máximos durante a série.

Pelos resultados apresentados, o poder de acurácia dos modelos LGBM se sobressai com folga em relação ao TBATS utilizado neste trabalho. Em relação ao WRMSSE, a métrica utilizada na competição, o ganho pela utilização do LGBM em relação ao TBATS foi de 0,894 para 0,648, o que representa uma diminuição de mais de 27% na medida válida para avaliação dos modelos. O poder da variável h_1 dentro do LGBM também foi significativo ao encurtar o WRMSSE de 0,713 para 0,648, significando um decréscimo de 9,1%. A Tabela 5.11 ilustra um comparativo entre todos os modelos executados.

Tabela 5.11: Métricas do WRMSSE para todos os modelos treinados

Modelo	WRMSSE Público	WRMSSE Privado
TBATS	0,948	0,894
LGBM	0,725	0,713
LGBM + TBATS (365)	0,645	0,667
LGBM + TBATS (750)	0,645	0,666
LGBM + TBATS (1.500)	0,681	0,648

Entre os cinco modelos apresentados, aquele que seria determinado como o vencedor seria o LGBM + TBATS (1.500) por possuir o menos WRMSSE durante o período de envio de previsões privadas para o site. Este WRMSSE alcançaria a 318^a colocação, ficando entre os top 6% competidores.

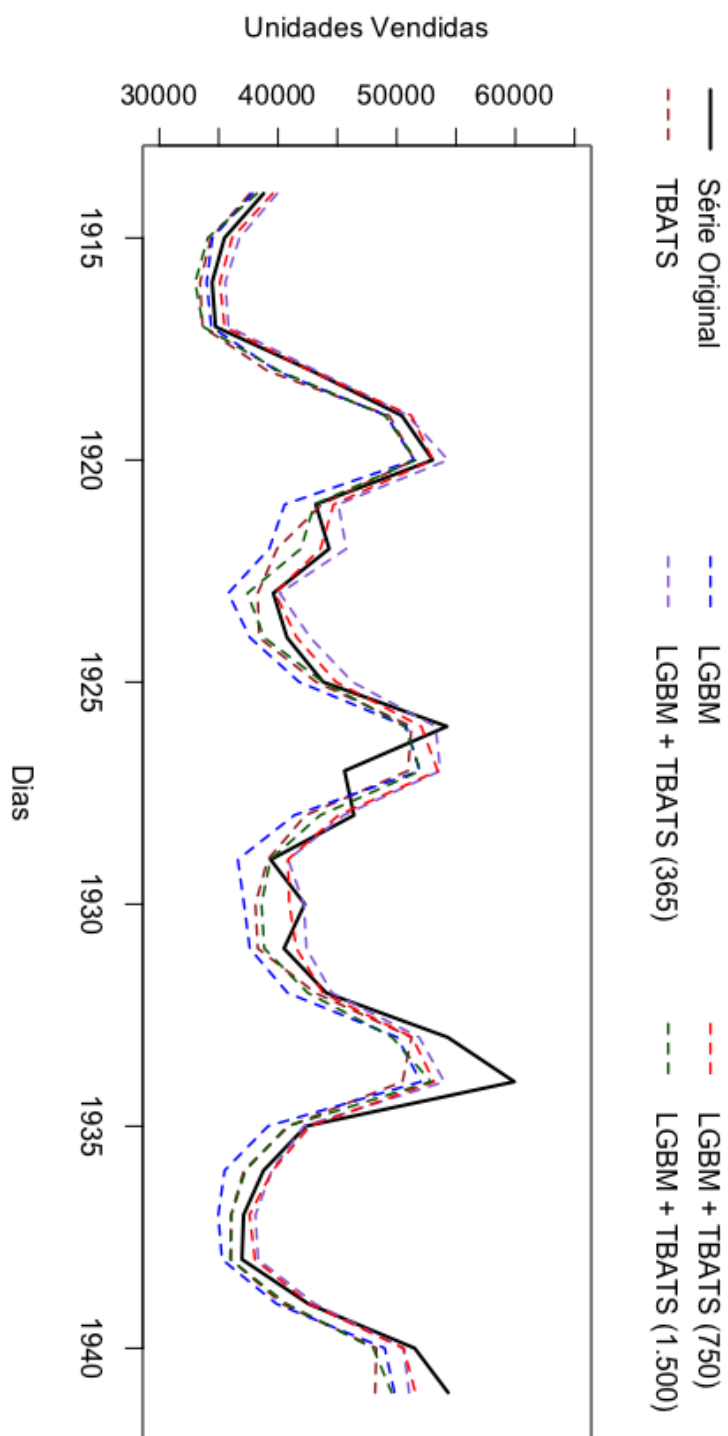


Figura 5.8: Comparativo entre previsões durante o período de validação da competição

Capítulo 6

Conclusão

Durante este trabalho analisamos o contexto teórico envolvendo problemas de séries temporais hierárquicas, passando pelos temas de agregação/desagregação de resultados, sazonalidades múltiplas ou complexas, modelos apropriados para o problema, solução envolvendo grandes massas de dados e a construção de variáveis que ajudem a aumentar a performance de modelos de aprendizado de máquina. Utilizamos como base do trabalho os dados da competição *M5-Forecasting*, que possuem uma característica hierárquica e uma larga escala de informações de produtos atribuídos a alguns mercados nos Estados Unidos.

Quando iniciamos este trabalho tivemos como benefício já possuir informações da solução vencedora e quais metodologias estavam se sobressaindo dentro da competição. Sendo assim, foi visto que as soluções que utilizassem o LGBM geralmente possuíam menor tempo de processamento e melhor acurácia nas previsões, o que nos guiou para utilização desse algoritmo. Outro ponto relevante nas discussões eram sobre os períodos de treinamento e validação dos modelos, pois não havia um consenso sobre a utilização das janelas mais recentes ou aproveitamento da série como um todo, pois a solução vencedora não fez proveito de todo o período temporal da série, enquanto as soluções seguintes aproveitaram todos os 1.914 dias para realizar o seu modelo.

Mas, conforme a revisão bibliográfica realizada e análise das principais soluções dos pro-

blemas, percebeu-se uma lacuna na utilização de soluções que envolvessem metodologias estatísticas em modelos de *Machine Learning*. Dessa forma foi proposto neste trabalho um modelo híbrido capaz de adicionar colunas utilizando metodologias estatísticas para banco de dados utilizados em problemas de *Machine Learning*. Avaliando o processo proposto neste trabalho, fica evidente que ele é dependente de três fatores: Método Estatístico, escolha de hierarquia para gerar previsões e primeiro intervalo para ajuste do modelo.

Aqui, abre-se alguns pontos discutíveis: o primeiro deles sobre o nível de hierarquia utilizado para gerar previsões em h_1 , pois foi visto pelos resultados dos modelos que uma abordagem em *top-down* tende a favorecer os resultados em hierarquia mais alta, ou seja, previsão da demanda total ou a nível de estado (Tabela 5.1) e, em contrapartida, a metodologia *bottom-up* é propensa a favorecer os níveis mais baixos, ou seja, previsão de demanda a nível de produto, mas não podemos concluir que a criação da variável em outros níveis acarretaria em um resultado melhor. Outro ponto é quanto ao primeiro horizonte de ajuste do modelo; Aqui enfrenta-se um *trade-off* entre escolher um grande período de treinamento e ter pouco preenchimento da variável podendo acarretar numa importância baixa para o modelo de *Machine Learning* ou adotar um pequeno intervalo para o ajuste do modelo estatístico e possuir uma quantidade relevante de previsões durante o período de treinamento podendo produzir um efeito mais considerável na acurácia da variável resposta.

Quanto aos diferentes horizontes de primeiro ajuste do TBATS para construção da variável h_1 , tiramos algumas constatações a partir dos resultados: i) Quanto maior o primeiro período de ajuste, maior foi a importância para o LGBM treinado; ii) O fator importância não implicou em métricas de avaliação melhores, em comparação com os três modelos contendo essa variável, aquele que teve o primeiro ajuste do TBATS para os primeiros 750 dias obteve métricas de avaliação melhores de forma geral; iii) Porém, levando em conta o período que contava para a competição, o modelo LGBM com o TBATS ajustado para os primeiros 1.500 dias se sagraria o vencedor.

Em comparação com as outras soluções da competição, o melhor resultado encontrado neste

trabalho ocuparia a 318ª colocação numa competição com 5.500 soluções enviadas para a plataforma, ou seja, estaria entre os 6% melhores resultados. O grande diferencial do primeiro colocado foi por ajustar diversos modelos LGBM por suas hierarquias e suas respectivas combinações para no final obter as previsões.

Uma das limitações deste trabalho ficou por conta do poder computacional para gerar diversos modelos LGBM entre as hierarquias e explorar suas relações por meio da geração das árvores feitas pelo algoritmo, como foi o caso da solução vencedora. Outra limitação de extrema importância é relacionada ao processo de validação do modelo. Para este trabalho construímos e elaboramos a variável h_1 sob uma série com intervalos sazonais sequenciais. Caso decidíssemos realizar uma validação cruzada ou *out-of-time*, esse método esbarraria na limitação da geração de valores presentes em janelas que não possuímos os verdadeiros valores da série em questão.

Por outro lado, os resultados encontrados nos mostraram que a construção de um modelo híbrido, associando uma metodologia de *Machine Learning* com uma variável utilizando uma metodologia estatística, melhorou as métricas e o poder de previsão tanto do TBATS quanto do LGBM sem a coluna proposta. Outra realização deste trabalho foi enriquecer a literatura de séries temporais hierárquicas com o estudo aprofundado de um caso real com múltiplas sazonalidades e propor uma forma de construir variáveis baseadas em metodologias estatísticas.

Para pesquisas futuras sugere-se a aplicação da metodologia de criação de variáveis em outras técnicas de *Machine Learning* para avaliação de sua influência causada nas previsões. Outra sugestão de pesquisas futuras seria a adoção de modelos estatísticos diferentes conforme a hierarquia da informação. Para o caso em estudo, o TBATS não se ajustaria bem para os casos mais granulares e intermitentes, sendo mais indicada a utilização do modelo Croston (Croston, 1972).

Referências Bibliográficas

- Al Daoud, Essam (2019). “Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset”. *International Journal of Computer and Information Engineering* 13.1, pp. 6–10.
- Barandas, Marília et al. (2020). “TSFEL: Time series feature extraction library”. *SoftwareX* 11, p. 100456.
- Christ, Maximilian et al. (2018). “Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package)”. *Neurocomputing* 307, pp. 72–77.
- Croston, John D (1972). “Forecasting and stock control for intermittent demands”. *Journal of the Operational Research Society* 23.3, pp. 289–303.
- De Livera, Alysha M, Hyndman, Rob J e Snyder, Ralph D (2011). “Forecasting time series with complex seasonal patterns using exponential smoothing”. *Journal of the American statistical association* 106.496, pp. 1513–1527.
- Diggle, Peter (1990). *Time series: a biostatistical introduction*. Oxford University Press.
- Gross, Charles W e Sohl, Jeffrey E (1990). “Disaggregation methods to expedite product line forecasting”. *Journal of forecasting* 9.3, pp. 233–254.
- Hyndman, Rob et al. (2019). “tsfeatures: Time series feature extraction”. *R package version* 1.0.
- Hyndman, Rob J e Athanasopoulos, George (2018). *Forecasting: principles and practice*. OTexts.
- Hyndman, Rob J et al. (2011). “Optimal combination forecasts for hierarchical time series”. *Computational statistics & data analysis* 55.9, pp. 2579–2589.

- Hyndman, Rob J et al. (2020). “Package ‘forecast’”. *Online*] <https://cran.r-project.org/web/packages/forecast/forecast.pdf>.
- Hyndman, Rob J et al. (2022). “Package ‘fpp2’”.
- Ke, Guolin et al. (2017). “Lightgbm: A highly efficient gradient boosting decision tree”. *Advances in neural information processing systems* 30, pp. 3146–3154.
- LGBM (s.d.). *LightGBM documentation*. Disponível em: <https://lightgbm.readthedocs.io/en/latest/>. Acesso em: 03 novembro 2021.
- Makridakis, S, Spiliotis, E e Assimakopoulos, V (2020). “The M5 accuracy competition: Results, findings and conclusions”. *Int J Forecast*.
- Mancuso, Paolo, Piccialli, Veronica e Sudoso, Antonio M (2021). “A machine learning approach for forecasting hierarchical time series”. *Expert Systems with Applications* 182, p. 115102.
- Morettin, Pedro A e Toloí, Clélia (2006). “Análise de séries temporais”. Em: *Análise de séries temporais*, pp. 538–538.
- Nun, Isadora et al. (2015). “Fats: Feature analysis for time series”. *arXiv preprint arXiv:1506.00010*.
- Phumchusri, Naragain e Ungtrakul, Phoom (2020). “Hotel daily demand forecasting for high-frequency and complex seasonality data: a case study in Thailand”. *Journal of Revenue and Pricing Management* 19.1, pp. 8–25.
- Spiliotis, Evangelos et al. (2020). “Comparison of statistical and machine learning methods for daily SKU demand forecasting”. *Operational Research*, pp. 1–25.
- Wang, Dehua, Zhang, Yang e Zhao, Yi (2017). “LightGBM: an effective miRNA classification method in breast cancer patients”. *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, pp. 7–11.
- Wickramasuriya, Shanika L, Athanasopoulos, George, Hyndman, Rob J et al. (2015). “Forecasting hierarchical and grouped time series through trace minimization”. *Department of Econometrics and Business Statistics, Monash University* 105.