Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Dissertação de Mestrado

Aprendizagem cruzada para previsão de séries temporais univariadas

por

Matheus Gorito de Paula

Brasília, 01 de Outubro de 2022

Aprendizagem cruzada para previsão de séries temporais univariadas

por

Matheus Gorito de Paula

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

Orientador: Prof. Dr. José Augusto Fiorucci

Brasília, 01 de Outubro de 2022

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. Guilherme Souza Rodrigues

EST/UnB

Prof. Dr. Flávio Barboza

FAGEN/UFU

Não é preciso ter olhos abertos para ver o sol, nem é preciso ter ouvidos afiados para ouvir o trovão. Para ser vitorioso você precisa ver o que não está visível.

(A Arte da Guerra, Sun Tzu)

Este trabalho é dedicado à minha família.

Agradecimentos

Em primeiro lugar agradeço à minha família, que estiveram ao meu lado durante todo o decorrer do mestrado, incentivando e me apoiando em todos os momentos.

À minha esposa Ana Beatriz, pelo seu companheirismo e apoio aos meus estudos.

Aos amigos Bruno e Andreia, por todos momentos de diversão e descontração.

Ao meu orientador, Prof. Dr. José Augusto Fiorucci, por todos apoio e incentivo durante a realização deste trabalho.

Aos membros da banca, que se dispuseram de seu tempo e dedicação para ajudar na melhora deste trabalho.

A todos professores do PPGEST/UnB por todos os conhecimentos transmitidos nas disciplinas cursadas ao longo do mestrado.

A todos os amigos que de um jeito ou de outro, me incentivaram a superar as dificuldades encontradas durante a graduação.

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

Resumo

Aprendizado de máquina se refere ao processo pelo qual os computadores desenvolvem o reconhecimento de padrões, ou a capacidade de aprender continuamente, ou fazer previsões com base em dados, e então, fazer ajustes sem serem especificamente programados para isso. Dentro dos métodos de aprendizado de máquina, esse trabalho foca na técnica de *Stacking*. Competições de Previsões de Séries Temporais são competições que têm como objetivo avaliar e comparar a acurácia de modelos de previsão de Séries Temporais. Nesse projeto utiliza-se o banco de Séries Temporais da competição M3 para realizar previsões utilizando os modelos de referência de Séries Temporais. Após, treina-se um modelo de *Boosting* com os resultados das previsões buscando obter resultados mais eficientes nas competições.

Palavras-chave: Aprendizado de Máquina, Séries Temporais, *Boosting*, *Stacking*

Abstract

Machine learning refers to the process by which computers develop pattern recognition, or the ability to continually learn, or make predictions based on data, and then make adjustments without being specifically programmed to do so. Within machine learning methods, this work focuses on the Stacking technique. Time Series Forecast Competitions are competitions that aim to evaluate and compare the accuracy of Time Series forecast models. In this project we use the Time Series database from the M3 competition to make predictions using the Time Series reference models. Afterwards, we train a Boosting model with the results of the predictions seeking to obtain more efficient results in competitions.

Keywords: Machine Learning, Time Series, Boosting, Stacking

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A proposta deste trabalho é motivada pelo interesse em obter melhores previsões na competição M3, introduzindo a técnica de janela deslizante para criar um conjunto de dados com previsões das séries temporais e aplicar modelos de aprendizado de máquina nesse conjunto de dados de forma a obter previsões. Desta forma, o objetivo do trabalho é construir um modelo de *stacking*, isto é, combinar previsões de modelos tradicionais de séries temporais com modelos de aprendizado de máquina, de forma a obter resultados que apresentem desempenhos comparáveis as técnicas vencedoras da competição M3. A escolha da abordagem por janela deslizante está alinhada com os resultados apresentados em Makridakis e Hibon (2000), onde uma das conclusões do estudo indicam que os modelos com melhor performance na competição M3 dependem do horizonte das previsões. Além disso, Makridakis e Hibon (2000) também apresenta que modelos que são combinações de outros modelos tendem a performar melhor nos dados da competição. Essas conclusões apresentadas são usadas de motivação para combinarmos a técnica de janela deslizante para construção do conjunto de dados a ser aplicado no modelo de aprendizado de máquina.

Inicialmente, é necessário conhecer alguns conceitos importantes que foram utilizados como base desse trabalho. O Aprendizado de Máquina, por exemplo, é uma área de estudo que envolve os conhecimentos de estatística e computação que vem se tornando muito popular pelas

aplicações em diversas áreas do conhecimento. Dentre as aplicações de técnicas de aprendizado de máquina, temos o campo de previsão de séries temporais. Portanto, uma série temporal é uma sequência de números coletados em intervalos regulares durante um período de tempo. Dentre os estudos de séries temporais, abordaremos nesse trabalho as competições de previsão de séries temporais. Essas competições buscam avaliar e comparar a acurácia de diversos métodos de previsões de Séries Temporais. Nesse contexto, temos a competição M3 que é a terceira edição da competição de Makridakis (Makridakis e Hibon, 2000). Observa-se que uma conclusão comum das edições das competições de Makridakis é que a combinação de métodos de previsão costumam superar os métodos individuais, mesmo quando a combinação é feita utilizando uma média aritmética simples (Makridakis e Hibon, 2000).

Motivados por esses resultados, alguns métodos de combinação com pesos otimizados ganharam destaque nas competições mais recentes, como a M4 ocorrida em 2018 (Makridakis, Spiliotis e Assimakopoulos, 2020; Petropoulos e Makridakis, 2019). Além disso, trabalhos recentes como os de Jaganathan e Prakash (2020), Smyl (2020), Petropoulos e Svetunkov (2020) e Fiorucci e Louzada (2020) apresentam o uso de combinação de técnicas de predição para competição de previsão de séries temporais. Neste contexto, o emprego de aprendizado de máquina para otimizar a combinação ou gerar estruturas híbridas de métodos de previsão consiste em um tema explorado na literatura (Smyl, 2020; Montero-Manso et al., 2020).

Para iniciar este trabalho, foi necessário coletar dados da competição M3 que podem ser encontramos em Makridakis e Hibon (2000). Logo em seguida, foi utilizado a técnica de janela deslizante combinado com modelos clássicos de séries temporais para construir um conjunto de dados contendo resultados de previsões em diversos horizontes, tais como, previsão de um passo a frente, previsão de 2 passos a frente e assim sucessivamente até o máximo do horizonte da série temporal. Esse conjunto de dados foi utilizado para treinar um modelo de aprendizado de máquina. Esse processo que envolve a aplicação de duas previsões é o chamado *stacking*. Por fim, esses resultados foram comparados com a literatura para validar a proposta do trabalho.

O restante deste trabalho foi organizado da seguinte maneira: O Capítulo 2 apresenta uma

revisão de modelos utilizados em séries temporais e conceitos importantes de técnicas de aprendizado de máquina. Além disso, é apresentado alguns conceitos básicos, incluindo uma introdução aos conceitos de aprendizado de máquina e competições de séries temporais. No Capítulo 3, introduzimos as principais característica dos dados da competição M3 e apresentamos a metodologia utilizada para criação do banco de dados. O Capítulo 4 elucida os resultados obtidos nesse trabalho e comparamos com resultados da literatura. O Capítulo 5 apresentamos as considerações finais sobre o trabalho e indicamos trabalhos futuros.

1.1 Revisão Bibliográfica

Dietterich (2000) apresenta o conceito de *ensemble* de modelos e suas vantagens, mostrando como modelos que são combinados produzem resultados melhores que usados de forma individual.

A técnica de boosting consiste em treinar modelos sequencialmente para reduzir o erro associado e foi apresentada por Friedman (Friedman, Hastie e Tibshirani, 2000). Chen e Guestrin (2016) apresentam a técnica de XGBoost e introduzem o pacote *xgboost* na linguagem R que permite a utilização da técnica.

Ahmed et al. (2010), apresenta o resultado dos principais algoritmos de aprendizado de máquina aplicados em dados da competição M3. Montero-Manso et al. (2020) mostra que combinações de modelos podem gerar resultados melhores em competições de previsão de séries temporais.

Meade (2000) introduz o uso da metodologia ARARMA, que é uma metodologia proposta em Parzen como um procedimento geral de modelagem de séries temporais em alternativa ao modelo ARIMA apresentado por Box e Jenkins. Billah et al. (2006) apresenta três abordagens para aplicação de técnicas de suavização exponencial e aplica em dados da competição M3 para observar e comparar os comportamentos das séries da competição.

Makridakis, Spiliotis e Assimakopoulos (2018a) aborda a acurácia e eficiência computacio-

nal da aplicações de algoritmos de aprendizado de máquina em relação as técnicas frequentistas de previsão de séries temporais aplicadas no conjunto de dados da competição M3.

Fildes (2020) apresenta os benefícios que as competições de previsão de séries temporais geram para comunidade de previsão. Além de apresentar técnicas inovadoras que possuem aplicações mercadológicas.

Makridakis, Spiliotis e Assimakopoulos (2018b) apresentam os resultados da competição M4, destacando as três principais descobertas: 1 - Os modelos mais acurados são combinações dos modelos tradicionais de séries temporais, 2 - Entre os modelos mais acurados estavam os híbridos entre abordagens estatísticas e de aprendizado de máquina, 3 - O modelo que obteve a segunda melhora acurácia foi um híbrido entre sete modelos estatísticos e um modelo de aprendizado de máquina que calculava os pesos de cada modelo estatístico de forma a minimizar a previsão.

Gilliland (2020) reflete sobre o uso das técnicas de aprendizado de máquina nos resultados da competição M4 em relação com a competição M3 e discute sobre o impacto da utilização desses algoritmos nos resultados obtidos.

Barker (2020) apresenta os conceitos de modelos estruturados e não estruturados para facilitar a identificação das técnicas utilizadas nas competições de previsão de séries temporais. Além disso, é discutido as inovações que o modelo vencedor da competição M4 apresentou.

Grushka-Cockayne e Jose (2020) examinam os intervalos de previsões submetidos na competição M4, avaliando a calibragem e a performance dos intervalos em diferentes horizontes de previsão. O artigo traz reflexões sobre o uso de agregações de intervalos para melhora na calibragem e acurácia dos intervalos.

Fry e Brundage (2019) apresentam um resumo da competição M4 com os principais resultados e as inovações que a competição gerou na comunidade de previsão de séries temporais. Ademais, analisa as principais diferenças entre os dados da competição e problemas reais de previsão.

Capítulo 2

Modelos de Previsão e Aprendizado de Máquina

Esse capítulo apresenta os conceitos fundamentais que permeiam o trabalho. São apresentados os principais modelos de séries temporais utilizados em competições de previsão de séries temporais. Além disso, introduz os conceitos de aprendizado de máquina que foram aplicados nesse trabalho. Mais detalhes sobre os assuntos abordados podem ser encontrados em Heckman (2007), Morettin e Toloi (2006), Wei et al. (2006), Wooldridge (2013), Hyndman e Athanasopoulos (2018), Fiorucci et al. (2015), Fiorucci et al. (2016), Ke et al. (2017) e Daoud (2019), Friedman, Hastie e Tibshirani (2000) e Chen e Guestrin (2016).

2.1 Séries Temporais

Entende-se por Série Temporal um conjunto de observações sobre uma variável aleatória, ordenadas no tempo e registrado em períodos regulares, onde existe alguma forma de dependência entre os dados observados em tempos diferentes. Logo, dado n períodos observados, vamos nos referir a série temporal como uma sequência de variáveis aleatórias $y_1, y_2, \ldots, y_n$ observadas em tempos discretos $t = 1, 2, \ldots, n$ (Morettin e Toloi, 2006).

O estudo das propriedades estatísticas da série temporal buscando prever eventos futuros baseados no comportamento histórico da série é chamado de previsão. Além disso, denomina-se horizonte de previsão o intervalo de tempo que separa a última observação usada no ajuste do modelo de previsão e o valor futuro a ser previsto. Isto é,

Definição 2.1.1. Dado $y_1, y_2, \ldots, y_n$ observações de uma série temporal e h o horizonte de previsão. Assim a projeção futura dos pontos $y_{n+1}, y_{n+2}, \ldots, y_{n+h}$ é dada pelo seguinte valor esperado condicional:

Segue que $\hat{y}_{n+i} = E[y_{n+i}|y_1, y_2, \ldots, y_n]$, $i = 1, 2, \ldots, h$.

O conceito de previsão de série temporal é uma ferramenta estatística comum em diversas áreas do estudo, como por exemplo: Finanças, Marketing, Vendas, Competições entre outras.

Em particular, esse estudo está interessado em aplicações a dados de competição.

2.2 Naive

Hyndman e Athanasopoulos (2018) apresentam uma das formas mais simples de se fazer previsão de séries temporais, o método Naive. Isto é,

Definição 2.2.1. Dado $y_1, y_2, \ldots, y_t$ observações de uma série temporal Y_t e h o horizonte de previsão. O modelo de previsão Naive consiste em estimar o valor futuro considerando apenas a observação anterior

$$\hat{y}_{t+h|y_1,\ldots,y_t} = y_t \quad h = 1, 2, \ldots$$

2.3 ARMA

ARMA é acrônimo para *AutoRegressive Moving Average*. Ou seja, é a combinação de um modelo de Média Móvel(MA) e um modelo Autorregressivo (AR).

Definição 2.3.1. Dado uma série temporal $y_1, \ldots, y_t$ estacionária, um modelo autorregressivo $AR(p)$ e $MA(q)$ um modelo de médias móveis. Temos que

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{2.1}$$

Segue que a equação 2.1 é a representação matemática de um modelo $ARMA(p, q)$, onde p é a ordem da componente autorregressiva e q é a ordem da componente de médias móveis.

2.4 Modelo Arima

Arima é um acrônimo para *AutoRegressive Integrated Moving Average*. Onde *Integrated* se refere a componente de diferenciação da série.

O componente de diferenciação da série é introduzido para permitir que a série se torne estacionária. Logo, podemos reescrever as observações y_t como $y'_t = y_t - y_{t-1}$.

Definição 2.4.1. Dado uma série temporal $y_1, \ldots, y_t$ não estacionária. Podemos tomar a diferença dos compontes da série de forma que $y'_t = y_t - y_{t-1}$. Assim, temos que $y'_1, \ldots, y'_t$ é uma série estacionária. Seja $AR(p)$ um modelo autorregressivo e $MA(q)$ um modelo de médias móveis. Temos que

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{2.2}$$

Segue que a equação 2.2 é a representação matemática de um modelo $ARIMA(p, d, q)$, onde p é a ordem da componente autorregressiva, d é o grau de diferenciação necessária para fazer com que a série se torne estacionária e q é a ordem da componente de médias móveis.

Quando combinamos modelos AR e MA complexos, se torna mais útil utilizar o operador de diferenciação B.

Assim, aplicando o operador de diferenciação na equação 2.2, temos que

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\epsilon_t$$

A função de previsão pontual de um modelo ARIMA é dada por

$$y'_{t+h} = E[y'_{t+h}|y'_t, y'_{t-1}, \dots]$$

onde h é o horizonte de previsão.

2.5 Modelos de Suavização

A maioria dos modelos de previsão de Séries Temporais se baseia no princípio de que as observações passadas contêm informações sobre os valores futuros. Além disso, os modelos buscam encontrar os padrões de ruídos nas observações e utilizá-los para prever os valores futuros.

Os modelos de suavização surgem da necessidade de unir os dois princípios. Assim, as técnicas de suavização consideram que as observações de valores extremos das Séries são eventos aleatórios e suavizando-os, pode-se encontrar padrões para realizar as previsões.

As previsões dos modelos de suavização exponencial são médias ponderadas de observações do passado, onde os pesos decaiem exponencialmente. Em outras palavras, quanto mais recente a observação maior o peso associado a ela (Hyndman e Athanasopoulos, 2018).

2.5.1 Suavização Exponencial Simples (SES)

A SES é a técnica de suavização exponencial mais simples e é muito utilizada para Séries que não apresentam um comportamento sazonal ou com tendência. Sua principal característica é o comportamento exponencial do parâmetro de suavização.

Logo, a estimação pontual do modelo SES segue

$$\begin{aligned}\hat{y}_{t+h|0:t} &= \alpha y_t + (1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots \\ &= \alpha \sum_k (1-\alpha)^k y_{t-k} + (1-\alpha)^t y_0\end{aligned}$$

em que, $0 < \alpha < 1$ é o parâmetro suavização.

Assim, observa-se que o SES se comporta como uma média ponderada que aplica pesos

maiores para observações mais recentes.

Para mais detalhes sobre essa técnica consultar Hyndman e Athanasopoulos (2018).

2.5.2 Suavização Exponencial de Holt

A Suavização de Exponencial de Holt é uma extensão da SES para dados com tendência. A técnica consiste em três equações, sendo uma de previsão e duas de suavização

$$\begin{aligned} \text{Equação de Previsão} \quad & \hat{y}_{t+h|t} = l_t + hb_t \\ \text{Equação de Nível} \quad & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Equação de Tendência} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \end{aligned}$$

em que l_t é a estimação no nível da série no tempo t, b_t é a estimativa da tendência da série no tempo t, $0 < \alpha < 1$ é o parâmetro de suavização do nível e $0 < \beta^* < 1$ é o parâmetro de suavização da tendência.

As previsões geradas pelo método de Suavização de Holt possuem uma tendência que cresce ou decresce indefinidamente. Logo, solucionar esse comportamento, foi introduzido uma constante ϕ que amortece o comportamento da tendência no futuro. Assim, temos que

$$\begin{aligned} \text{Equação de Previsão} \quad & \hat{y}_{t+h|t} = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t \\ \text{Equação de Nível} \quad & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}) \\ \text{Equação de Tendência} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1} \end{aligned}$$

em que $0 < \phi < 1$ é o parâmetro de amortecimento.

Para mais detalhes sobre essa técnica, consultar Hyndman e Athanasopoulos (2018) e Morettin e Toloi (2006).

2.5.3 Suavização Exponencial Sazonal de Holt-Winters (HW)

Como visto nas seções anteriores, as técnicas de Suavização Exponencial de Holt resolvem o problema de Séries com Tendência. Contudo, se fez necessário estender a técnica para Séries com comportamento Sazonais.

Com isso, surgiu a técnica de Suavização Exponencial de Holt-Winters que compreende uma equação de previsão e três equações de suavização.

Devido a variação da natureza da componente sazonal, temos que a técnica de HW assume duas formas.

Holt-Winters Aditivo

O método aditivo é utilizado quando a componente sazonal tem um comportamento aproximadamente constante.

Ou seja, temos que

$$\begin{aligned}
\text{Equação de Previsão} \quad & \hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \\
\text{Equação de Nível} \quad & l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Equação de Tendência} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
\text{Equação de Sazonalidade} \quad & s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}$$

em que k é a parte inteira da divisão $\dfrac{(h-1)}{m}$, sendo m o tamanho do ciclo sazonal.

Holt-Winters Multiplicativo

O método multiplicativo é utilizado preferencialmente quando existem variações sazonais proporcionais por nível da Série.

De forma análoga ao método aditivo, temos que

$$\begin{aligned}
\text{Equação de Previsão} \quad & \hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)} \\
\text{Equação de Nível} \quad & l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Equação de Tendência} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
\text{Equação de Sazonalidade} \quad & s_t = \gamma\frac{y_t}{l_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m}
\end{aligned}$$

2.6 Método Theta

O método Theta foi apresentado por Assimakopoulos e Nikolopoulos (2000) e propõe uma decomposição da série temporal analisada baseada na sua curvatura local. A decomposição é feita através de um parâmetro theta (θ) aplicado na segunda diferença entre as datas.
As linhas de decomposição são chamadas de linhas de theta e são representadas por $Z_t(\theta)$. De forma matemática, temos que, dado uma série $y_1, \ldots, y_n$ segue que

$$Z_t(\theta) = \theta y_t + (1-\theta)(A_n - B_n t)$$

Onde $\theta \in \mathbb{R}$, y_t é a série temporal original e A_n e B_n são os estimadores de uma regressão linear sob $y_1, \ldots, y_n$ obtidos via mínimos quadrados.

A decomposição permite uma melhor entendimento que normalmente não pode ser adquirido pelos métodos tradicionais de decomposição.

Nesse trabalho, mencionaremos o modelo Theta como STheta.

2.6.1 Modelo Theta Otimizado

O Modelo Theta Otimizado (OTM) foi proposto por Fiorucci et al. (2015) como uma forma de generalizar o método Theta, buscando otimizar a seleção da segunda linha de theta. Em linhas gerais, temos que, dado uma série $y_1, \ldots, y_n$ e uma generalização da combinação das linhas de

theta dada por

$$X_t = \omega Z_t(\theta_1) + (1-\omega)Z_t(\theta_2),$$

em que $\omega = \dfrac{\theta_2 - 1}{\theta_2 - \theta_1}$.

Assim, podemos reescrever y_t da seguinte forma

$$y_t = \left(1 - \frac{1}{\theta}\right)(A_n - B_n t) + \frac{1}{\theta}Z(\theta), \quad t = 1, \ldots, n$$

Quando $\theta = 2$, o modelo OTM equivale ao modelo STheta. Tal efeito será utilizado nesse trabalho e será referido como modelo STM.

2.6.2 Modelo Theta Otimizado Dinâmico

O Modelo Theta Otimizado Dinâmico (DOTM) introduz modificações nas componentes A_n e B_n, considerando-as como funções dinâmicas ao invés de componentes com valores fixos.

Fiorucci et al. (2016) obteve o seguinte conjunto de equações dinâmicas para representar o DOTM:

$$\begin{aligned}
y_t &= \mu_t + \epsilon_t \\
\mu_t &= l_{t-1} + \left(1 - \frac{1}{\theta}\right)\left[(1-\alpha)^{t-1}A_{t-1} + \left(\frac{1-(1-\alpha)^t}{\alpha}\right)B_t\right] \\
l_t &= \alpha y_t + (1-\alpha)l_{t-1} \\
A_t &= \bar{y}_t - \frac{t+1}{2}B_t \\
B_t &= \frac{1}{t+1}\left[(t-2)B_{t-1} + \frac{6}{t}(y_t - \bar{y}_{t-1})\right] \\
\bar{y}_t &= \frac{1}{t}\left[(t-1)\bar{y}_{t-1} + Y_t\right],
\end{aligned}$$

onde $\alpha \in (0,1)$ é o parâmetro de suavização e l é uma função do parâmetro de nível e θ.

Quando $\theta = 2$, o modelo DOTM consistem em um abordagem com coeficientes dinâmicos para o STheta. Tal efeito será utilizado nesse trabalho e será referido como modelo DSTM.

2.7 Ensemble

Aprendizado de máquina é o estudo de criação de regras automáticas que permitem realizar previsões de novas observações baseadas no comportamento das observações passadas. Ou seja, o objetivo é criar uma regra que generalize o comportamento das observações, permitindo realizar previsões baseado em observações novas.

Estruturar tal regra é uma tarefa complicada. Contudo, definir regras mais simples que produzem previsões razoavelmente adequadas é uma tarefa mais simples.

Ensemble são métodos de aprendizado de máquina que combinam um conjunto de modelos base mais simples em busca de gerar um modelo que apresente melhor performance que os modelos base separadamente (Dietterich, 2000).

A previsões estimadas através dos modelos bases são combinadas utilizando técnicas estatísticas, tais como média, moda ou por método mais sofisticados que encontram relações entre os modelos bases e defini quais estimações utilizar em cada momento.

Segundo Cha Zhang (2012), as principais vantagens de se utilizar os métodos de *ensemble* são a performance e a robustez. A primeira se deve ao fato de um modelo resultado da combinação de modelos bases pode fazer melhor estimativas que os modelos bases sozinhos. A robustez se deve a diminuição da dispersão nas estimativas de previsões do modelo.

A forma de alcançar essas estimativas, segundo Cha Zhang (2012), se deve a redução da variância do erro das previsões adicionando viés.

É importante observar que existem várias formas de se aplicar a técnica de *ensemble*, *bagging*, *boosting* e *stacking*.

2.7.1 Bagging

A técnica de *bagging* consiste no processo de treinamento de um modelo base em subconjuntos diferentes dos dados de treinamento. Os subconjunto dos dados de treinamento são definidos via amostragem aleatória, permitindo que o modelo base seja ajustados utilizando conjuntos

distintos de treinamento. Em seguida, ele combina os resultados dos modelos treinados para gerar previsões.

Essa estratégia permite a redução da variância e minimiza o *overfitting*, que é nome dado ao fenômeno que ocorre quando o modelo se ajusta ao ruído dos dados de treinamento e não consegue generalizar o comportamento dos dados (Dietterich, 1995).

A Figura 2.1 ilustra o comportamento da técnica de *bagging*.

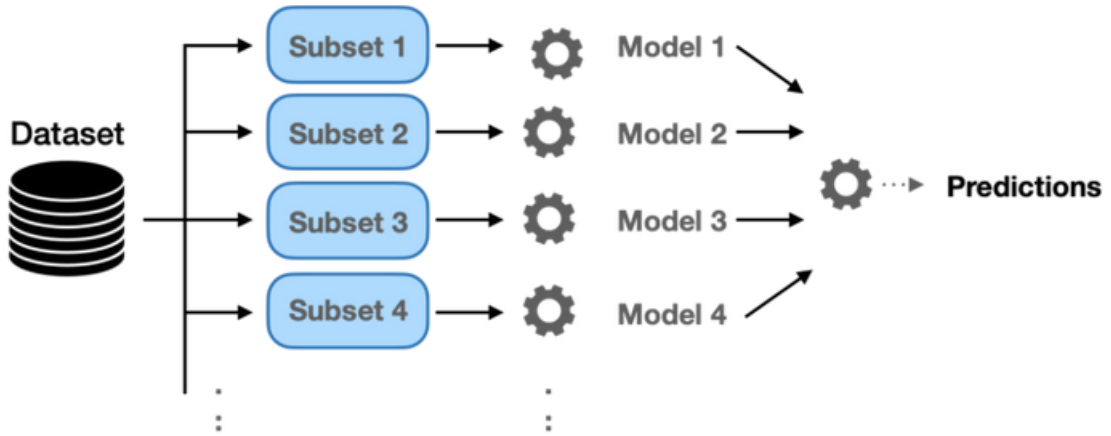

Figura 2.1: Funcionamento da técnica de Bagging.
(Freund e Schapire, 1997)

2.7.2 Boosting

Na técnica de *boosting*, cada modelo base é treinado com um conjunto de dados, de forma sequencial e de uma maneira adaptativa, onde um modelo base depende dos anteriores, e no final são combinados de uma maneira determinística (Schapire, 2003).

A principal diferença entre as técnicas de *bagging* e *boosting* são que a primeira os modelos bases são treinados em paralelo e na segundo temos um treinamento dos modelos base de forma sequencial.

A Figura 2.2 ilustra o funcionamento da técnica de *boosting*.

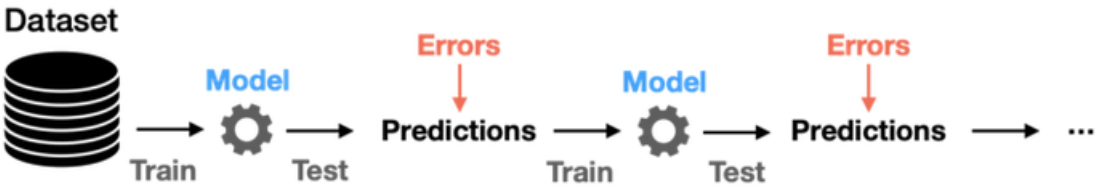

Figura 2.2: Funcionamento da técnica de Boosting.
(Freund e Schapire, 1997)

2.7.3 Stacking

Os processos que compõem a técnica de *stacking* são chamados de camadas. Onde a camada 0 contém os modelos base, e a camada 1 é o modelo que agrega as previsões dos modelos base.

De forma geral, as previsões dos modelos bases são utilizadas como insumo de um novo modelo que, por fim, encontra correlações entre essas previsões para obter uma estimativa melhor do comportamento dos dados (Cha Zhang, 2012).

A Figura 2.3 ilustra o funcionamento da técnica de *stacking*.

Figura 2.3: Funcionamento da técnica de Stacking.
(Freund e Schapire, 1997)

2.8 Validação Cruzada

Validação cruzada é uma técnica estatística de re-amostragem utilizada para avaliar e comparar modelos de aprendizado de máquina por meio da divisão do dados em dois segmentos: uma para treinar o modelo e outro para validá-lo (Berrar, 2018).

A técnica consiste em ajustar o modelo de forma iterativa, em diferentes conjuntos de treino e, em seguida validar o modelo treinado. É importante que os conjuntos de treino e validação se cruzem no processo de treinamento, uma vez que, todas as observações devem ser utilizadas para treinar e validar o modelo.

Para ilustrar o uso da técnica de validação cruzada, utilizaremos o *k-fold*. A técnica consiste em particiona os dados em k-conjuntos com tamanhos iguais, as chamadas *folds*. Em seguida, k-iterações de treinamento e validação são executas de tal forma que em cada iteração um conjunto é utilizado para validação e $k-1$-*folds* são utilizadas para treinar o modelo. A média da performance nas k-iterações é chamada de performance da validação cruzada.

De fato, considere $\hat{f}_{-k}$ o modelo que foi treinado em todos menos a k-ésima *fold*. Logo, $\hat{y}_i = \hat{f}_{-k}(x_i)$ é o valor estimado para y_i, e x_i é um elemento da k-ésima *fold*. Tomando, $\hat{\epsilon}_{cv}$ como o erro de estimação do processo de validação cruzada, segue que

$$\hat{\epsilon}_{cv} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}_{-k}(x_i))$$

A Figura 2.4 ilustra o comportamento da divisão *k-fold* de acordo com a iteração.

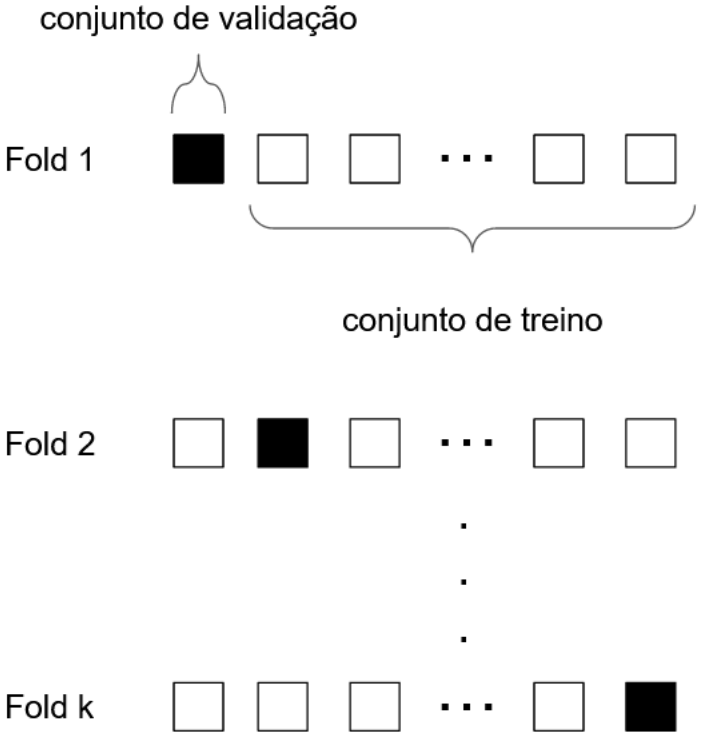

Figura 2.4: Divisão dos *folds*.

2.9 Gradient Boosting

O *Gradient Boosting* (GBM) é uma técnica de aprendizado de máquina, em particular de *boosting*, que produz um modelo de previsão a partir de modelos bases (Natekin e Knoll, 2013).

O funcionamento do GBM segue alguns passos. Primeiro, ajusta-se um modelo F_0 aos dados e obtém-se os resíduos $(y - F_0(x))$, onde y é a variável objetivo.

Em seguida, ajustamos um modelo h_1 aos resíduos. Então, combinamos o modelo F_0 e o modelo h_1 para gerarmos o novo modelo F_1

$$F_1 = F_0(x) + h_1(x)$$

logo, os erros associados ao modelo F_1 serão menores que do modelo F_0.

De forma análoga, temos para o modelo F_2

$$F_2 = F_1(x) + h_2(x)$$

Assim, segue que, as combinações dos modelos podem ser feitas m vezes até que os erros sejam minimizados.

$$F_m = F_{m-1}(x) + h_m(x)$$

Dado uma função de aprendizado $h(x, \theta)$ com uma função de custo dada por $\psi(y, f(x))$, o processo de estimar os parâmetros de forma direta se torna muito complexo. Logo, é necessário uma abordagem iterativa, onde a função de aprendizado é atualizada a cada iteração, $h(x, \theta_t)$.

O incremento em t é dado pela função

$$g_t(X) = E\left[\frac{\partial \psi(y, f(x))}{\partial f(x)} | X\right]_{f(x)=\tilde{f}^{t-1}(x)}$$

Essa abordagem permite que a otimização seja feita via método dos mínimos quadrados, o

que torna o problema mais simples de ser resolvido.

Definições mais profundas e entendimento dos algoritmos podem ser encontrados em Daoud (2019).

2.9.1 XGBoost

XGBoost é um aprimoramento do *Gradient Boosting* que permite que a técnica seja mais escalável e flexível. Além disso, tem se destacado em problemas de previsões Séries Temporais e de dados não-estruturados, como imagens e textos. Sendo uma das ferramentas mais utilizadas na Ciência de Dados.

A principal diferença entre o *XGBoost* e o *Gradient Boosting* é que uma nova função de regularização é introduzida para controlar o *overfitting*. Logo, a técnica se torna mais robusta durante o processo de treinamento do modelo (Daoud, 2019).

A regularização é feita através da adição de um termo na função de custo

$$L(f) = \sum_{i=1}^{m} L(\hat{y}_i, y_i) + \sum_{m=1}^{M} \Omega(\delta_m)$$

Onde $\Omega(\delta) = \alpha|\delta| + 0.5\beta||w||^2$, $|\delta|$ é o número de galhos da árvore, w é o valor de cada folha e Ω é a função de regularização.

Além disso, o *XGBoost* utiliza uma nova função de ganho, dada por

$$\begin{aligned} G_j &= \sum_{i \in I_j} g_i \\ H_j &= \sum_{i \in I_j} h_i \\ Gain &= \frac{1}{2}\left[\frac{G_L^2}{H_L + \beta} - \frac{G_R^2}{H_R + \beta} + \frac{(G_R + G_L)^2}{H_R + H_L + \beta}\right] - \alpha \end{aligned}$$

Onde $g_i = \partial_{\hat{y}_i} L(\hat{y}_i, y_i)$, $h_i = \partial^2_{\hat{y}_i} L(\hat{y}_i, y_i)$, G é o *score* da árvore à direita e H é o *score* da árvore à esquerda.

2.9.2 Light Gradient Boosting Machine

O *Light Gradient Boosting Machine* (LGBM) é modelo que utiliza a técnica de *boosting* e busca aumentar a eficiência computacional do modelo. O LGBM veio como alternativa ao *XGBoost*, pois apesar do *XGBoost* ser um modelo muito utilizado, ele apresenta problemas de eficiência computacional e de escalabilidade quando aplicado a um conjunto grande de dados. Uma das principais razões para esse problema é que para cada nova variável, é necessário recalcular todas as métricas para todas as possibilidades de divisão dos dados em conjuntos distintos (Daoud, 2019).

Para corrigir esse problema, Ke et al. (2017) introduziu as técnicas de *Gradient-based One-Side Sampling* (GOSS) e *Exclusive Feature Bundling* (EFB). A técnica do GOSS tem a função de reduzir a quantidade de divisão do conjunto dos dados para calcular as métricas utilizada para selecionar o melhor modelo. Enquanto a técnica EFB empacota variáveis que são mutuamente exclusivas, assim, reduzindo número de variáveis do modelo.

A utilização dessas técnicas permitiu que o *LGBM* se torna-se um método mais eficiente que o *XGBoost*.

De forma geral, como pode ser visto na Figura 2.5 a principal diferença é que o *LGBM* constrói as árvores folha por folha.

(a) LGBM. (b) XGBoost.

Figura 2.5: Diferença entre LGBM x XGBoost.

2.10 Métricas

Uma parte muito importante desse trabalho é a avaliação das métricas de desempenho dos modelos. A partir da análise das mesmas será possível escolher o melhor modelo de aprendizado

de máquina. Além disso, possibilitará comparação com os resultados de modelos aplicados na competição M3.

Nesse trabalho, veremos com mais clareza as métricas MAPE e sMAPE, pois são utilizadas na literatura de competição de previsão de séries temporais (Fiorucci et al., 2016; Fildes, 2020; Makridakis e Hibon, 2000).

2.10.1 MAPE

Segundo Kim e Kim (2016), Erro Absoluto Médio Percentual (MAPE) é uma medida estatística que avalia a acurácia da previsão. Sua utilização é recomendada na maioria dos livros texto, como Hanke e Wichern (2005) e Bowerman, O'Connell e Koehler (2005). Segundo de Myttenaere et al. (2016), MAPE é usualmente utilizado como uma função de perda em problemas de regressão e para avaliação de modelos. Ademais é a métrica mais utilizada nas competições de Makridakis, foco desse trabalho.

Seja, A_t e F_t o valor real e o valor predito, respectivamente, de uma variável no tempo t. A função MAPE é definida como

$$MAPE = \frac{1}{h}\sum_{t=1}^{h}\left|\frac{A_t - F_t}{A_t}\right|, \tag{2.3}$$

onde o horizonte de previsão é h. A rigor, é necessário multiplicar o resultado da equação 2.3 por 100, mas foi omitido sem perda de generalidade.

A função MAPE apresenta como vantagem sua fácil interpretabilidade em termos do erro relativo e como não depende de escala, se torna uma métrica comumente utilizada para comparar resultados de grandezas diferentes (de Myttenaere et al., 2016).

Apesar da função MAPE ter sido utilizada como métrica oficial nas competições M1 e M2, bem como para diversas outras aplicações, a função MAPE não é uma função simétrica e por conta disso, de acordo com teoria de espaços métricos, a MAPE não pode ser considerada matematicamente como métrica.

2.10.2 sMAPE

Para evitar as desvantagens da métrica MAPE, Makridakis e Hibon (2000) introduziu a métrica de Erro Absoluto Médio Percentual Simétrico (sMAPE).

A função sMAPE é definida como

$$SMAPE = \frac{1}{h}\sum_{t=1}^{h}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \tag{2.4}$$

onde A_t é o valor atual da observação t, F_t é o valor predito da observação t e h é o horizonte de previsão. A rigor, é comum multiplicar o resultado da equação 2.4 por 100, mas foi omitido sem perda de generalidade.

Capítulo 3

Metodologia

Neste capítulo, vamos apresentar o desafio que esse trabalho se propõe a resolver e a composição das séries temporais da competição M3. Além disso, vamos abordar o processo de construção do banco de dados utilizado no processo de modelagem.

3.1 Desafio

A proposta do trabalho se baseia nos resultados apresentados em Ahmed et al. (2010), Makridakis e Hibon (2000), Makridakis, Spiliotis e Assimakopoulos (2018a), Petropoulos e Svetunkov (2020) e Fildes (2020), que apresentam as vantagens do uso combinado de modelos tradicionais de séries temporais com modelos de aprendizado de máquina. Além dos trabalhos citados indicarem o benefício do uso combinado de modelos de séries temporais com modelos de aprendizado de máquina, esse trabalho propõem o uso da técnica de janela deslizante para auxiliar na construção do conjunto de dados que será usado no treinamento do modelo de aprendizado de máquina. Essa abordagem por janela deslizante permitirá criar novas variáveis para auxiliar o aprendizado do modelo. Além de aumentar o número de observações que o modelo terá para aprender.

O trabalho foca nas séries temporais da competição M3, na utilização dos modelos apre-

sentados no Capítulo 2 e na técnica de janela deslizante para elaborar o conjunto de dados. O objetivo fim do trabalho é desenvolver um *stacking* dos modelos tradicionais de séries temporais com modelos de aprendizado de máquina a fim de comparar os resultados com a literatura. A fim de entender se o uso da janela deslizante para criação do conjunto de dados traz um resultado mais eficiente as previsões.

Cabe ressaltar que tal proposta esta alinhada com as abordagens mais bem sucedidas na competição M4, a qual concluiu que método de aprendizagem de máquina puros são inadequados para serem empregados para séries temporais univariadas de baixa frequência como as disponíveis no banco de dados da M3. Por outro lado, métodos híbridos, os quais combinam aprendizagem de máquina com modelos estatísticos, obtiveram o topo do *ranking*. Tais resultados e conclusões podem ser revisados em Makridakis, Spiliotis e Assimakopoulos (2020).

Abaixo apresentamos o algoritmo de **stacking** proposto nesse trabalho.

1. Construir via janela deslizante um banco de dados de treinamento, o qual é formado por predições de diversos modelos estatísticos para todas as séries temporais do banco de dados da M3.

2. Treinar um modelo de aprendizagem de máquina supervisionado para combinar via regressão as predições dos modelos estatísticos de acordo com as características de cada série temporal.

3. Construir o banco de dados de teste, o qual deve ser formado pelas previsões dos modelos estatísticos para todas as séries do banco de dados da M3.

4. Calcular as previsões finais utilizando o modelo treinado na etapa 2 para o banco de dados de teste obtido na etapa 3.

3.2 Competição de Séries Temporais

Uma competição de Séries Temporais consiste em avaliar e selecionar as melhores técnicas de previsão utilizando-se apenas de dados históricos das séries.

Nesse trabalho, iremos focar nas competições de Makridakis (também conhecida como M - número da edição da competição) que são competições organizadas pelo IIF(International Institute of Forecasters) desde 1982.

Dentre as competições, utilizaremos as séries disponibilizadas nas competições M3 (Makridakis e Hibon, 2000).

As séries disponibilizadas no banco de dados das competições são subdividas pelo horizonte de observações, podendo ser Anual, Mensal, Quadrimestral e Outras.

3.3 Dados M3

As séries temporais utilizadas nesse trabalho são provenientes da competição M3, (Makridakis e Hibon, 2000). A manipulação das séries temporais foi feita através do *software* R, uma vez que, as informações da competição M3 já estão internalizadas na biblioteca *Mcomp*.

A competição M3 contém 3003 séries temporais com diferentes frequências. A Tabela 3.1 apresenta a distribuição das séries de acordo com as diferentes frequências.

Tabela 3.1: Séries Temporais da Competição M3.

Frequência	Horizonte de Previsão	Número de séries temporais
Anual	413	645
Mensal	308	1428
Trimestral	519	756
Outras	204	174

Ademais, como mostra a Tabela 3.2, as séries M3 são divididas por característica das observações.

Tabela 3.2: Tipo de Séries Temporais

Tipo	Total de Séries
Demografia	413
Finanças	308
Indústria	519
Macro	731
Micro	828
Outras	204

3.4 Preparação do banco de dados para aprendizagem cruzada

Na abordagem de *stacking* proposta nesse trabalho, utilizamos a técnica de janela deslizante como diferencial na elaboração do conjunto de dados utilizado para o ajuste do modelo de aprendizado de máquina. A escolha do uso da janela deslizante se deve ao fato de podermos elaborar um conjunto de dados com mais informações sobre as séries temporais e ter previsões em diversos horizontes de tempo.

A técnica de janela deslizante consiste em dividir a série temporal em diversos conjuntos de ajuste e teste, onde em cada passo um é usado para ajustar o modelo de série temporal e o segundo é usado para verificar as previsões. Dado n o número de observações da série temporal e h o horizonte de previsão da série temporal. A Figura 3.1 ilustra a aplicação da técnica de janela deslizante.

Seguindo a ilustração da Figura 3.1, o banco de dados de treinamento será então formado basicamente como uma planilha em que cada linha representa uma célula apresentada como amarela na figura, a qual deve conter o valor real do ponto da série, as características da série e as previsões dos modelos estatísticos para aquele ponto de acordo com cada horizonte.

O uso dessa técnica permite criar um número maior de observações baseado no horizonte de previsão da janela deslizante. Além disso, podemos utilizar como variável para o conjunto de dados as informações sobre a característica da série temporal.

Com o uso da janela deslizante, ajustamos os modelos apresentados no Capítulo 2 e salva-

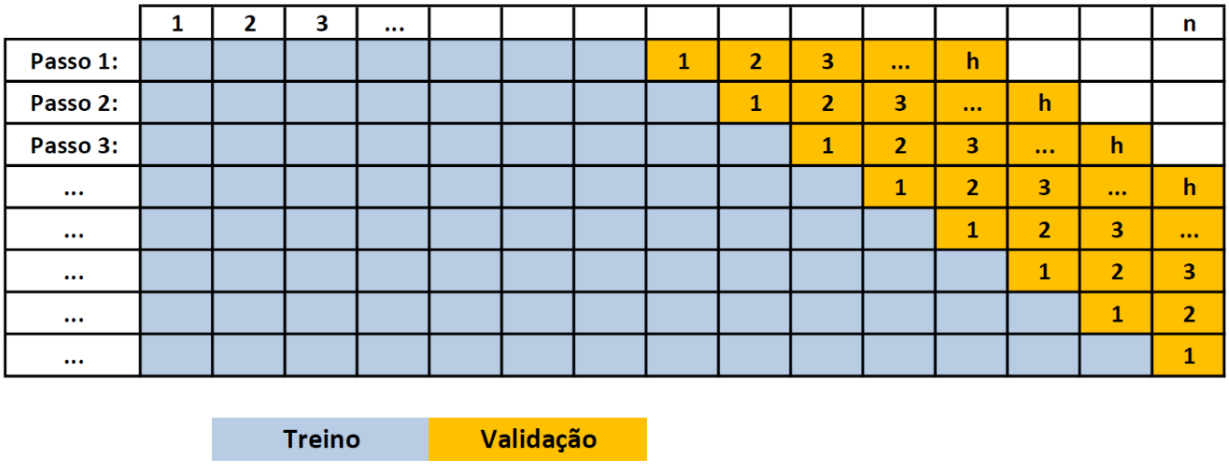

Figura 3.1: Janela deslizante.

mos as previsões estimadas para cada janela.A Figura 3.2 ilustra algumas linhas do banco de dados de treinamento.

ANOS	MESES	ID	TIPO	TAM_SERIE	VALOR_OBSERVADO	HORIZONTE	ARIMA	SES	HOLT	HW	NAIVE	DAMPED	STM	OTM	DOTM	DSTM	STHETA	ETS
1991	1	Q1	MICRO	36	5591.95	1	5706.60	5706.60	5796.40	5734.85	5706.60	5756.47	5771.70	5802.96	5788.20	5771.70	5771.70	5706.60
1991	4	Q1	MICRO	36	5605.15	2	5706.60	5706.60	5886.19	5802.45	5706.60	5805.34	5835.81	5898.01	5867.17	5834.57	5835.81	5706.60
1991	4	Q1	MICRO	36	5605.15	1	5591.95	5591.96	5680.52	5679.93	5591.95	5642.29	5655.99	5663.52	5655.03	5655.99	5655.99	5591.96
1991	7	Q1	MICRO	36	5630.00	3	5706.60	5706.60	5975.97	5849.62	5706.60	5853.23	5899.91	5993.06	5944.75	5896.24	5899.91	5706.60
1991	7	Q1	MICRO	36	5630.00	2	5591.95	5591.96	5769.06	5734.75	5591.95	5691.61	5718.25	5733.24	5714.65	5716.54	5718.25	5591.96
1991	7	Q1	MICRO	36	5630.00	1	5605.15	5605.15	5693.37	5673.51	5605.15	5653.39	5666.04	5672.46	5661.37	5666.04	5666.04	5605.15
1991	10	Q1	MICRO	36	5589.20	4	5706.60	5706.60	6065.76	6156.12	5706.60	5900.16	5964.02	6088.11	6021.03	5956.75	5964.02	5706.60
1991	10	Q1	MICRO	36	5589.20	3	5591.95	5591.96	5857.61	6019.49	5591.95	5739.95	5780.51	5802.96	5772.72	5775.52	5780.51	5591.96
1991	10	Q1	MICRO	36	5589.20	2	5605.15	5605.15	5781.59	5948.05	5605.15	5700.66	5726.45	5739.21	5715.55	5724.73	5726.45	5605.15
1991	10	Q1	MICRO	36	5589.20	1	5630.00	5630.00	5718.21	5903.95	5630.00	5677.26	5924.20	5929.70	5921.14	5924.20	5924.20	5630.00
1992	1	Q1	MICRO	36	5551.25	5	5706.60	5706.60	6155.55	6184.37	5706.60	5946.15	6028.12	6183.15	6096.07	6016.12	6028.12	5706.60

Figura 3.2: Conjunto de dados de treinamento.

Para facilitar o entendimento do conjunto de dados gerado, a Tabela 3.3 representa o dicionário de variáveis.

Todo procedimento para a construção do conjunto de dados foi feito com o auxílio do o *software* R e os dados foram salvos em um banco de dados MYSQL local. Essa escolha, facilita o acesso via qualquer linguagem de programação e permite escalabilidade dos dados.

Tabela 3.3: Dicionário de Variáveis

Variáveis	Significado
ANOS	Ano de observação
MESES	Mês de observação
ID	Identificador único
TIPO	Característica das observações
TAM _SERIE	Quantidade de observações
VALOR _OBSERVADO	Valor observado real
HORIZONTE	Horizonte da previsão
ARIMA	Previsão do modelo ARIMA
SES	Previsão do modelo SES
HOLT	Previsão do modelo HOLT
HW	Previsão do modelo Holt-Winters
NAIVE	Previsão do modelo Naive
DAMPED	Previsão do modelo Holt Amortecido
STM	Previsão do modelo Theta Padrão
OTM	Previsão do modelo Theta Otimizado
DOTM	Previsão do modelo Theta Otimizado Dinâmico
DSTM	Previsão do modelo Theta Otimizado Dinâmico com $\theta = 2$
STHETA	Previsão do modelo Theta Otimizado com $\theta = 2$
ETS	Previsão do modelo Suavização Exponencial de espaço estado

3.5 Combinação de Previsão via aprendizado de máquina

O processo de combinar previsões de modelos clássicos de séries temporais com modelos de aprendizado de máquina pode ser visto em Montero-Manso et al. (2020), Ahmed et al. (2010), Barker (2020), Gilliland (2020), Petropoulos e Svetunkov (2020), Smyl (2020) e Jaganathan e Prakash (2020).

Essa abordagem conjunta, denominada de *stacking*, consiste em utilizar as previsões dos modelos clássico de séries temporais como entrada de um outro modelo de aprendizado de máquina. Essa abordagem tende a estimar previsões mais robustas do que os modelos separadamente (Makridakis e Hibon, 2000).

O trabalho propõem o uso da técnica de janela deslizante para construção do conjunto de dados e o uso da técnica de *stacking* para realizar previsões. O uso da janela deslizante proporcionou a geração do conjunto de dados apresentado na Seção 3.4.

Wirth e Hipp (2000) propõem que o processo de ajuste de um modelo de aprendizado de máquina consiste em 6 passos: Entendimento de Negócio, Entendimento dos Dados, Preparação dos Dados, Modelagem, Avaliação e Implantação. Nesse trabalho, vamos focamos nas etapas de Preparação dos Dados, Modelagem e Avaliação.

A etapa de Preparação do dados é o momento em que aplicamos técnicas de transformação matemática nas variáveis existentes para extrair o maior potencial dos dados, e eventualmente, criar novas variáveis.

Tome o conjunto de dados apresentado na Seção 3.4 e aplicamos a técnica de *encoder*, que é um processo em que variáveis categóricas são convertidas em variáveis numéricas, nas colunas categóricas. Em seguida, separamos, de forma aleatória, o conjunto de dados em dois novos conjuntos: conjunto de treino e conjunto de teste. Essa separação é necessária para conseguirmos validar o ajuste do modelo e podermos calcular as métricas de avaliação do modelo (Tan et al., 2021).

A etapa de Modelagem é o momento onde várias técnicas de modelagem são aplicadas, e

seus parâmetros calibrados para otimização. Nesse trabalho, as técnicas de modelagem que serão utilizadas são *XGBoost* e o *LGBM*, apresentados no Capítulo 2. Durante o ajuste dos modelos, aplica-se a técnica de validação cruzada, apresentado no Capitulo 2, com um valor de 5 *folds*.

A etapa de Avaliação será discutida no próximo Capítulo.

Capítulo 4

Resultados

Nesse capítulo vamos apresentar a avaliação dos modelos nas bases criadas no capítulo 3. Ademais, faremos a comparação dos resultados obtidos da abordagem proposta com os resultados apresentados em Fiorucci et al. (2016).

4.1 Avaliação dos Modelos de Aprendizado de Máquina

No Capítulo 3 vimos os passos para o processo de ajuste de modelos de aprendizado de máquina.

Nessa seção, vamos avaliar os modelos ajustados e escolher qual deles utilizar na nossa abordagem de *stacking*.

A etapa de Avaliação consiste em utilizar o modelo ajustado e fazer previsões. Em seguida, comparamos as previsões do modelos com as observações do conjunto de teste. A escolha do modelo de aprendizado de máquina será baseado nos valores obtidos do MAPE, pois, vamos comparar o desempenho dos modelos em todas as séries da M3 (Makridakis, Spiliotis e Assimakopoulos, 2018a). A Tabela 4.1 mostra uma comparação entre o XGBoost e o LGBM
.

Optamos por utilizar o LGBM para o prosseguimento do trabalho. Assim, o próximo passo

Modelo	MAPE(%)
XGBoost	16.5
LGBM	15.3

Tabela 4.1: XGBoost X LGBM.

é retreinar o LGBM e fazer otimização de parâmetros. Esse passo foi feito com o auxílio da função *RandomizedSearchCV*. Essa função permite treinar vários modelos LGBM com diferentes conjuntos parâmetros, definidos de maneira aleatória, buscando encontrar um conjunto de parâmetros que minimize o erro de previsão. Essa etapa é conhecida como otimização de hiperparâmetros e é uma etapa importante do processo de treinamento de uma modelo de aprendizado de máquina.

Feito a escolha do modelo, o próximo passo é comparar a abordagem proposta com a literatura.

4.2 Modelos de séries temporais x Aprendizado de Máquina

Makridakis e Hibon (2000), citam as vantagens do uso da abordagem conjunta entre modelos de séries temporais e modelos de aprendizado de máquina em dados de competição. A abordagem proposta nesse trabalho, desenvolvida no Capítulo 3, busca gerar resultados superiores aos apresentados por Fiorucci et al. (2016).

O uso combinado da janela deslizante com as previsões dos modelos de séries temporais, em conjunto com o LGBM permitiu que nossa abordagem de *stacking* gerasse previsões mais robustas. Esse fato é ilustrado quando calculamos as métricas MAPE e sMAPE e comparamos com os resultados da literatura. A Tabela 4.2 apresenta os resultados da métrica sMAPE.

Ao olharmos as séries separadas por frequência, observamos que o desempenho da abordagem de *stacking* proposta é superior ao dos modelos competidores.

No conjunto das séries anuais e mensais, a abordagem proposta apresenta resultados melhores que os modelos de séries temporais (Fiorucci et al., 2016). Segundo Makridakis, Spiliotis

Modelos	Anual	Trimestral	Mensal	Outros	Todas
Stacking	**14.77**	9.51	**13.60**	4.91	**12.73**
Naive	17.88	11.32	18.18	6.30	16.58
SES	17.78	10.83	16.14	6.30	15.07
Damped	17.07	10.96	16.25	**4.30**	15.02
ETS	16.89	9.69	14.07	4.34	13.28
ARIMA	17.62	9.99	15.30	4.54	14.27
STheta	16.74	9.23	13.83	4.93	13.05
STM	16.73	9.24	13.85	4.93	13.06
OTM	16.60	**9.14**	14.11	4.85	13.21
DSTM	16.69	9.24	13.82	4.92	13.04
DOTM	15.94	9.28	13.74	4.58	12.90

Tabela 4.2: Comparação dos Resultados

e Assimakopoulos (2017) e Ahmed et al. (2010) abordagens combinadas de modelos de séries temporais com modelos de aprendizado de máquina apresentam melhores resultados nas séries mensais da competição M3.

Nas séries trimestrais, observa-se que o desempenho da abordagem proposta é melhor que que os modelos clássicos de séries temporais, mas não consegue superar o desempenho dos modelos OTM. Os resultados apresentados em Fiorucci et al. (2016) e Fiorucci et al. (2015) indicam o bom desempenho dos modelos derivados do método Theta em séries temporais da competição M3.

O conjunto das séries com periodicidade diferentes das apresentadas anteriormente, representadas no banco de dados como outros, observa-se um comportamento bem diferente das demais. Nesse conjunto o modelo vencedor é o Suavização Exponencial de Holt Amortecido (Fiorucci et al., 2016).

Portanto, é possível observar que a abordagem de *stacking* proposta nesse trabalho teve um bom desempenho nas séries temporais da competição M3. Com destaque para as séries de frequência anual e mensal.

Ademais ao comparar-se com os modelos Theta, que são modelos com excelente performance nos dados da competição M3 (Fiorucci et al., 2016), é possível ver um desempenho

semelhante nas séries trimestrais.

Makridakis, Spiliotis e Assimakopoulos (2018a) apresentam um conjunto de modelos de aprendizado de máquina ajustados aos dados da competição M3 sem o uso da técnica de janela deslizante. A Tabela 4.3 ilustra os resultados apresentados por Makridakis, Spiliotis e Assimakopoulos (2018a) em comparação com a abordagem proposta.

Modelos	sMAPE(%)
Stacking	12.73
Rede Neural Artificial (AAN)	14.10
Função de Base Radial (RBF)	15.79

Tabela 4.3: Comparação de Resultados.

Makridakis, Spiliotis e Assimakopoulos (2018a) apresentam o uso de redes neurais artificias nos dados da competição M3. A técnica de AAN é um método de aprendizado de máquina que ensina computadores a processar dados de uma forma inspirada pelo cérebro humano. É chamado de aprendizado profundo, que usa nós ou neurônios interconectados em uma estrutura de camadas, semelhante ao cérebro humano. A AAN cria um sistema adaptativo divido em camadas, onde temos a camada de entrada, camada de saída e camda intermediária. Os dados são processados em uma camada e depois é enviado pra camada seguinte seguindo o comportamento de uma função de ativação, usualmente a função sigmoidal, permitindo que usem os erros e se aprimorarem continuamente (Mohammadi et al., 2017; Kůrková, 1992). Sarimveis et al. (2004) apresenta a Função de Base Radial, que é uma derivação do conceito de AAN. Apresentando três diferenças principais em relação às redes AAN:

- Elas sempre apresentam uma única camada intermediária;
- Os neurônios de saída são sempre lineares;
- Os neurônios da camada intermediária têm uma função de base radial como função de ativação, ao invés de uma função sigmoidal.

É interessante observar que o modelo de *stacking* proposto tem um desempenho melhor no conjunto geral das séries temporais da competição M3. Além disso, vale ressaltar que o uso da técnica de janela deslizante permite que o modelo de aprendizado de máquina possa encontrar mais relações entre as séries temporais, assim, sendo um dos motivos do bom desempenho da abordagem proposta.

Capítulo 5

Conclusão

O objetivo geral deste trabalho foi cumprido. Isto significa que a abordagem proposta de utilizar a técnica de janela deslizante para construção do conjunto de dados e em seguida o uso de um algoritmo de aprendizado de máquina se mostrou eficiente em comparação com a literatura. Notou-se que os resultados, gerados por esta abordagem foram superiores que os resultados da literatura nas séries mensais e anuais. Efetivamente, foi possível utilizar a técnica de janela deslizante para elaborar uma conjunto de dados mais rico em informação, isto é, com um número maior de previsões em horizontes diferentes para cada série temporal.

A importância do resultado obtido é evidenciada em Makridakis, Spiliotis e Assimakopoulos (2020) e Makridakis e Hibon (2000), onde mostram que a abordagem pura de aprendizado de máquina apresentam resultados inapropriados para prever séries temporais univariadas de baixa frequência, que é o caso das séries da competição M3. Além disso, Makridakis, Spiliotis e Assimakopoulos (2020) destacam a importância do uso combinado de modelos tradicionais de séries temporais com modelos de aprendizado de máquina. Fato esse, que foi ilustrado nesse trabalho, indicando que a adição da técnica de janela deslizante traz vantagens para a construção do conjunto de dados utilizado no modelo de aprendizado de máquina.

Os resultados obtidos nesse trabalho abrem espaço para o uso da abordagem proposta nos dados da competição M4 (Makridakis, Spiliotis e Assimakopoulos, 2020). Além da compe-

tição M4 ter um banco de série temporais maior que o da competição M3, podemos ver em Barker (2020) e Grushka-Cockayne e Jose (2020) que o uso de modelos de aprendizado de máquina híbridos, ou seja, combinados com modelos estatísticos clássicos vêm apresentando bons resultados na competição.

Referências Bibliográficas

Ahmed, Nesreen K et al. (2010). "An empirical comparison of machine learning models for time series forecasting". *Econometric reviews* 29.5-6, pp. 594–621.

Assimakopoulos, Vassilis e Nikolopoulos, K. (out. de 2000). "The theta model: A decomposition approach to forecasting". *International Journal of Forecasting* 16, pp. 521–530.

Barker, Jocelyn (2020). "Machine learning in M4: What makes a good unstructured model?" *International Journal of Forecasting* 36.1. M4 Competition, pp. 150–155.

Berrar, Daniel (jan. de 2018). "Cross-Validation". Em:

Billah, Baki et al. (2006). "Exponential smoothing model selection for forecasting". *International Journal of Forecasting* 22.2, pp. 239–247.

Bowerman, Bruce L, O'Connell, Richard T e Koehler, Anne B (2005). *Forecasting, time series, and regression: an applied approach*. Vol. 4. South-Western Pub.

Cha Zhang, Yunqian Ma (2012). "Ensemble Machine Learning". Em: *Ensemble Machine Learning*. Springer New York, NY, pp. 1–332.

Chen, Tianqi e Guestrin, Carlos (2016). "XGBoost: A scalable tree boosting system". *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Daoud, Essam Al (2019). "Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset". *International Journal of Computer and Information Engineering* 13.1, pp. 6 –10.

de Myttenaere, Arnaud et al. (2016). "Mean Absolute Percentage Error for regression models". *Neurocomputing* 192. Advances in artificial neural networks, machine learning and computational intelligence, pp. 38–48.

Dietterich, Thomas G. (2000). "Ensemble Methods in Machine Learning". *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–15.

Dietterich, Tom (1995). "Overfitting and undercomputing in machine learning". *ACM computing surveys (CSUR)* 27.3, pp. 326–327.

Fildes, Robert (2020). "Learning from forecasting competitions". *International Journal of Forecasting* 36.1. M4 Competition, pp. 186–188.

Fiorucci, Jose A. et al. (2016). "Models for optimising the theta method and their relationship to state space models". *International Journal of Forecasting* 32.4, pp. 1151–1161.

Fiorucci, Jose Augusto e Louzada, Francisco (2020). "GROEC: Combination method via Generalized Rolling Origin Evaluation". *International Journal of Forecasting* 36.1, pp. 105–109.

Fiorucci, José Augusto et al. (2015). "The optimised theta method". *arXiv preprint arXiv:1503.03529*.

Freund, Yoav e Schapire, Robert E (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Friedman, Jerome, Hastie, Trevor e Tibshirani, Robert (2000). "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)". *The Annals of Statistics* 28.2, pp. 337 –407.

Fry, Chris e Brundage, Michael (2019). "The M4 Forecasting Competition - A Practitioner's View". *International Journal of Forecasting* July. None, p. 5.

Gilliland, Michael (2020). "The value added by machine learning approaches in forecasting". *International Journal of Forecasting* 36.1. M4 Competition, pp. 161–166.

Grushka-Cockayne, Yael e Jose, Victor Richmond R. (2020). "Combining prediction intervals in the M4 competition". *International Journal of Forecasting* 36.1. M4 Competition, pp. 178–185.

Hanke, John E e Wichern, Dean W (2005). *Business forecasting*. Pearson Educación.

Heckman J. J.; Leamer, E. (2007). *Handbook of Econometrics (Book 2)*. North Holland.

Hyndman, Robin John e Athanasopoulos, George (2018). *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts.

Jaganathan, Srihari e Prakash, P.K.S. (2020). "A combination-based forecasting method for the M4-competition". *International Journal of Forecasting* 36.1, pp. 98–104.

Ke, Guolin et al. (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". *Advances in Neural Information Processing Systems*. Ed. por I. Guyon et al. Vol. 30. Curran Associates, Inc.

Kim, Sungil e Kim, Heeyoung (2016). "A new metric of absolute percentage error for intermittent demand forecasts". *International Journal of Forecasting* 32.3, pp. 669–679.

Kůrková, Věra (1992). "Kolmogorov's theorem and multilayer neural networks". *Neural Networks* 5.3, pp. 501–506.

Makridakis, Spyros e Hibon, Michele (2000). "The M3-Competition: results, conclusions and implications". *International journal of forecasting* 16.4, pp. 451–476.

Makridakis, Spyros, Spiliotis, Evangelos e Assimakopoulos, Vassilios (2018a). "Statistical and Machine Learning forecasting methods: Concerns and ways forward". *PLOS ONE* 13, pp. 1–26.

Makridakis, Spyros, Spiliotis, Evangelos e Assimakopoulos, Vassilios (2018b). "The M4 Competition: Results, findings, conclusion and way forward". *International Journal of Forecasting* 34.4, pp. 802–808.

Makridakis, Spyros, Spiliotis, Evangelos e Assimakopoulos, Vassilios (2020). "The M4 Competition: 100,000 time series and 61 forecasting methods". *International Journal of Forecasting* 36.1, pp. 54–74.

Makridakis, Spyros, Spiliotis, Evangelos e Assimakopoulos, Vassilis (2017). "The accuracy of machine learning (ML) forecasting methods versus statistical ones: extending the results of the M3-Competition". *no. October*.

Meade, Nigel (2000). "A note on the Robust Trend and ARARMA methodologies used in the M3 Competition". *International Journal of Forecasting* 16.4, pp. 517–519.

Mohammadi, Mohammad Reza et al. (2017). "A brief review over neural network modeling techniques". *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 54–57.

Montero-Manso, Pablo et al. (2020). "FFORMA: Feature-based forecast model averaging". *International Journal of Forecasting* 36.1, pp. 86–92.

Morettin, PA e Toloi, C (2006). *Análise de séries temporais*.

Natekin, Alexey e Knoll, Alois (2013). "Gradient boosting machines, a tutorial". *Frontiers in neurorobotics* 7, p. 21.

Petropoulos, Fotios e Makridakis, Spyros (jul. de 2019). "The M4 competition: Bigger. Stronger. Better". *International Journal of Forecasting* 36.

Petropoulos, Fotios e Svetunkov, Ivan (2020). "A simple combination of univariate models". *International Journal of Forecasting* 36.1, pp. 110–115.

Sarimveis, Haralambos et al. (2004). "A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms". *Computers Chemical Engineering* 28.1. Escape 12, pp. 209–217.

Schapire, Robert E. (2003). "The Boosting Approach to Machine Learning: An Overview". Em: *Nonlinear Estimation and Classification*. Ed. por David D. Denison et al. New York, NY: Springer New York, pp. 149–171.

Smyl, Slawek (2020). "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting". *International Journal of Forecasting* 36.1, pp. 75–85.

Tan, Jimin et al. (2021). "A critical look at the current train/test split in machine learning". *arXiv preprint arXiv:2106.04525*.

Wei, William WS et al. (2006). *Time series analysis: univariate and multivariate methods*. Pearson Addison Wesley.

Wirth, Rüdiger e Hipp, Jochen (2000). "CRISP-DM: Towards a standard process model for data mining". *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Manchester, pp. 29–39.

Wooldridge, J. M. (2013). *Introductory econometrics: a modern approach*. Mason, Ohio South-Western Cengage Learning.