UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE CIÊNCIAS EXATAS E DA TERRA
BACHARELADO EM ESTATÍSTICA

JOÃO PEDRO FLORENTINO DE OLIVEIRA NASCIMENTO

ALGORITMOS DE APRENDIZADO ONLINE PARA PREVISÃO DE SÉRIES TEMPORAIS : UMA ANÁLISE DO MODELO ARIMA INCREMENTAL

CUIABÁ - MT
2024

João Pedro Florentino de Oliveira Nascimento

ALGORITMOS DE APRENDIZADO ONLINE PARA PREVISÃO DE SÉRIES TEMPORAIS : UMA ANÁLISE DO MODELO ARIMA INCREMENTAL

Trabalho de Conclusão de Curso apresentado à banca examinadora do Curso de Bacharelado em Estatístisca da Universidade Federal de Mato Grosso, como requisito parcial, para obtenção do título de Bacharel em Estatística

Orientador: Prof. Dr. Anderson C. S. Oliveira

CUIABÁ - MT
2024

JOÃO PEDRO FLORENTINO DE OLIVEIRA NASCIMENTO

ALGORITMOS DE APRENDIZADO ONLINE PARA PREVISÃO DE SÉRIES TEMPORAIS : UMA ANÁLISE DO MODELO ARIMA INCREMENTAL

Monografia apresentada à banca examinadora do Curso de Bacharelado em Estatístisca da Universidade Federal de Mato Grosso, como requisito parcial, para obtenção do título de Bacharel em Estatística

BANCA EXAMINADORA

Prof. Dr Anderson Castro Soares de Oliveira Orientador
Prof. Dr Jose Nilton da Cruz Examinador
Prof. Dr Juliano Bortolini Examinador

CUIABÁ - MT
2024

RESUMO

Este trabalho investigou a performance do modelo de aprendizado online, ARIMA incremental, abordando sua adaptabilidade e poder preditivo por meio de simulações sob diversas condições de *drift* e na aplicação a dados reais. Contrariamente à hipótese inicial de crescimento sublinear, os resultados mostraram que tanto o *regret* médio quanto o mediano crescem de maneira linear, porém com um coeficiente angular significativamente menor que $\alpha = 1$, destacando a capacidade do modelo de manter um aumento controlado no *regret* e oferecer previsões competitivas em comparação a um modelo considerado ideal. Na análise utilizando dados reais, o modelo ARIMA incremental exibiu um desempenho similar ao de um modelo ARIMA ajustado por janelas móveis, que realiza uma seleção automatizada de parâmetros. Em situações de alta volatilidade, ambos os modelos enfrentaram desafios; porém, a versão incremental mostrou-se ligeiramente superior em contextos de volatilidade baixa a média. Tais resultados sugerem que a abordagem incremental é capaz de alcançar desempenhos comparáveis aos de métodos que dependem de seleção e ajuste complexo de parâmetros, especialmente em ambientes dinâmicos que exigem rápida adaptação.

Palavras-chaves: ARIMA incremental; aprendizado online; condições de *drift*; adaptabilidade; volatilidade; desempenho preditivo.

ABSTRACT

This work investigated the performance of the incremental ARIMA model in online learning, addressing its adaptability and predictive power through simulations under various *drift* conditions and in the application to real data. Contrary to the initial hypothesis of sublinear growth, the results showed that both the mean and median regret grow linearly, yet with a coefficient significantly lower than $\alpha = 1$, highlighting the model's ability to maintain controlled regret growth and provide competitive forecasts compared to an ideally performing model. In the analysis using real data, the incremental ARIMA model exhibited performance similar to that of a traditional ARIMA model adjusted by moving windows, which performs an automated parameter selection. In situations of high volatility, both models faced challenges; however, the incremental version proved to be slightly superior in low to medium volatility contexts. Such results suggest that the incremental approach is capable of achieving performances comparable to methods that rely on complex parameter selection and adjustment, especially in dynamic environments that require quick adaptation.

Keywords: Incremental ARIMA; Online Learning; *Drift* Conditions; Adaptability, Volatility; Predictive Performance.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 Introdução

A modelagem de séries temporais foca na análise de dados ordenados sequencialmente no tempo e é empregada em diversos campos do saber, como economia, finanças, meteorologia e engenharia, devido à sua capacidade de discernir e antecipar padrões temporais em diferentes fenômenos (BROCKWELL; DAVIS, 2016; BOX et al., 2015; MORETTIN; TOLOI, 2018).

A característica predominante de uma série temporal é sua dependência temporal; isto é, observações consecutivas tendem a exibir autocorrelação significativa. Esta autocorrelação, inerente às séries temporais, carrega consigo tanto desafios quanto oportunidades. Por um lado, é explorada para projetar valores futuros baseando-se em dados anteriores, por outro, essa mesma característica impede a aplicação adequada de muitos modelos tradicionais de aprendizado de máquina, que presumem independência entre as observações. Assim, para tratar a complexidade temporal desses dados, técnicas específicas como o modelo autoregressivo (AR) e o modelo de médias móveis (MA) foram desenvolvidas (BROCKWELL; DAVIS, 2016).

A principal finalidade da modelagem de séries temporais é a elaboração de previsões, as quais podem levar a decisões mais fundamentadas, otimização de recursos e um melhor entendimento das tendências inerentes ao fenômeno em estudo. Entretanto, é crucial que um modelo uma vez ajustado, seja continuamente verificado e ajustado conforme necessário, para assegurar sua adaptabilidade a cenários nos quais a distribuição dos dados pode sofrer alterações ao longo do tempo, mantendo, assim, sua precisão e relevância (ANAVA et al., 2013; LI; WU; LIU, 2023).

Nesse contexto, os algoritmos de aprendizado de máquina incrementais trazem a proposta de aprender e adaptar-se continuamente à medida que novos dados são disponibilizados. Diferentemente dos métodos convencionais, a atualização de seus parâmetros é feita sem a necessidade de reprocessar todo o conjunto de dados o que os tornam especialmente úteis em ambientes onde os dados são atualizados com alta frequência e o custo computacional de re-treinamento é alto (BOTTOU, 1998).

Os modelos ARMA, uma integração dos modelos autoregressivos (AR) e de médias móveis (MA), são fundamentais para a análise de séries temporais devido à sua capacidade de capturar diferentes dependências temporais em dados sequenciais. Um avanço significativo nessa área foi apresentado por Anava et al. (2013), cujo estudo evidenciou que o modelo ARMA incremental converge assintoticamente para o melhor modelo ARMA em retrospecto. Esta descoberta revela a possibilidade de otimização progressiva dos parâmetros do modelo.

O modelo Autoregressivo Integrado de Médias Móveis (ARIMA) se destaca como um dos modelos lineares mais aplicados em previsões de séries temporais, valorizado por suas excelentes propriedades estatísticas e sua adaptabilidade considerável. Dentro do contexto dos modelos ARIMA, uma variedade de algoritmos foi desenvolvida com o objetivo de implementar uma versão incremental do modelo (LI; WU; LIU, 2023; LIU et al., 2016; SHAO et al., 2021). Esses avanços procuram aprimorar a capacidade do ARIMA de adaptar-se e aprender com novos dados, promovendo melhorias contínuas na precisão e na confiabilidade das previsões geradas pelo modelo. Esse progresso incremental se revela essencial em campos onde as condições podem mudar rapidamente, requerendo modelos que possam ajustar-se de maneira ágil e precisa às novas circunstâncias.

Assim, este trabalho foi desenvolvido com a finalidade de explorar o modelo ARIMA incremental por meio da avaliação do seu poder preditivo e capacidade de adaptabilidade em cenários com dinâmicas distintas por meio de estudos de simulação e de um estudo de caso nos dados do *Europe Brent Spot Price*.

2 Objetivos

2.1 Objetivo geral

Investigar poder preditivo e capacidade de adaptabilidade do modelo ARIMA incremental em ambientes que exigem respostas rápidas às mudanças.

2.2 Objetivos específicos:

- Verificar se o *regret* do modelo ARIMA incremental apresenta um crescimento sublinear em relação ao número de iterações quando submetido a diferentes condições de *d*rift.
- Comparar a capacidade de *n*owcast dos modelos ARIMA e ARIMA incremental utilizando os dados do *Europe Brent Spot Price*.
- Avaliar o impacto da volatilidade nos modelos ARIMA e ARIMA incremental utilizando os dados do *Europe Brent Spot Price*.

3 Referencial Teórico

3.1 Variação Dinâmica e Adaptabilidade: Instabilidade dos Dados

Em cenários caracterizados pela variabilidade e volatilidade dos dados, em que alterações podem ser rápidas e inesperadas, é vital entender e abordar o fenômeno conhecido como *drifts*. Essa instabilidade nos dados pode se manifestar por uma série de razões, sendo elas mudanças abruptas nas tendências de mercado, flutuações climáticas, ou avanços tecnológicos e sociais (GAMA et al., 2014; LIU et al., 2016)..

A origem dessas mudanças bruscas nos dados é diversa e, se não forem prontamente identificadas e adequadamente gerenciadas, têm o potencial de provocar uma degradação significativa no desempenho e na precisão dos modelos de aprendizado de máquina. Como consequência, modelos que não estão preparados para se adaptar a esses *drifts* podem começar a gerar previsões cada vez mais imprecisas, tornando-se, assim, obsoletos em um curto espaço de tempo (GAMA et al., 2014; LIU et al., 2016)..

De acordo com Gama et al. (2014), existem diferentes tipos de *drifts*, os quais são classificados principalmente baseando-se em sua natureza e na velocidade com que ocorrem:

- *Drifts* Abruptos: representam mudanças repentinas e significativas na distribuição dos dados. Por exemplo, uma alteração abrupta nas preferências dos consumidores devido a um evento externo inesperado.

- *Drifts* Incrementais: são caracterizados por mudanças que ocorrem de forma gradual ao longo do tempo, como uma crescente mudança na popularidade de um produto específico ao longo de meses.

- *Drifts* recorrentes: Referem-se à reaparição de certos padrões ou tendências em intervalos regulares. Este tipo de *drifts* é comum em dados sazonais, como vendas durante períodos festivos.

A detecção eficiente de *drifts* no fluxo de dados é fundamental para preservar a acurácia e a relevância de modelos preditivos em cenários dinâmicos. Uma estratégia frequentemente adotada para tal é a implementação de janelas deslizantes, onde a distribuição de dados em uma janela recente é comparada com a de uma janela anterior. Discrepâncias significativas entre estas podem sinalizar uma possível mudança de conceito (DITZLER et al., 2015).

Uma abordagem eficaz para identificar tais mudanças é o uso do algoritmo KSWIN, fundamentado no teste não paramétrico de Kolmogorov-Smirnov, adaptado para fluxos de dados. Este método compara a distribuição de uma janela recente de dados com uma anterior para identificar discrepâncias significativas, que podem indicar uma mudança de conceito. A mudança é considerada significativa se:

$$\mathrm{dist}_{W,R} > \sqrt{-\frac{\ln \alpha}{r}},$$

em que α representa o nível de significância e r é o tamanho da janela R. Se a diferença nas distribuições empíricas entre as janelas R e W é substancial, assumindo-se que ambas deveriam provir da mesma distribuição, interpreta-se que ocorreu uma mudança de conceito (RAAB; HEUSINGER; SCHLEIF, 2020).

Após a detecção do *drift*, é necessário estabeler uma estratégias de adaptação do modelo de forma a manter a relevância e eficácia do modelo (ŽLIOBAITė, 2010). Alguns métodos incluem:

- Aprendizado Incremental: atualização do modelo em tempo real à medida que novos dados chegam.

- Esquecimento: descartando dados antigos que podem não ser mais relevantes para a distribuição atual.
- *Ensemble* de Modelos: utilização de múltiplos modelos e ponderação de suas previsões com base em sua performance recente.

3.2 Aprendizado Incremental

A aprendizagem incremental, ou aprendizagem *online*, é uma estratégia em que o processo de aprendizagem é contínuo e persistente ao longo do tempo, diferentemente do aprendizado em lote, que é consolidado em uma única ocasião com um conjunto completo de dados. Esta abordagem é vital em cenários onde re-treinar um modelo do zero a cada vez que novos dados são recebidos torna-se impraticável ou é excessivamente oneroso em termos computacionais, permitindo, ao invés disso, a atualização contínua do modelo à medida que mais dados se tornam disponíveis (CESA-BIANCHI; LUGOSI, 2006).

De acordo com Cesa-Bianchi e Lugosi (2006), considere um espaço de entrada $\mathcal{X}$ e um espaço de saída $\mathcal{Y}$. A cada iteração ou rodada t do processo de aprendizado incremental, o seguinte protocolo é seguido:

1. Revelação do Ponto de Entrada: um ponto x_t pertencente ao espaço de entrada $\mathcal{X}$ é revelado ao algoritmo de aprendizado, servindo como a nova entrada ou instância a ser considerada pelo modelo.
2. Realização da Predição: o algoritmo, utilizando x_t e os dados previamente revelados, realiza uma predição $\hat{y}_t$ referente à saída esperada para a instância atual.
3. Revelação da Saída Verdadeira: posteriormente à realização da predição, a verdadeira saída y_t, do espaço de saída $\mathcal{Y}$, é revelada, representando o valor real correspondente à instância considerada.
4. Cálculo da Perda e Atualização do Modelo: o modelo incide em uma perda, quantificada pela função de perda ℓ aplicada sobre a previsão e a verdadeira saída, i.e., $\ell(\hat{y}_t, y_t)$. Esta perda é então utilizada para atualizar o modelo, refinando seus parâmetros e aprimorando sua capacidade preditiva para futuras instâncias.

Dentro do aprendizado incremental, o objetivo é minimizar a perda acumulada, ou o *regret* (*regret*), ao longo das iterações. Isso sugere que o algoritmo de aprendizado deve adaptar-se e evoluir continuamente, melhorando sua precisão e reduzindo as discrepâncias em relação às verdadeiras saídas conforme mais dados são incorporados no processo de aprendizado (CESA-BIANCHI; LUGOSI, 2006; SHAO et al., 2021).

3.2.1 Algoritmos de aprendizado incremental

Algoritmos de aprendizagem incremental têm exercido uma influência significativa em diversas areas, possibilitando avanços em otimização e análise de dados. Robbins e Monro (1951) introduziu o *Stochastic Gradient Descent* (SGD). Este método foi proposto como uma solução eficaz para cenários onde o gradiente real da função de custo é desconhecido, necessitando ser estimado a partir de dados disponíveis.

Posteriormente, o *Perceptron* foi introduzido por Rosenblatt (1958). Apesar de ser principalmente associado a tarefas de classificação, o *Perceptron* também pode ser interpretado como um algoritmo incremental para o ajuste de hiperplanos em cenários de regressão.

Avançando na linha do tempo, Widrow e Hoff (1960) apresentaram o *Least Mean Squares* (LMS), um algoritmo projetado para otimização em problemas de regressão linear. Este método adaptativo modifica os pesos de um modelo linear por meio de uma regra iterativa, possibilitando ajustes incrementais aos dados à medida que são recebidos.

Finalmente, o *Recursive Least Squares* (RLS), detalhado por Haykin (2014), surgiu como uma solução robusta, especialmente apropriada para sistemas que lidam com grandes volumes de dados, reforçando ainda mais a importância do aprendizado incremental em ambientes de regressão.

3.2.1.1 *Stochastic Gradient Descent* (SGD)

O *Stochastic Gradient Descent*, frequentemente abreviado como SGD, é um método iterativo de otimização para funções de custo diferenciáveis. Ele foi projetado para encontrar o mínimo local (ou máximo) de uma função. Ao contrário dos métodos tradicionais de *Gradient Descent*, que calculam o gradiente da função de custo usando todo o conjunto de dados (conhecido como *batch* ou *full-batch*), o SGD estima este gradiente usando apenas um único exemplo de treinamento por vez Robbins e Monro (1951).

Considere uma função de custo $J(\boldsymbol{\theta})$ que queremos minimizar, onde $\boldsymbol{\theta}$ são os parâmetros do modelo. No contexto da regressão linear, $J(\boldsymbol{\theta})$ seria o erro quadrático médio. Para atualizar os parâmetros $\boldsymbol{\theta}$ usando SGD, o algoritmo segue a seguinte regra de atualização:

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \eta \nabla J(\boldsymbol{\theta}; x^{(i)}, y^{(i)}),$$

em que, η é a taxa de aprendizado, e ∇J é o gradiente da função de custo com relação aos parâmetros $\boldsymbol{\theta}$, calculado usando um único exemplo $(x^{(i)}, y^{(i)})$. A principal vantagem do SGD é sua eficiência. Como utiliza apenas um exemplo de treinamento por vez para calcular o gradiente, o SGD é computacionalmente mais eficiente, especialmente para conjuntos de dados muito grandes. Além disso, a natureza estocástica do SGD pode ajudá-lo a escapar de mínimos locais indesejados em funções de custo não-convexas.

No entanto, o SGD também apresenta desafios. Devido à sua natureza estocástica, o caminho de otimização do SGD é mais ruidoso e menos estável do que o *Gradient Descent* em lote. Isto pode resultar em convergência mais lenta ou mesmo na falha de convergência em certos cenários. Para remediar isso, muitas variantes do SGD foram propostas, incluindo a utilização de taxas de aprendizado adaptativas e algoritmos com *momentum* Robbins e Monro (1951).

3.2.1.2 *Perceptron*

O *Perceptron* é um dos algoritmos mais antigos no domínio da aprendizagem de máquina, proposto por Rosenblatt (1958). Embora seja mais frequentemente associado a tarefas de classificação binária, o *Perceptron* também pode ser aplicado em contextos de regressão, especialmente em cenários incrementais. Ele é essencialmente um modelo linear, semelhante à regressão linear, mas com uma diferença chave: a utilização de uma função de ativação, geralmente uma função degrau, que determina a saída do modelo. Em problemas de classificação, esta função de ativação é responsável por produzir uma das duas classes.

No contexto da regressão, a função de ativação pode ser omitida, permitindo que o *Perceptron* produza uma saída contínua.

A regra de aprendizado do *Perceptron* é simples e iterativa. Considere um conjunto de exemplos de treinamento $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Para cada exemplo de treinamento, o Perceptron faz uma previsão com base nos pesos atuais. Se a previsão estiver correta, os pesos permanecem inalterados. Caso contrário, os pesos são ajustados de acordo com a seguinte regra de atualização:

$$w := w + \eta(y^{(i)} - \hat{y}^{(i)})x^{(i)},$$

em que , w são os pesos, η é a taxa de aprendizado, $y^{(i)}$ é o valor verdadeiro, e $\hat{y}^{(i)}$ é a previsão.

No contexto do aprendizado incremental, o Perceptron é especialmente atraente devido à sua simplicidade e eficiência. Ele ajusta os pesos com base em cada exemplo de treinamento, permitindo que o modelo se adapte rapidamente às mudanças nos dados. No entanto, como um modelo linear, o Perceptron tem limitações. Ele não pode capturar relações não-lineares sem ser estendido, por exemplo, por meio de camadas adicionais ou transformações não-lineares das entradas. Ainda assim, em muitos cenários de regressão incremental, onde a simplicidade e a eficiência são prioritárias, o Perceptron pode ser uma escolha adequada.

3.2.1.3 *Least Mean Squares* (LMS)

O algoritmo *Least Mean Squares*, comumente conhecido como LMS, é uma técnica simples, porém poderosa, para atualização adaptativa de filtros. Ele surgiu como uma solução para problemas que requeriam otimização iterativa e foi um dos primeiros métodos a oferecer uma abordagem incremental para ajustar modelos lineares (WIDROW; HOFF, 1960).

Em muitas aplicações práticas, os dados podem mudar ao longo do tempo ou podem ser recebidos sequencialmente. Isso requer algoritmos que possam se adaptar dinamicamente a novos dados sem precisar ser re-treinados do zero. O LMS, com sua abordagem iterativa, provou ser uma solução eficiente para tais desafios, especialmente em sistemas de comunicação e processamento de sinal.

Dado um vetor de entrada $\boldsymbol{x(n)}$ e um sinal desejado $d(n)$, um filtro linear tenta produzir uma estimativa $\hat{d}(n)$ tal que o erro $e(n) = d(n) - \hat{d}(n)$ seja minimizado. O algoritmo LMS atualiza os pesos do filtro linear $w(n)$ usando a seguinte regra:

$$w(n+1) = w(n) + \mu e(n)x(n),$$

em que μ é a taxa de aprendizado. A atualização é proporcional ao erro e a entrada, o que significa que se o erro for grande, os ajustes nos pesos serão mais substanciais.

A maior vantagem do LMS é sua simplicidade e capacidade de adaptar-se *online*. Isso o torna particularmente útil para aplicações onde os dados estão mudando dinamicamente. Além disso, o LMS tem convergência garantida sob condições apropriadas da taxa de aprendizado. No entanto, o desempenho do LMS pode ser afetado por escolhas inadequadas da taxa de aprendizado. Uma taxa muito alta pode causar oscilações e uma taxa muito baixa pode resultar em uma convergência muito lenta. Variantes do LMS, como o *Normalized* LMS (NLMS), foram propostas para remediar algumas dessas limitações.

3.2.1.4 *Recursive Least Squares* (RLS)

O algoritmo *Recursive Least Squares*, comumente conhecido como RLS, é uma abordagem adaptativa para resolver problemas de regressão linear de forma incremental. O RLS tem suas raízes na teoria de estimação ótima e foi desenvolvido como uma alternativa ao método tradicional de mínimos quadrados e ao algoritmo LMS. Uma das principais diferenças entre o RLS e o LMS é que, enquanto o LMS atualiza os pesos em uma direção proporcional ao gradiente do erro instantâneo, o RLS atualiza os pesos de maneira a minimizar o erro acumulado ao longo de um horizonte temporal, fazendo uso de uma "janela de tempo". Isto dá ao RLS uma capacidade única de considerar uma história mais rica de erros, tornando-o menos sensível a ruídos e variações (HAYKIN, 2014).

A ideia principal por trás do RLS é atualizar os parâmetros do modelo linear de forma recursiva à medida que novos dados são recebidos, sem a necessidade de reprocessar todos os dados anteriores. Seja $\mathbf{w}$ o vetor de pesos do modelo linear e considere a tarefa de estimar este vetor com base em um novo par de entrada-saída $(\mathbf{x}_t, y_t)$. O RLS atualiza os pesos usando as seguintes fórmulas:

$$\mathbf{e}_t = y_t - \mathbf{w}_{t-1}^T \mathbf{x}_t,$$

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \mathbf{P}_t \mathbf{x}_t \mathbf{e}_t,$$

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t},$$

em que $\mathbf{P}_t$ é a matriz de covariâncias dos pesos no tempo t e $\mathbf{e}_t$ é o erro de predição. O RLS atualiza esta matriz de covariâncias e os pesos de forma iterativa, permitindo assim uma adaptação rápida e eficiente às mudanças nos dados.

Devido à sua natureza adaptativa e sua capacidade de considerar uma história extensiva de erros, o RLS é especialmente útil em cenários em que os dados podem ser não-estacionários ou onde as relações subjacentes entre as entradas e saídas mudam ao longo do tempo. Além disso, ao evitar a necessidade de inversão de matrizes em grande escala, o RLS é computacionalmente eficiente e adequado para aplicações em tempo real. No entanto, o RLS também tem seus desafios. A inicialização da matriz de covariâncias e a escolha de parâmetros, como o fator de esquecimento (em variantes que o utilizam), podem influenciar significativamente o desempenho do algoritmo. Portanto, um ajuste cuidadoso é muitas vezes necessário para obter o melhor desempenho em uma aplicação específica.

3.2.2 Métricas de avaliação de modelo incremental

No contexto de aprendizado incremental, a avaliação do desempenho do modelo é crucial, e, nesse contexto, o *regret* e o *Mean Squared Error* (MSE) emergem como métricas de fundamental importância.

3.2.2.1 *Regret*

O *regret* serve como uma métrica significativa na avaliação de algoritmos de aprendizado incremantal, proporcionando uma medida acumulativa das discrepâncias entre o desempenho do algoritmo e o rendimento ótimo possível de um modelo estático retrospectivo. Ele essencialmente quantifica o 'pesar' por não ter adotado a estratégia mais eficaz à luz dos dados subsequentemente revelados (SHALEV-SHWARTZ, 2012).

Esta métrica é especialmente valiosa em ambientes incertos, objetivando oferecer um critério robusto para avaliar a eficácia dos algoritmos. Um algoritmo que demonstra um *regret* sublinear, onde $\frac{R_T}{T}$

converge para zero quando T se aproxima do infinito, é visto como superior, indicando que, em termos médios, ele se aproxima do desempenho do melhor modelo estático possível (SHALEV-SHWARTZ, 2012).

Formalmente, considere um algoritmo de aprendizado *online* que toma uma sequência de decisões $a_1, a_2, \ldots, a_T$ ao longo de T períodos de tempo. Para cada decisão a_t, existe um custo associado (ou perda) dado por $L_t(a_t)$. O *regret* R_T até o tempo T é definido pela diferença entre o custo total que o algoritmo incorreu e o custo que teria sido acumulado pela melhor ação fixa possível, em retrospecto. Matematicamente, isso é expresso como:

$$R_T = \sum_{t=1}^{T} L_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} L_t(a), \tag{3.1}$$

em que $\mathcal{A}$ representa o conjunto de todas as ações possíveis. O termo à esquerda da subtração indica o custo total que o algoritmo enfrentou, enquanto o termo à direita denota o custo do melhor modelo fixo, olhando em retrospectiva.

Como detalhado na Equação 3.1, o *regret* total R_T é calculado reforçando a ideia de que um algoritmo eficaz deve minimizar este valor ao longo do tempo.

Uma das principais vantagens de utilizar o *regret* como métrica é que ele oferece uma comparação justa entre diferentes algoritmos, sem depender de conhecimento prévio sobre os dados. Além disso, em cenários reais, muitas vezes é impraticável ter um modelo estático perfeito para comparação, tornando o conceito de *regret* ainda mais valioso para avaliar algoritmos online (CESA-BIANCHI; LUGOSI, 2006).

Os limites máximos de *regret* em aprendizado incremental são vitais, atuando como um objetivo teórico para os algoritmos e servindo como uma forma de medir a eficácia do algoritmo *online* diante das condições dadas (SHALEV-SHWARTZ, 2012; CESA-BIANCHI; LUGOSI, 2006)..

A desigualdade de Hoeffding, por exemplo, é uma ferramenta valiosa nesse contexto. Ela fornece uma ligação entre a média das amostras e a expectativa real de uma variável aleatória, o que é útil ao analisar a performance de um algoritmo *online* sob incerteza. Outras técnicas e resultados teóricos, como o Teorema de Minimax e a desigualdade de McDiarmid, também podem ser usados para estabelecer e analisar os limites superiores do *regret*. A escolha da técnica ou resultado a ser utilizado depende em grande parte do algoritmo e do contexto específico em questão.

No caso do gradiente descendente *online*, a rapidez com que o algoritmo converge pode ser fundamental para determinar seu desempenho. Se o *Gradient Descent* convergir rapidamente e o *regret* crescer sublinearmente, pode-se inferir que o algoritmo está se adaptando eficazmente às mudanças nos dados.

Além das técnicas matemáticas, a estrutura do problema de aprendizado *online*, as características dos dados e as premissas feitas sobre o ambiente (como adversarial *vs.* estocástico) também influenciam os limites superiores do *regret*. Assim, ao projetar ou analisar algoritmos *online*, é importante considerar todos esses fatores para obter uma compreensão completa e precisa do desempenho potencial do algoritmo(CESA-BIANCHI; LUGOSI, 2006).

3.2.2.2 *Mean Squared Error* (MSE)

O *Mean Squared Error* (MSE) é uma medida de desempenho fundamental para modelos de predição, sendo calculado como a média dos quadrados das diferenças entre os valores reais e os valores previstos de uma variável, como demonstrado na Equação 3.2:

$$MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{3.2}$$

Em cenários de aprendizado incremental, onde cada nova observação ou lote de observações ajusta o modelo, o MSE é recalculado para abranger todas as observações até o ponto atual t. Esta adaptação é detalhada na Equação 3.3:

$$MSE_t = \frac{1}{t}\sum_{i=1}^{t}(y_i - \hat{y}_i)^2. \tag{3.3}$$

Aqui, y_i representa o valor observado na iteração i, $\hat{y}_i$ indica a previsão do modelo para essa iteração, e t denota o total acumulado de observações.

O *Root Mean Squared Error* (RMSE), a raiz quadrada do MSE, proporciona uma interpretação da magnitude do erro nas mesmas unidades da variável de saída, e é definido conforme a Equação 3.4:

$$RMSE_t = \sqrt{MSE_t}. \tag{3.4}$$

À medida que novos dados são introduzidos, tanto o MSE quanto o RMSE são atualizados para refletir a performance atual do modelo, utilizando técnicas como janelas deslizantes para focar na performance recente, abrangendo apenas as últimas n observações.

3.2.2.3 *Mean Absolute Error* (MAE)

O *Mean Absolute Error* (MAE) é uma métrica de avaliação de modelos de predição que mede a média das diferenças absolutas entre os valores previstos e os valores reais. Esta medida oferece uma avaliação direta da magnitude média dos erros de previsão, sem dar ênfase desproporcional a grandes erros, o que pode ser vantajoso em aplicações com presença de *outliers*. A formulação clássica do MAE é expressa na Equação 3.5:

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{3.5}$$

No contexto de aprendizado incremental, onde cada nova amostra pode modificar o modelo e sua performance, o MAE é recalculado para incorporar todas as observações até o momento atual t, adaptando-se assim às mudanças mais recentes. Esta adaptação é detalhada na Equação 3.6:

$$MAE_t = \frac{1}{t}\sum_{i=1}^{t}|y_i - \hat{y}_i|. \tag{3.6}$$

Aqui, y_i representa o valor real observado na iteração i, $\hat{y}_i$ é a previsão feita pelo modelo na mesma iteração, e t denota o total acumulado de observações consideradas.

Para manter o foco na performance recente em ambientes dinâmicos, técnicas de janelas deslizantes são frequentemente aplicadas ao cálculo do MAE, limitando a análise às últimas n observações. Essa abordagem ajuda a refletir de maneira mais precisa as mudanças na performance do modelo frente a novos dados, garantindo que as métricas sejam atualizadas para refletir a performance mais atual do modelo.

3.3 Aprendizado Incremental em Séries Temporais

O aprendizado incremental em séries temporais apresenta seus próprios desafios e oportunidades. A natureza sequencial dos dados, juntamente com a necessidade de prever eventos futuros com base nas observações mais recentes, torna essencial que os modelos possam se adaptar rapidamente às mudanças nos padrões temporais.

Um marco inicial na incorporação explícita do aprendizado *online* em análises de séries temporais foi estabelecido por Anava et al. em 2013. Este estudo almejou prever séries temporais empregando o modelo ARMA (*Autoregressive Moving Average*), postulando mínimas premissas acerca dos termos de ruído. Os algoritmos formulados se demonstraram competentes para tarefas de previsão, independente da necessidade de os termos de ruído serem Gaussianos, serem identicamente distribuídos ou até serem independentes entre si. A metodologia incremental adotada mostrou que a precisão do modelo tende assintoticamente à precisão do melhor modelo ARMA fixo, retrospectivamente (ANAVA et al., 2013). Posteriormente, em 2016, Liu et al. desenvolveram técnicas para a inferência do modelo ARIMA incremental, adotando igualmente suposições mínimas, e evidenciaram que a eficácia dos modelos inferidos convergem, de forma assintótica, para a eficácia do melhor modelo ARIMA fixo (LIU et al., 2016). Em 2021, SHAO et al., desenvolveram técnicas para o ajuste do modelo ARIMA incremental sem a necessidade da seleção de hiperparâmetros a priori. Mais recentemente, em 2023, Li et al. conceberam uma estrutura para estimar os parâmetros do ARIMA incremental em cenários predominantemente impactados por ruído (LI; WU; LIU, 2023).

A incorporação do aprendizado *online* em modelos de séries temporais não se restringe apenas aos modelos ARMA e ARIMA. Redes neurais, particularmente *Long Short-Term Memory* (LSTM) e Redes Neurais Recorrentes (RNN), foram ajustadas para abordagens de aprendizado *online*, dado que sua arquitetura é inerentemente adequada para lidar com sequências. O campo continua evoluindo, com novas técnicas e abordagens sendo desenvolvidas para abordar os desafios específicos do aprendizado *online* em séries temporais. A capacidade de adaptar-se rapidamente e aprender de novos dados à medida que são gerados é de imenso valor, especialmente em aplicações em tempo real onde a reação rápida às mudanças é crucial.

3.3.1 Modelo ARMA

O modelo ARMA, uma fusão de dois modelos *AutoRegressive* (AR) e *Moving Averages* (MA), é frequentemente empregado na análise de séries temporais para modelar séries univariadas (BOX et al., 2015; MORETTIN; TOLOI, 2018).

No modelo $ARMA(p, q)$, a parte autoregressiva retrata a variável atual como uma composição linear de seus valores anteriores. O termo p refere-se à ordem do componente autoregressivo, representando o número de *lags* (defasagens) incluídos no modelo. Em termos matemáticos, o componente $AR(p)$ pode ser expresso como:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \varepsilon_t.$$

O componente de média móvel do modelo $ARMA(p, q)$ representa a variável da série temporal como uma combinação linear dos termos de erros anteriores. A ordem q do componente representa o número de termos de erros anteriores considerados no modelo. Este componente pode ser matematicamente

representado como:

$$X_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + ... + \theta_q\varepsilon_{t-q}.$$

Ao juntar os componentes AR e MA, o modelo ARMA(p, q) é descrito por:

$$X_t = c + \phi_1 X_{t-1} + ... + \phi_p X_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + ... + \theta_q\varepsilon_{t-q}.$$

O ARMA é especialmente eficaz para modelar várias estruturas de autocorrelação em séries temporais, sendo particularmente apto para séries temporais estacionárias. No entanto, para séries com tendências ou sazonalidades, modelos como o ARIMA, que integra termos de diferenciação para atingir estacionariedade, são mais adequados (BOX et al., 2015; MORETTIN; TOLOI, 2018)..

Anava et al. (2013) propuseram uma abordagem incremental para modelos ARMA, na qual os coeficientes ϕ e θ são ajustados iterativamente à medida que novos dados são recebidos. O objetivo central desta abordagem é minimizar a função de perda de maneira dinâmica. A minimização é alcançada atualizando os coeficientes usando uma variação do método de *Gradient Descent*.

$$\begin{aligned} \phi_i^{(t+1)} &= \phi_i^{(t)} - \eta\nabla_{\phi_i} J_t, \\ \theta_i^{(t+1)} &= \theta_i^{(t)} - \eta\nabla_{\theta_i} J_t, \end{aligned}$$

em que:

- J_t é a função de perda no tempo t.
- $\nabla_{\phi_i} J_t$ e $\nabla_{\theta_i} J_t$ representam os gradientes da função de perda em relação aos coeficientes ϕ_i e θ_i, respectivamente.
- η é a taxa de aprendizado.

A metodologia introduzida por Anava et al. (2013) é caracterizada pela sua operação sob condições mínimas quanto aos termos de ruído. Não é essencial que os erros ϵ_t sigam uma distribuição Gaussiana, sejam identicamente distribuídos ou independentes entre si. Vale destacar que Anava et al. (2013) define um limite superior para o *regret* do algoritmo ARMA *online*, evidenciando que a eficácia do algoritmo converge assintoticamente ao desempenho do modelo ARMA ótimo em retrospecto. Este limiar é representado como $O(GD\sqrt{T})$, onde G, D, e T são parâmetros que estão vinculados especificamente ao algoritmo e ao conjunto de dados em questão.

O principal desafio desta metodologia reside em assegurar a convergência e estabilidade do modelo, especialmente considerando a natureza iterativa e potenciais mudanças na estrutura subjacente da série temporal. Contudo, esta abordagem do modelo ARMA, apoiada em técnicas de aprendizado *online*, oferece uma adaptação contínua ao fluxo de informações, tornando-a valiosa em ambientes dinâmicos onde os padrões podem evoluir.

3.3.2 Modelo ARIMA

O modelo ARIMA (AutoRegressive Integrated Moving Average) é uma extensão do modelo ARMA e é frequentemente aplicado em séries temporais que apresentam tendências não estacionárias. A principal adição é o componente de integração, que diferencia a série temporal para torná-la estacionária (BROCKWELL; DAVIS, 2016; BOX et al., 2015; MORETTIN; TOLOI, 2018).

No modelo $ARIMA(p, d, q)$, temos:

- Diferenciação (I) - a diferenciação é o processo de subtrair observações consecutivas em uma série temporal, com o objetivo de tornar a série estacionária. O termo "d"no ARIMA(p, d, q) refere-se à ordem de diferenciação, ou seja, o número de vezes que a série é diferenciada para alcançar estacionariedade. Uma série diferenciada d vezes é denotada como $\nabla^d X_t$, onde ∇ é o operador de diferenciação.

- Modelo Autoregressivo (AR) - como no modelo ARMA, o componente autoregressivo em um modelo ARIMA descreve a variável da série temporal em função de seus valores passados. Assim, o componente AR(p) é expresso da mesma forma:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t.$$

- Modelo de Média Móvel (MA) - o componente de média móvel também permanece inalterado no ARIMA, sendo representado como:

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}.$$

Combinando todos os componentes, o modelo $ARIMA(p, d, q)$ pode ser descrito como:

$$\nabla^d X_t = c + \phi_1 \nabla^d X_{t-1} + \ldots + \phi_p \nabla^d X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}.$$

O ARIMA é amplamente reconhecido por sua capacidade de modelar séries temporais com tendências, tornando-se uma escolha versátil e poderosa para muitas aplicações de previsão em diversas disciplinas (BROCKWELL; DAVIS, 2016; BOX et al., 2015; MORETTIN; TOLOI, 2018).

Na abordagem incremental proposta por (LIU et al., 2016), os coeficientes ϕ e θ são ajustados de maneira semelhante ao modelo ARMA. O foco desta abordagem é minimizar a função de perda de forma dinâmica, o que é concretizado através da atualização iterativa dos coeficientes, utilizando uma variação do método de Gradiente Descendente:

$$\begin{aligned} \phi_i^{(t+1)} &= \phi_i^{(t)} - \eta \nabla_{\phi_i} J_t, \\ \theta_i^{(t+1)} &= \theta_i^{(t)} - \eta \nabla_{\theta_i} J_t, \end{aligned}$$

em que:

- J_t é a função de perda no tempo t.

- $\nabla_{\phi_i} J_t$ e $\nabla_{\theta_i} J_t$ representam os gradientes da função de perda em relação aos coeficientes ϕ_i e θ_i, respectivamente.

- η é a taxa de aprendizado.

O desempenho do modelo ARIMA incremental converge para o melhor modelo ARIMA em retrospecto (LIU et al., 2016).

4 Material e Métodos

Este capítulo detalha a metodologia empregada para avaliar o poder preditivo e adaptabilidade dos modelos ARIMA incremental e ARIMA tradicional. A escolha metodológica visa fornecer um panorama sobre a aplicabilidade e o desempenho desses modelos em condições de mercado dinâmicas e voláteis.

O modelo ARIMA incremental foi ajustado utilizando a biblioteca `River` (MONTIEL et al., 2021) da linguagem Python (ROSSUM; JR, 1995).Os parâmetros do modelo foram fixados à *priori*, com parâmetros $p = 1$, $d = 1$, $q = 1$, baseando-se em várias considerações estratégicas. Primeiramente, a configuração $1, 1, 1$ representa uma estrutura de modelo simplificada, mas amplamente aplicável, que é capaz de capturar a autocorrelação de primeira ordem, tendências estacionárias e o efeito médio do ruído branco nos dados, procurando equilibrar a complexidade do modelo e o sobreajuste servindo como uma linha de base para demais estudos comparativos.

Os parâmetros do modelo ARIMA tradicional foram selecionados com o uso do algoritmo Auto ARIMA e reajustados em janelas móveis. Essa estratégia proporciona um estudo comparativo justo, onde ambas as abordagens são testadas quanto à sua eficácia em adaptar-se e prever dentro de um mesmo contexto de mercado.

Este enfoque proporciona uma plataforma para investigar se ajustes mais simples e diretos em modelos ARIMA podem competir ou superar estratégias mais complexas de seleção automática de parâmetros, especialmente em ambientes de mercado que exigem respostas rápidas às mudanças.

4.1 Estudo de Simulação

O estudo de simulação foi utilizado para avaliar analiticamente a adaptabilidade e poder preditivo do modelo ARIMA incremental, através da investigação do comportamento do *regret* associado ao modelo ARIMA incremental em um espectro de cenários simulados que incorporam diversas condições de *drifts*.

A hipótese central é averiguar se o *regret* do modelo ARIMA incremental, definido como a diferença acumulada entre os erros de previsão do modelo e um *benchmark* ideal, cresce de maneira sublinear com o aumento do número de iterações, indicativo de uma adaptabilidade eficaz do modelo em resposta a condições de *drifts* variadas.

4.1.1 Parâmetros de Simulação

Visando garantir a alinhamento dos parâmetros de simulação com cenários realísticos, desenvolveu-se um método sistemático e automatizado para extrair esses parâmetros dos dados históricos dos preços do petróleo Brent. Esse procedimento foi efetuado na linguagem de programação `R`, com o apoio das bibliotecas `tidyverse` (WICKHAM; RSTUDIO, 2023) e `forecast` (HYNDMAN et al., 2024), que são, respectivamente, reconhecidas por suas capacidades de manipulação de dados e análise de séries temporais.

Inicialmente, os dados históricos de preços do petróleo *Brent* foram carregados a partir de um arquivo CSV, onde a coluna de datas foi devidamente convertida para o formato de data. Posteriormente, definiu-se uma função `extract_arima_params` com o objetivo de percorrer a série temporal em janelas de tamanho fixo, aplicando o modelo ARIMA a cada uma delas para extrair seus parâmetros.

A função foi projetada para iterar sobre a série temporal utilizando um tamanho de janela (`window_size`) de 30 dias e um passo (`step_size`) de 1 dia. Para cada janela de dados, o modelo

ARIMA foi ajustado automaticamente por meio da função `auto.arima`, que determina os melhores parâmetros (p, d, q) para o modelo. Em seguida, os coeficientes do modelo ajustado, a variância dos resíduos (`sigma2`), as datas de início e término da janela, e a ordem de diferenciação (d) foram extraídos e armazenados.

Ao concluir a iteração sobre toda a série temporal, os parâmetros ARIMA extraídos foram então consolidados em um *dataframe* final. Este *dataframe* contém as colunas necessárias para representar os coeficientes ARIMA (p, d, q), a variância dos erros, e as datas de início e término de cada janela analisada. Por fim, os parâmetros extraídos foram exportados para um novo arquivo CSV, facilitando sua utilização em análises subsequentes de aprendizado incremental.

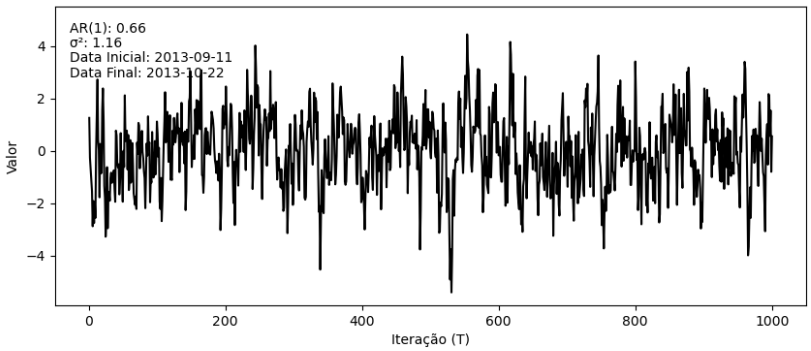
4.1.2 Geração das Séries Temporais Base

Para a geração das séries temporais base, adotou-se um procedimento de simulação computacional na linguagem de programação `R`, com o suporte das bibliotecas `tidyverse` (WICKHAM; RSTUDIO, 2023), `simts` (GUERRIER et al., 2023), e `forecast` (HYNDMAN et al., 2024). Este método visou criar séries temporais sintéticas que mimetizam as propriedades estatísticas de séries temporais reais, sem incorporar tendências ou desvios sistemáticos.

A reprodutibilidade dos resultados foi assegurada desde o início, definindo uma semente fixa (`set.seed(1337)`) para a geração de números pseudoaleatórios. Subsequentemente, procedeu-se à leitura dos parâmetros ARIMA, que haviam sido extraídos anteriormente e armazenados em um arquivo CSV. Com base nesses parâmetros, iniciou-se a execução iterativa do procedimento de simulação, visando a criação de 1000 séries temporais sintéticas. Em cada ciclo da simulação, selecionou-se aleatoriamente um conjunto de parâmetros ARIMA, prosseguindo apenas se verificada a presença de coeficientes AR, para garantir a produção de séries que adequadamente representassem dinâmicas temporais específicas.

Para cada iteração, os coeficientes AR e MA, a ordem de diferenciação e a variância dos termos de erro (σ^2) foram integrados na simulação, utilizando-se a função `gen_arima` da biblioteca `simts` para gerar séries de 1000 observações que refletissem os parâmetros ARIMA designados, como mostra a Figura 1

Figura 1 – Série temporal Sintética.



Finalmente, cada série temporal sintetizada, juntamente com seus parâmetros ARIMA correspondentes, foi armazenada em diretórios sequencialmente nomeados, indicativos de cada simulação realizada, consolidando os resultados em arquivos CSV e arquivos de metadados para facilitar análises subsequentes.

4.1.3 Geração das Séries Temporais Com *drifts*

A geração de séries temporais com a introdução de *drifts* foi realizada para simular variações dinâmicas no comportamento das séries temporais base, sem qualquer tipo de *drifts* prévio. A implementação foi feita em *Python*, utilizando uma função específica `aplicar_drift`, projetada para modificar uma série temporal existente de acordo com diferentes tipos de *drifts*: abrupto, incremental e recorrente.

Durante a geração de séries temporais com *drifts*, foram utilizados certos parâmetros para simular diferentes dinâmicas de mudança, conforme os tipos de *drifts* aplicados. Os parâmetros detalhados a seguir fornecem uma compreensão das configurações usadas para introduzir variações nas séries temporais:

- ***drifts* Abrupto:**
  - *Delta:* um valor fixo adicionado à série temporal a partir de pontos específicos, definindo uma mudança abrupta. No estudo, um delta de 5 unidades foi usado para simular essa mudança repentina.
  - *Número de Pontos:* dois pontos de *drifts* foram escolhidos aleatoriamente para a aplicação do *drifts* abrupto, introduzindo duas mudanças significativas na série temporal.
- ***drifts* Incremental:**
  - *Duração:* o período durante o qual o *drifts* incremental é aplicado. A duração do *drifts* foi definida aleatoriamente para cada série, variando para simular uma introdução gradual de mudança.
  - *Incremento Base:* representa a magnitude base de mudança por unidade de tempo. Junto com variações adicionais, define como a série temporal se altera incrementalmente ao longo do período de *drifts*.
  - *Amplitude da Variação:* a amplitude da componente sinusoidal adicionada ao incremento base, introduzindo uma variação no ritmo de mudança ao longo do tempo.
  - *Frequência da Variação:* determina a frequência da oscilação sinusoidal aplicada ao incremento, impactando diretamente no padrão de variação da mudança incremental.
- ***drifts* Recorrente:**
  - *Período:* a distância entre os picos da função senoidal utilizada para simular o *drifts* recorrente. Valores aleatórios foram escolhidos para cada série, visando diversificar as características de oscilação.
  - *Amplitude:* a magnitude das oscilações introduzidas pelo *drifts* recorrente. A amplitude também foi definida aleatoriamente, resultando em variações na intensidade das oscilações observadas nas séries temporais.

Cada tipo de *drifts* foi projetado para testar a resposta dos modelos sob diferentes cenários de mudança, variando de alterações súbitas a mudanças graduais e padrões oscilatórios. Esses parâmetros fornecem um meio robusto de simular condições realistas e desafiadoras, essenciais para avaliar a adaptabilidade de modelos temporais em prever séries dinâmicas.

4.1.4 Avaliação dos Modelos sob Condições Variadas

4.1.5 Modelo ARIMA Incremental

O modelo ARIMA foi configurado com parâmetros $p = 1$, $d = 1$, $q = 1$, acompanhados por um regressor composto por uma padronização (`StandardScaler`) seguida de uma regressão linear (`LinearRegression`) com otimização via *AdaGrad* e taxa de aprendizado de 0.1. Este modelo foi avaliado pela sua capacidade de *nowcasting* (previsão do ponto $t+1$) e pelo cálculo do RMSE incremental, fundamentado no poder preditivo do *nowcast*.

4.1.6 Modelo ARIMA com Ajuste por Janelas Móveis

Para o modelo ARIMA tradicional, o processo de ajuste inicial foi realizado após a acumulação das primeiras 10 observações. Subsequentemente, o modelo foi reajustado a cada nova observação. A janela de dados considerada para cada ajuste do modelo abrangeu os últimos 30 pontos disponíveis ou o total de pontos antes do primeiro ajuste. O Auto ARIMA foi configurado para explorar automaticamente combinações de parâmetros dentro dos limites de $p = 1$ a 3 e $q = 1$ a 3, sem considerar sazonalidade ($m = 0$), e optando por um processo de seleção passo a passo.

O *nowcast* produzido por cada modelo e os respectivos valores de RMSE, baseados na acurácia do *nowcast*, foram registrados a fim de permitir uma análise comparativa direta do desempenho preditivo dos modelos em diferentes cenários de *drifts*.

4.1.7 Cálculo do *regret* nas Simulações

O conceito de *regret* foi empregado como uma métrica para avaliar e comparar o desempenho acumulado dos modelos ao longo do tempo, refletindo a qualidade das decisões de previsão em face das incertezas. O cálculo do *regret* foi implementado através da comparação dos erros de *nowcasting* produzidos por ambos os modelos em cada ponto da série temporal. A função `calcular_regret` desempenha essa análise, seguindo os passos detalhados abaixo:

1. o *regret* acumulado é calculado iterativamente ao longo da série temporal, onde para cada ponto, o *regret* instantâneo é determinado pela diferença absoluta entre os erros de *n*owcasting do modelo ARIMA incremental e do modelo ARIMA tradicional. Isso reflete a penalidade ou o custo adicional incorrido por optar pelo modelo incremental em detrimento do Auto ARIMA em cada passo de previsão.

2. os valores de *regret* acumulados foram registrados, permitindo uma análise da evolução do desempenho relativo dos modelos ao longo do tempo.

4.1.8 Análise Pós-Simulação do Comportamento do *regret* e RMSE

Após a conclusão das simulações, uma análise detalhada do comportamento do *regret* acumulado foi realizada para cada cenário considerado a fim de verificar se o crescimento médio e mediano do *regret* se mantém sublinear em relação ao aumento do número de iterações (T).

A média e a mediana do *regret* acumulado foram então calculadas, juntamente com os limites inferior e superior baseados nos quantis de 2,5% e 97,5%, respectivamente. Estas estatísticas descrevem

a distribuição central e a variabilidade do *regret* ao longo das simulações, permitindo uma avaliação da tendência geral do crescimento do *regret*.

A análise da distribuição do RMSE ao longo das simulações ofereceu informações adicionais sobre a consistência e a confiabilidade dos modelos sob diferentes condições de mudança nos dados.

Os resultados foram visualizados graficamente: um gráfico de boxplot apresentando o RMSE sob as diferentes condições de simulação para cada modelo e um gráfico que ilustra o *regret* acumulado (mediano e médio) em função do número de iterações. O envelope representando os quantis de 2,5% a 97,5% oferece uma visão da variabilidade do *regret* entre as simulações. A linha $y = x$ foi incluída como referência para o crescimento linear.

4.2 Ajuste e Avaliação de Modelos em Dados Reais

Para a aplicação prática e a validação dos modelos em condições reais, a série temporal do *Europe Brent Spot Price* foi utilizada. Caracterizada por sua volatilidade e relevância econômica, esta série proporciona um cenário ideal para avaliar a eficácia e a adaptabilidade dos modelos ARIMA e ARIMA incremental, os quais foram ajustados conforme a metodologia empregada no estudo de simulação.

4.2.1 Avaliação dos Modelos

Os modelos foram avaliados através das métricas: MAE e RMSE, ambas calculadas a partir das diferenças entre os valores previstos(*nowcast*) e os valores reais da série temporal do *Europe Brent Spot Price*.

O MAE foi calculado para ambos os modelos, fornecendo uma medida direta da magnitude média dos erros de previsão, independentemente da direção. Valores mais baixos de MAE indicam um melhor desempenho preditivo.

De forma similar, o RMSE foi computado para avaliar a precisão das previsões. Diferentemente do MAE, o RMSE dá maior peso a erros maiores, sendo particularmente útil em situações onde é crítico evitar grandes desvios nas previsões.

4.2.2 Análise de Impacto da Volatilidade

Adicionalmente, investigou-se como variações na volatilidade dos preços impactam o desempenho dos modelos. Para isso, a série temporal foi segmentada em diferentes janelas de volatilidade, classificadas em categorias de "Baixa", "Média"e "Alta". Essa classificação baseou-se nos desvios padrões dos retornos diários do preço, calculados em janelas móveis de tamanho variável (HULL, 2018). As categorias foram definidas como segue: "Baixa"para volatilidade menor que o quantil de $1/3$, "Média"para volatilidade entre os quantis de $1/3$ e $2/3$, e "Alta"para volatilidade acima do quantil de $2/3$. Tanto o RMSE quanto o MAE foram calculado separadamente para cada categoria de volatilidade e para diferentes tamanhos de janelas, sendo os resultados apresentados graficamente.

Você pode adicionar uma seção específica no final do capítulo de Material e Métodos para destacar a disponibilidade do código no GitHub da seguinte maneira:

4.3 Disponibilidade do Código no GitHub

Todo o código utilizado neste estudo está disponível publicamente no GitHub para garantir a transparência, replicabilidade e reutilização dos métodos e análises realizadas. O repositório contém os scripts em Python e R utilizados para ajustar os modelos ARIMA incremental e tradicional, realizar o estudo de simulação, gerar as séries temporais sintéticas, aplicar os drifts e avaliar o desempenho dos modelos em dados reais.

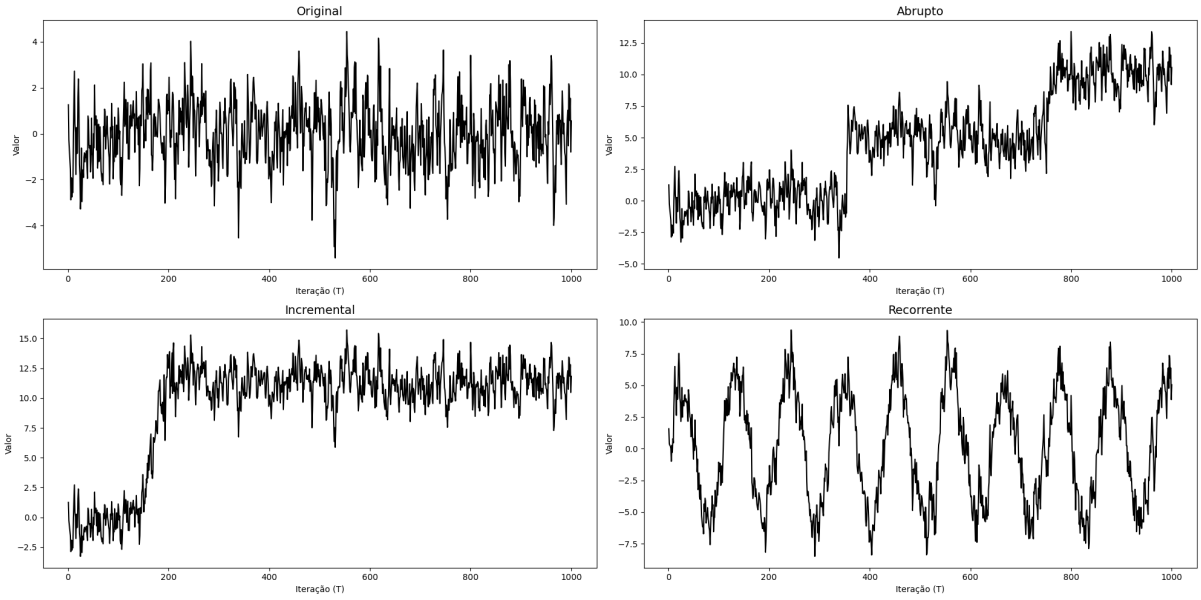
O código pode ser acessado no seguinte repositório do GitHub: <https://github.com/jpolnasc/estatistica.tcc>

5 Resultados

5.1 Análise dos Resultados de Simulação

Na Figura 2, são apresentados exemplos de séries temporais simuladas em quatro cenários distintos: séries estacionárias (original), séries com *drift* abrupto, séries com *drift* incremental e séries com *drift* recorrente. Estes cenários foram criados para testar a adaptabilidade e eficiência dos modelos em situações variadas, refletindo as complexidades encontradas em dados do mundo real.

Figura 2 – Série temporal Sintética e *drifts*.



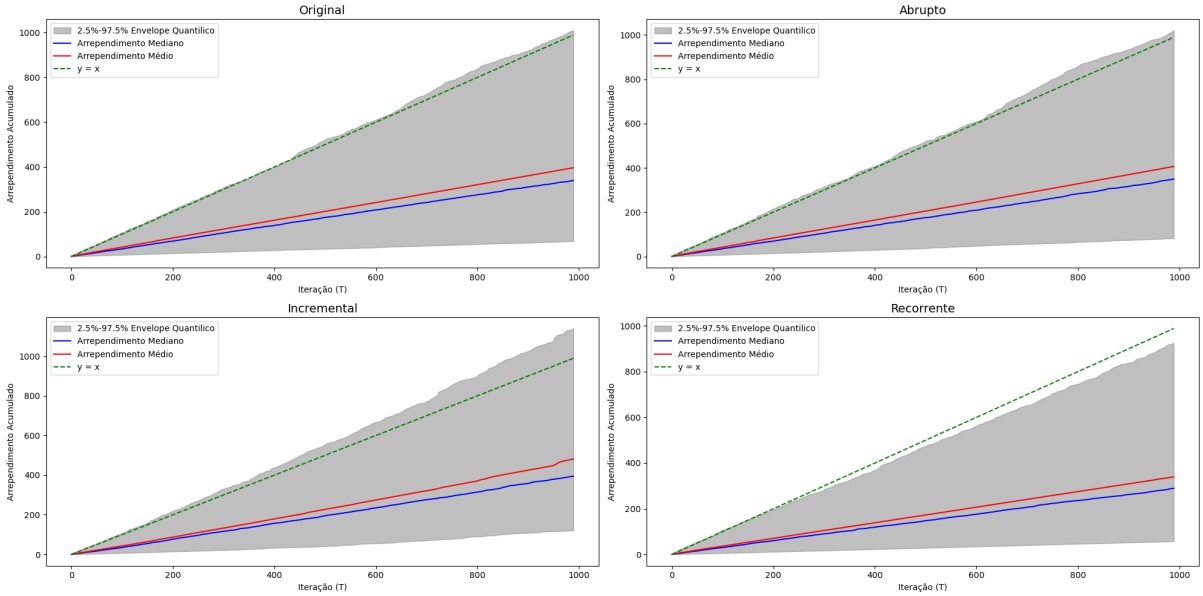
O *regret* acumulado, calculado a partir da comparação entre o modelo ARIMA incremental e o modelo ARIMA com janelas móveis de 30 unidades de tempo, é detalhado na Figura 3 para cada um dos quatro cenários examinados. O *regret* aqui representa o custo adicional de utilizar o modelo incremental em comparação ao modelo com janelas móveis, que serve como *benchmark* ideal neste contexto.

O modelo ARIMA incremental é projetado para adaptar-se continuamente às mudanças nos padrões dos dados, um aspecto fundamental no manejo de séries temporais que apresentam características dinâmicas, como as observadas nos cenários de nossa simulação. Por outro lado, o modelo ARIMA com janelas móveis, que recalibra-se com base nos dados mais recentes dentro de um intervalo fixo, foi utilizado como *benchmark* no cálculo do *regret*.

A hipótese central investigada neste estudo é se o *regret* do modelo ARIMA incremental apresenta um crescimento sublinear em relação ao número de iterações. No entanto, a análise dos resultados, conforme ilustrada na Figura 3, indica que tanto o *regret* mediano quanto o médio não demonstram um crescimento sublinear, mas sim um crescimento linear com um coeficiente angular significativamente menor que $\alpha = 1$. Este crescimento linear mais suave sugere que o modelo ARIMA incremental consegue manter um aumento controlado no *regret* com o aumento no número de iterações T, o que é em si um indicativo de um desempenho relativamente bom mesmo em cenários com diferentes tipos de *drifts*.

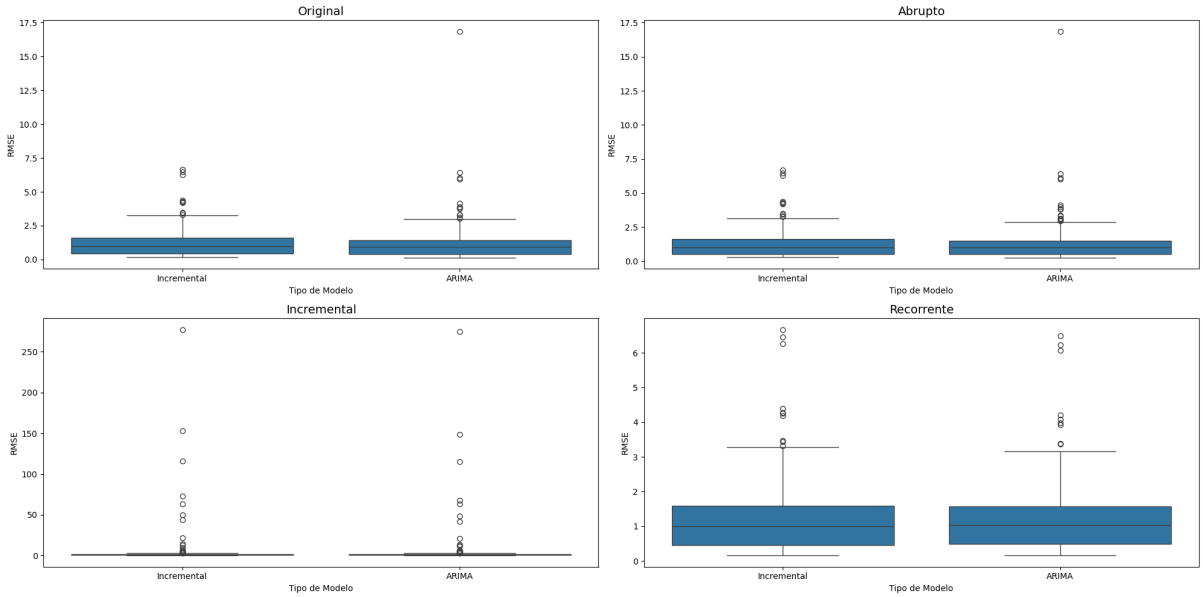
A Figura 4, apresenta o RMSE dos modelos utilizados na simulação para cada tipo de *drifts*. Observa-se que os modelos Incremental e ARIMA exibem distribuições de RMSE comparativamente

Figura 3 – *Regret* acumulado.



similares. Especificamente no cenário de *drift* incremental, destaca-se que existem iterações dentro do estudo de simulação onde ambos modelos registram valores atípicos significativamente elevados em relação à mediana da distribuição. Esses valores atípicos sugerem momentos específicos de desempenho diminuído, que podem ser atribuídos a mudanças graduais nos dados que afetam adversamente o modelo. A presença desses *outliers* enfatiza a importância de considerar a robustez de qualquer modelo em face de variações incrementais, reiterando a necessidade de estratégias de modelagem que possam se adaptar a tais mudanças com eficácia.

Figura 4 – Comparação de desempenho entre os modelos ARIMA e ARIMA Incremental sob diferentes cenários de mudança nos dados.



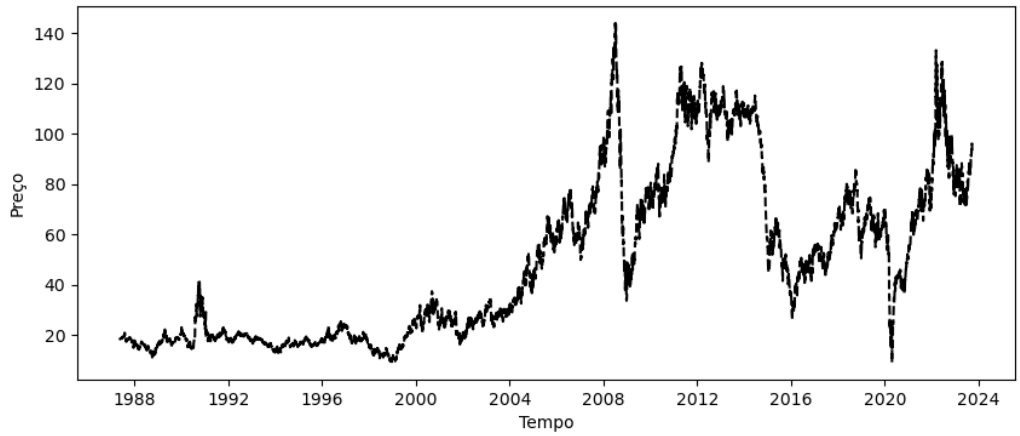
Portanto, considerando tanto a distribuição de RMSE quanto o padrão de crescimento do *regret* acumulado, é evidente que modelos adaptativos como o ARIMA Incremental demonstram aptidão para ambientes dinâmicos. Nestes contextos, as características das séries temporais podem sofrer alterações rápidas ou evoluir gradualmente. Eles não só mantêm a precisão preditiva diante da variabilidade dos dados mas também asseguram uma maior eficiência operacional, o que é crucial em aplicações práticas onde as condições evoluem continuamente.

5.2 Avaliação dos Modelo em Dados Reais

5.2.1 Descrição da Série

Na Figura 5, é apresentada a série temporal do *Europe Brent Spot Price*, um indicador vital no mercado global de energia. Este preço, que reflete o valor à vista do petróleo *Brent*, serve como uma referência global essencial para os preços do petróleo bruto, influenciando significativamente as decisões econômicas e políticas ao redor do mundo.

Figura 5 – Série temporal: Europe Brent Spot Price.



A série evidencia diferentes ciclos de preços, marcados por períodos de aumento, estabilidade e declínio, refletindo uma variedade de fatores econômicos, geopolíticos e tecnológicos que afetam a oferta e demanda global de petróleo.

Inicialmente, observa-se uma tendência forte de aumento até o ano de 2008. Este período foi caracterizado por um crescimento econômico robusto, especialmente em mercados emergentes, que impulsionou a demanda por petróleo. No entanto, a crise financeira global de 2008 desencadeou uma queda abrupta nos preços, refletindo uma desaceleração dramática na atividade econômica mundial e, consequentemente, na demanda por petróleo (ZHANG; YU; WANG, 2009; ZAVADSKA; MORALES; COUGHLAN, 2020).

Em 2009, os preços começaram a se recuperar, impulsionados por medidas de estímulo econômico em várias grandes economias, que ajudaram a restaurar a confiança dos investidores e a demanda por energia. Esta recuperação estendeu-se até meados de 2015, período em que os preços mostraram relativa estabilidade. Este equilíbrio foi mantido por um delicado balanço entre oferta e demanda, apesar das

tensões geopolíticas em regiões produtoras de petróleo e das mudanças nas políticas energéticas de diversos países (ZAVADSKA; MORALES; COUGHLAN, 2020; ZHANG; ZHANG, 2015)

No entanto, em 2016, os preços experimentaram uma forte queda devido a uma sobreoferta global, em parte resultado do aumento da produção de petróleo de xisto nos Estados Unidos e da decisão da OPEP de não reduzir a produção. Essa queda foi também exacerbada por preocupações com o crescimento econômico global (HAMEED ASMAA NORI, 2023; ZAVADSKA; MORALES; COUGHLAN, 2020).

A partir de então, até meados de 2020, os preços começaram a se recuperar gradualmente, à medida que o mercado se ajustava e as tensões em algumas áreas produtoras diminuíam. Essa recuperação, no entanto, foi interrompida abruptamente pela pandemia de COVID-19, que causou uma das maiores quedas na demanda por petróleo na história, levando a uma queda drástica nos preços (HAMEED ASMAA NORI, 2023; ZAVADSKA; MORALES; COUGHLAN, 2020).

Após essa queda em 2020, houve um rápido crescimento até 2023, impulsionado pela recuperação econômica global e pela retomada da demanda por petróleo. No entanto, este período de crescimento foi seguido por uma nova queda, refletindo as incertezas contínuas no mercado global, incluindo as preocupações com a sustentabilidade da recuperação econômica e os impactos de transições energéticas em direção a fontes mais limpas e renováveis (SUN et al., 2023; ZAVADSKA; MORALES; COUGHLAN, 2020).

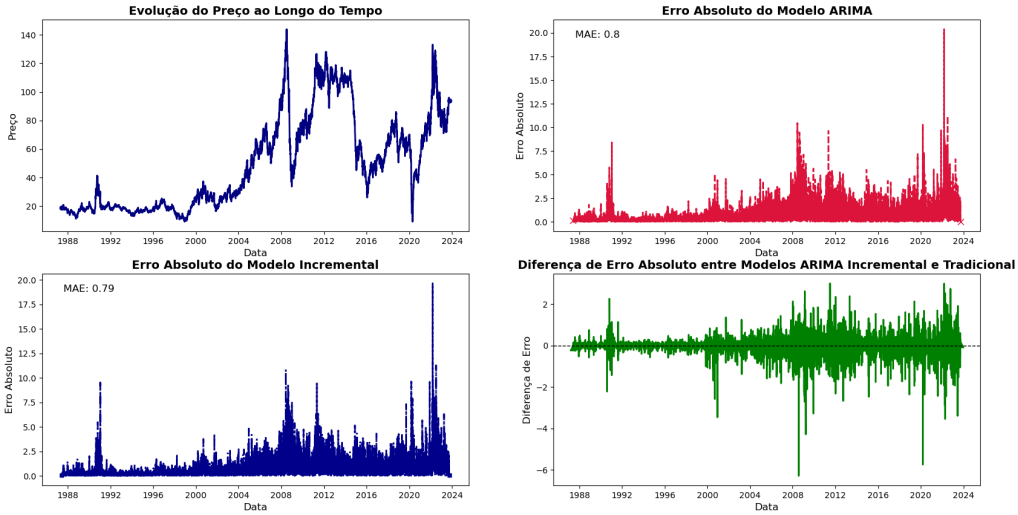
Em resumo, a série temporal do *Europe Brent Spot Price* captura a complexa dinâmica entre a economia global, políticas energéticas e desenvolvimentos geopolíticos. Cada ciclo de preço reflete uma combinação única de fatores, oferecendo perspectivas valiosas sobre as tendências do mercado global de energia e as forças que moldam o mundo contemporâneo.

5.2.2 MAE

Na Figura 6, é apresentada a análise do Erro Absoluto Médio (MAE) para o modelo ARIMA tradicional e o modelo ARIMA incremental, juntamente com a diferença de erros entre eles, utilizando a série temporal do *Europe Brent Spot Price* como referência. Observa-se um comportamento similar dos dois modelos ao longo do tempo, demonstrando a consistência de seus desempenhos durante o período avaliado. O MAE médio, calculado ao longo de todo o período, foi de 0,80 para o modelo ARIMA tradicional e de 0,79 para o modelo ARIMA incremental.

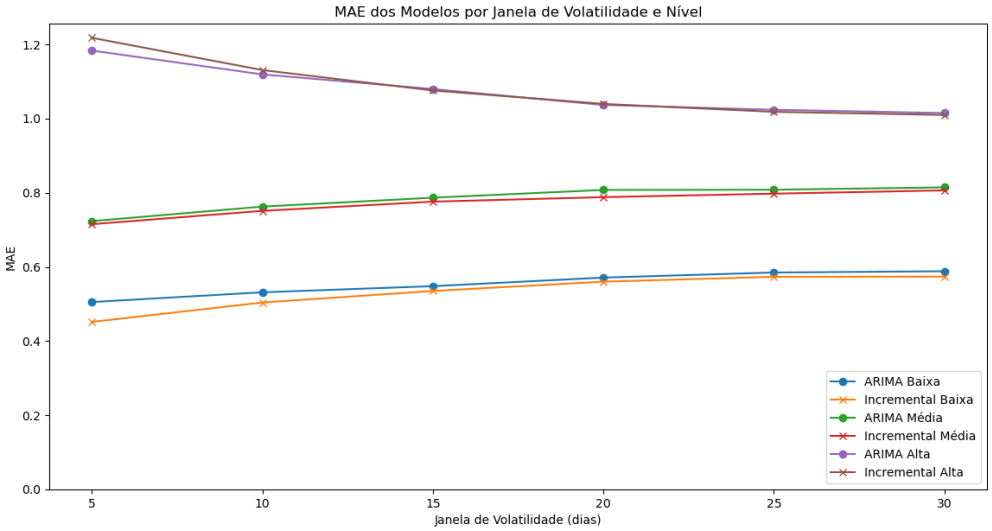
Esta comparação indica que, apesar das pequenas variações na performance dos modelos em momentos específicos, ambos conseguiram fornecer previsões com precisão quase idêntica em relação aos preços do petróleo *Brent*. A proximidade dos valores de MAE reflete a eficácia de ambas as abordagens de modelagem em capturar e prever as tendências da série temporal analisada, com o modelo incremental apresentando uma ligeira vantagem em termos de precisão.

Figura 6 – MAE: Arima e Arima Incremental.



O MAE dos modelos, analisado em função das janelas de tempo e do nível de volatilidade, é apresentado na Figura 7, em que o eixo x representa a janela de volatilidade em dias (5, 10, 15, 20, 25, 30), e o eixo y, o MAE.

Figura 7 – MAE por Janela e Nível de Volatilidade.



Pode-se observar que:

- a alta volatilidade está associada aos valores mais elevados de MAE, enquanto a volatilidade baixa corresponde aos menores valores de MAE para ambos os modelos. Este padrão sugere que a precisão das previsões é significativamente desafiada em condições de alta volatilidade, destacando a sensibilidade dos modelos a flutuações acentuadas no mercado.

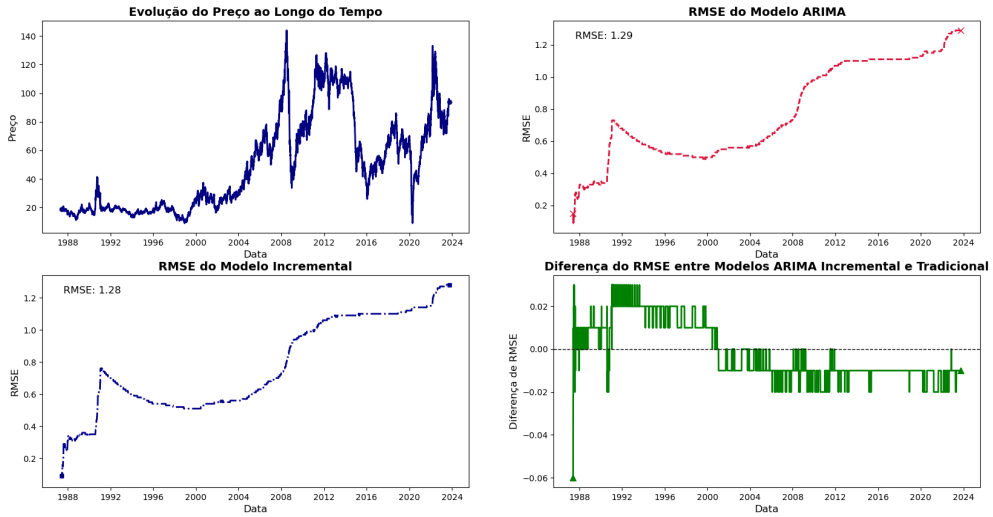
- é observado que, em ambos os modelos, o MAE tende a diminuir progressivamente à medida que a janela de volatilidade aumenta em cenários de alta volatilidade. Contudo, em condições de volatilidade baixa e média, a extensão da janela de volatilidade não promove uma redução consistente no MAE. Este comportamento indica uma adaptação diferenciada dos modelos em resposta às variações na escala de análise da volatilidade, apontando para as nuances de como cada nível de volatilidade impacta a acurácia das previsões.

- o modelo ARIMA incremental apresentou um MAE ligeiramente inferior em comparação ao ARIMA tradicional em todas as janelas de volatilidade consideradas, em especial nos níveis de volatilidade baixa e média. Esta observação sugere uma vantagem marginal do modelo incremental na manutenção da precisão das previsões em ambientes menos voláteis.

5.2.3 RMSE

Na Figura 8, é apresentada a série temporal do *Europe Brent Spot Price* e comparamos o RMSE entre os modelos ARIMA tradicional e incremental, além de analisar a diferença de RMSE entre eles.

Figura 8 – RMSE: ARIMA e ARIMA Incremental.

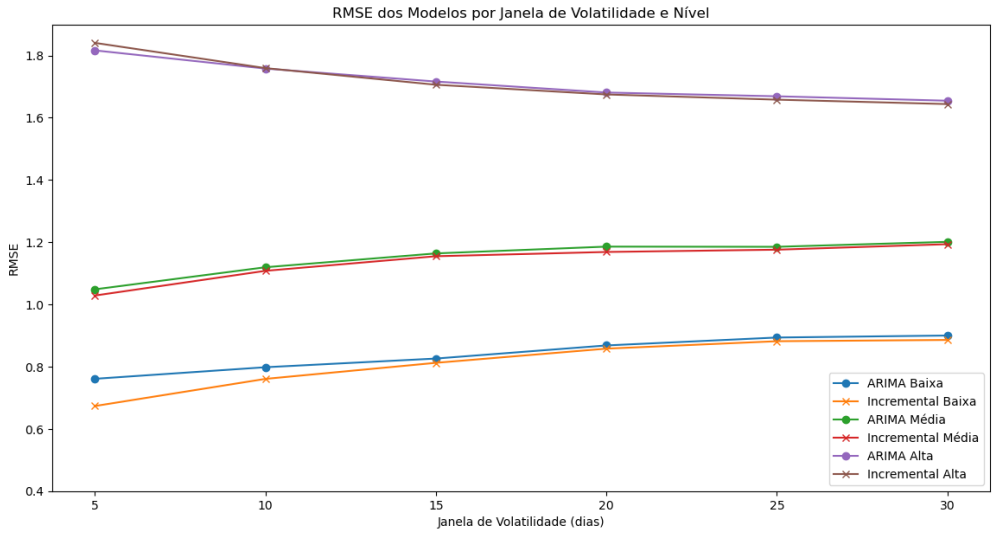


Os valores do RMSE são similares ao longo do período analisado, registrando 1,29 para o modelo ARIMA e 1,28 para o incremental, o que evidencia um desempenho quase idêntico entre os dois modelos. Uma análise mais detalhada revela que até o ano de 2000, o modelo incremental apresenta valores de RMSE ligeiramente superiores; contudo, essa tendência se inverte após 2000, com o modelo incremental mostrando um desempenho levemente melhor. Esse padrão sublinha a habilidade similar dos modelos de se ajustarem às mudanças no comportamento do preço do *Brent*, enfatizando a necessidade de adaptar as estratégias de modelagem para acompanhar as variações e tendências do mercado ao longo do tempo

A Figura 9 ilustra a RMSE para modelos ARIMA tradicional e incremental, analisados em função da janela de tempo e do nível de volatilidade do mercado. O eixo x desta figura representa a janela de volatilidade em dias (5, 10, 15, 20, 25, 30), enquanto o eixo y exibe o RMSE. Pode-se observar que:

- altos níveis de volatilidade estão associados a maiores valores de RMSE para ambos os modelos, indicando que previsões precisas se tornam mais desafiadoras sob essas condições e destacando a sensibilidade dos modelos a variações intensas no mercado.
- observa-se uma diminuição progressiva do RMSE à medida que a janela de volatilidade aumenta em cenários de alta volatilidade; entretanto, essa tendência não é consistentemente observada em ambientes de volatilidade baixa e média, apontando para uma resposta diferenciada dos modelos em relação às escalas de análise da volatilidade.
- o modelo ARIMA incremental exibe um RMSE ligeiramente inferior ao do modelo ARIMA tradicional em todas as janelas de volatilidade, especialmente em níveis baixos e médios de volatilidade, sugerindo que o modelo incremental tem uma vantagem marginal em manter a acurácia das previsões em condições menos voláteis.

Figura 9 – RMSE por Janela e Nível de Volatilidade.



6 Considerações Finais

Com base nos resultados obtidos no estudo de simulação, observou-se que tanto o *regret* médio quanto o mediano exibiram um crescimento linear, e não sublinear, em relação ao aumento de T. No entanto, o coeficiente angular deste crescimento é significativamente menor que $\alpha = 1$, o que demonstra a capacidade do modelo ARIMA incremental de oferecer previsões competitivas, controlando efetivamente o *regret* mesmo em cenários com diferentes tipos de *drifts*.

Na avaliação dos modelos nos dados reais, observou-se que ambos modelos apresentaram um desempenho praticamente idêntico. Quanto ao desempenho dos modelos em relação ao nível de volatilidade, observou-se os maiores valores de RMSE e MAE em períodos de alta volatilidade, ressaltando a sensibilidade dos modelos a flutuações intensas no mercado. Os modelos apresentaram uma resposta diferenciada às variações na escala da análise de volatilidade, sugerindo nuances na maneira como cada nível de volatilidade influencia a precisão dos mesmos. O modelo ARIMA incremental apresentou menores níveis de MAE e RMSE nos níveis de volatilidade baixa e média, sugerindo uma vantagem marginal do modelo incremental na manutenção da precisão das previsões em ambientes menos voláteis.

Em comparação direta, o modelo incremental estabelece-se como notavelmente mais eficiente, minimizando o tempo de ajuste em relação ao modelo ajustado pelo Auto ARIMA de maneira expressiva.

Os resultados deste trabalho corroboram a hipótese inicial, indicando que estratégias de modelagem mais diretas e menos complexas podem, de fato, oferecer desempenhos comparáveis a métodos que envolvem seleções automáticas e elaboradas de parâmetros, particularmente em ambientes dinâmicos.

Em trabalhos futuros, seria importante aprofundar a análise dos métodos de ajuste automático já existentes para modelos ARIMA incremental, explorando a adaptabilidade e o tempo de ajuste em diferentes conjuntos de dados e cenários de mercado. Adicionalmente, comparar de forma sistemática essas abordagens de ajuste automático com outras metodologias de modelagem de séries temporais poderia elucidar as condições sob as quais cada método se destaca, contribuindo assim para uma seleção de modelo mais informada em estudos futuros e aplicações práticas. Essa linha de investigação não apenas enriqueceria o entendimento das capacidades e limitações dos modelos ARIMA incremental, mas também destacaria potenciais áreas para inovação e otimização dentro do campo da análise de séries temporais.

REFERÊNCIAS

ANAVA, O. et al. Online learning for time series prediction. *Journal of Machine Learning Research*, Microtome Publishing, v. 30, p. 172–184, 2013. ISSN 1532-4435. 26th Conference on Learning Theory, COLT 2013 ; Conference date: 12-06-2013 Through 14-06-2013. 7, 16, 17

BOTTOU, L. Online algorithms and stochastic approximations. *Online Learning and Neural Networks*, Cambridge University Press, Cambridge, UK, 1998. 7

BOX, G. et al. *Time Series Analysis: Forecasting and Control*. [S.l.]: Wiley, 2015. (Wiley Series in Probability and Statistics). 7, 16, 17, 18

BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. New York, NY: Springer, 2016. 7, 17, 18

CESA-BIANCHI, N.; LUGOSI, G. *Prediction, Learning, and Games*. [S.l.]: Cambridge University Press, 2006. ISBN 9780521841085. 10, 14

DITZLER, G. et al. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, IEEE, v. 10, n. 4, p. 12–25, 2015. 9

GAMA, J. et al. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 46, n. 4, p. 1–37, 2014. 9

GUERRIER, S. et al. *simts: Time Series Analysis Tools*. [S.l.], 2023. R package version 0.2.2. Disponível em: <https://CRAN.R-project.org/package=simts>. 20

HAMEED ASMAA NORI, A. Z. A. A critical analysis of the global oil fluctuation rate and its impact on opec countries from 2010 to 2021. *Journal of Humanities and Social Sciences Research*, v. 2, n. 3, Jul. 2023. Disponível em: <https://jhssrjournal.com/index.php/journal/article/view/223>. 28

HAYKIN, S. *Adaptive Filter Theory*. [S.l.]: Pearson, 2014. 11, 13

HULL, J. C. *Options, Futures, and Other Derivatives*. 10. ed. [S.l.]: Pearson, 2018. ISBN 978-0-13-447208-9. 23

HYNDMAN, R. et al. *forecast: Forecasting Functions for Time Series and Linear Models*. [S.l.], 2024. R package version 8.22.0. Disponível em: <https://CRAN.R-project.org/package=forecast>. 19, 20

LI, Y.; WU, K.; LIU, J. Self-paced arima for robust time series prediction. *Knowledge-Based Systems*, v. 269, p. 110489, 2023. ISSN 0950-7051. 7, 16

LIU, C. et al. Online arima algorithms for time series prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 30, n. 1, Feb. 2016. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/10257>. 7, 9, 16, 18

MONTIEL, J. et al. River: machine learning for streaming data in python. 2021. 19

MORETTIN, P.; TOLOI, C. *Análise de séries temporais: modelos lineares univariados*. [S.l.]: BLUCHER., 2018. ISBN 9788521213529. 7, 16, 17, 18

RAAB, C.; HEUSINGER, M.; SCHLEIF, F. Reactive soft prototype computing for concept drift streams. *CoRR*, abs/2007.05432, 2020. Disponível em: <https://arxiv.org/abs/2007.05432>. 9

ROBBINS, H.; MONRO, S. A stochastic approximation method. *The annals of mathematical statistics*, JSTOR, p. 400–407, 1951. 10, 11

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. 11

ROSSUM, G. V.; JR, F. L. D. *Python tutorial*. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995. 19

SHALEV-SHWARTZ, S. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, v. 4, n. 2, p. 107–194, 2012. 13, 14

SHAO, W. et al. Adaptive online learning for the autoregressive integrated moving average models. *Mathematics*, v. 9, n. 13, p. 1523, 2021. Disponível em: <https://doi.org/10.3390/math9131523>. 7, 10

SUN, C. et al. The role of china's crude oil futures in world oil futures market and china's financial market. *Energy Economics*, v. 120, p. 106619, 2023. ISSN 0140-9883. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0140988323001172>. 28

WICKHAM, H.; RSTUDIO. *tidyverse: Easily Install and Load the 'Tidyverse'*. [S.l.], 2023. R package version 2.0.0. Disponível em: <https://CRAN.R-project.org/package=tidyverse>. 19, 20

WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: *IRE WESCON convention record*. [S.l.: s.n.], 1960. v. 4, p. 96–104. 11, 12

ZAVADSKA, M.; MORALES, L.; COUGHLAN, J. Brent crude oil prices volatility during major crises. *Finance Research Letters*, v. 32, p. 101078, 2020. ISSN 1544-6123. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1544612318304380>. 27, 28

ZHANG, X.; YU, L.; WANG, S. The impact of financial crisis of 2007-2008 on crude oil price. In: ALLEN, G. et al. (Ed.). *Computational Science – ICCS 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 643–652. 27

ZHANG, Y.-J.; ZHANG, L. Interpreting the crude oil price movements: Evidence from the markov regime switching model. *Applied Energy*, v. 143, p. 96–109, 2015. ISSN 0306-2619. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306261915000112>. 28

ŽLIOBAITė, I. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*, 2010. 9