Análise Comparativa de Métodos Estatísticos e de Aprendizado de Máquina em Receitas Orçamentárias do Rio Grande do Sul

Autor: Aray Gustavo Furtado Feldens

Orientador: Dr. Rodrigo Righi

RESUMO

O avanço das tecnologias de aprendizado de máquina (ML), oferecem novas perspectivas para a previsão de séries temporais orçamentárias. Está é essencial para a administração eficiente de recursos financeiros por organizações e entidades governamentais. Diante da rápida evolução das tecnologias em Inteligência Artificial (IA) e ML, e da falta de um modelo padrão universalmente aceito, surge o desafio de escolher modelos adequados para análises financeiras. O objetivo deste trabalho é desenvolver uma metodologia para selecionar e aplicar modelos preditivos precisos em séries temporais orçamentárias, utilizando dados de receitas do Rio Grande do Sul.

A metodologia adotada consiste na aplicação e comparação de diversos modelos estatísticos e de ML (ARIMA, SARIMA, Random Forest e XGBoost). Esta abordagem permite avaliar a eficácia de cada modelo em função das características específicas das receitas orçamentárias.

Como resultados foi observado que modelos de ML são mais eficazes em categorias com padrões complexos e não lineares, enquanto modelos estatísticos são preferíveis para séries com padrões claros de sazonalidade e estacionariedade. Portanto, os achados deste estudo oferecem contribuições significativas para o campo das finanças públicas. Eles fornecem uma abordagem prática e validada para a previsão de receitas orçamentárias, facilitando a tomada de decisões fiscais informadas e a alocação precisa de recursos. Além disso, destacam a importância de uma seleção cuidadosa de modelos, adaptando-os às particularidades de cada série temporal, para responder de maneira ágil às mudanças econômicas.

PALAVRAS-CHAVE: Previsão de séries temporais; Aprendizado de máquina; Finanças públicas; Séries Temporais Orçamentárias.

SUMÁRIO

1. INTRODUÇÃO

A previsão de séries temporais orçamentárias representa uma tarefa crucial para organizações e entidades governamentais que buscam uma administração eficiente de recursos financeiros. Com o avanço contínuo das tecnologias de aprendizado de máquina (ML), emergiu uma capacidade aprimorada para realizar previsões mais acuradas e confiáveis em séries temporais financeiras. Este estudo tem o propósito de desenvolver uma metodologia para a seleção e aplicação de modelos preditivos específicos a séries temporais orçamentárias, financeiras e fiscais.

A essência da análise de séries temporais reside na investigação da causalidade entre eventos passados e futuros, abrangendo uma ampla gama de disciplinas, incluindo medicina, climatologia, economia e astronomia. Com a disponibilidade crescente de grandes volumes de dados e o avanço tecnológico no processamento computacional, uma vasta diversidade de modelos e ferramentas para a análise de séries temporais foi desenvolvida. No entanto, a ausência de um modelo universalmente superior e a rápida evolução das tecnologias em Inteligência Artificial (IA), ML, redes neurais, deep learning e autoML tornam a escolha do método mais apropriado um desafio considerável. De acordo com Zhang et al. (2022), o campo do ML está experimentando um crescimento exponencial, o que ressalta a necessidade de uma metodologia organizada e unificada que possa efetivamente selecionar e aplicar tanto as técnicas recentes quanto as tradicionais.

O objetivo principal deste estudo é testar os modelos tradicionais estatísticos e de ML para previsões financeiras usando uma base de dados histórica de receitas orçamentárias do Estado do Rio Grande do Sul. Os objetivos secundários são identificar as diferenças de performance destes modelos em um caso real e prático, bem como avançar na metodologia e recomendações para um conjunto de práticas.

Este estudo se justifica pela necessidade crescente de ferramentas de previsão mais robustas e confiáveis no campo das finanças públicas, onde decisões baseadas em dados precisos e atualizados são cruciais. Portanto, este trabalho pode contribuir para o avanço na aplicação de ML em finanças e oferece uma abordagem prática e validada para profissionais e decisores do setor financeiro.

2. REFERENCIAL TEÓRICO

2.1 Séries temporais orçamentárias

Séries temporais orçamentárias constituem um pilar fundamental da análise financeira, desempenhando um papel crítico na tomada de decisões estratégicas e no planejamento de longo prazo. Estas séries são compostas por dados financeiros históricos, coletados de forma sequencial ao longo do tempo. De acordo com Box e Jenkins (2015), a previsão de séries temporais é um problema central em diversas áreas da ciência e da engenharia, sendo os modelos autoregressivos frequentemente empregados para essa finalidade devido à sua eficácia e flexibilidade.

No âmbito do setor público, as séries temporais orçamentárias e financeiras são vitais para o planejamento e a gestão eficaz das receitas e despesas governamentais. A análise dessas séries é fundamental para a formulação de políticas fiscais e monetárias eficientes (AUERBACH & GORODNICHENKO, 2012). Governos recorrem a essa análise para avaliar o impacto de políticas implementadas, possibilitando ajustes necessários para assegurar a estabilidade econômica e promover o bem-estar social (Romer, 2019).

No setor privado, as séries temporais financeiras são essenciais para uma gestão de recursos, investimentos e riscos mais efetiva. Empresas e investidores se baseiam na análise de séries temporais financeiras para tomar decisões fundamentadas sobre investimentos, alocação de recursos e gestão de riscos (CAMPBELL, LO, & MACKINLAY, 1997). A compreensão das tendências de mercado e a habilidade de prever eventos futuros são essenciais para adaptar-se às mudanças nas condições econômicas e maximizar a rentabilidade (TSAY, 2005).

2.2 Aprendizado de máquina

O aprendizado de máquina (ML), uma subárea vital da inteligência artificial, foca no desenvolvimento de algoritmos e modelos capazes de permitir que um sistema aprenda e se adapte a partir de dados. Esta área tem experimentado uma adoção crescente na previsão de séries temporais financeiras, revolucionando as abordagens tradicionais com a capacidade de processar e analisar grandes volumes de informações complexas.

Conforme Géron (2017), as redes neurais representam uma das técnicas de aprendizado de máquina mais potentes para previsão de séries temporais, oferecendo uma flexibilidade significativa e a capacidade de capturar padrões não-lineares nos dados. Além das redes neurais, outros métodos de ML, como máquinas de vetor de suporte (SVMs) e algoritmos de florestas aleatórias, também têm

demonstrado eficácia na previsão de séries temporais, cada um com suas particularidades e vantagens (James et al., 2013).

A aplicação do ML em séries temporais financeiras não se limita apenas à previsão. Técnicas como aprendizado profundo (Deep Learning) e redes neurais recorrentes (RNNs) estão sendo exploradas para modelar a complexidade e a dinâmica temporal desses dados (GOODFELLOW et al., 2016). Essas abordagens permitem não apenas prever tendências futuras, mas também identificar relações causais e padrões ocultos que são cruciais para a análise financeira.

Contudo, a implementação de ML em séries temporais financeiras apresenta desafios únicos, especialmente no que se refere à natureza não estacionária dos dados financeiros e à necessidade de processamento de grandes conjuntos de dados. A escolha do modelo adequado e a otimização de hiperparâmetros são cruciais para o sucesso dessas aplicações (BISHOP, 2006).

2.3 Abordagens Existentes em Aprendizado de Máquina para Previsão de Séries Temporais Orçamentárias e Financeiras

As técnicas de aprendizado de máquina oferecem um amplo espectro de abordagens para a previsão de séries temporais em contextos orçamentários e financeiros. A escolha eficaz dessas técnicas é crucial e deve ser baseada em uma compreensão profunda de fatores como a natureza dos dados, o horizonte de previsão desejado e os objetivos específicos da análise, conforme destacado por Hyndman e Athanasopoulos (2018).

Métodos Estatísticos Tradicionais:

- Média Móvel (MA): Utilizada para suavizar flutuações de curto prazo e identificar tendências de longo prazo (BOX et al., 2015).
- AutoRegressão (AR): Modela como cada valor da série temporal é uma função linear de seus valores anteriores (BOX et al., 2015).
- ARIMA: Combina AR e MA, adequado para dados não estacionários (BOX et al., 2015).
- Sazonalidade, Tendência e Resíduos (STL): Decompõe séries temporais (CLEVELAND et al., 1990).
- Modelos de Espaço de Estados: Abordam a relação entre observações e estados latentes em séries temporais (DURBIN, KOOPMAN, 2012).

Métodos Modernos de Aprendizado de Máquina:

- Redes Neurais Recorrentes (RNNs): Eficientes para modelar dados sequenciais (GOODFELLOW, et al., 2016).
- Long Short-Term Memory (LSTM): Especialmente projetada para evitar o problema do desaparecimento do gradiente em RNNs (HOCHREITER, SCHMIDHUBER, 1997).
- Redes Neurais Convolucionais (CNNs) para Séries Temporais: Adaptadas para a análise temporal (GOODFELLOW et al., 2016).
- Máquinas de Vetores de Suporte (SVMs) para Regressão: Utilizadas em previsões de séries temporais (DRUCKER et al., 1997).
- Florestas Aleatórias para Séries Temporais: Empregam múltiplas árvores de decisão (BREIMAN, 2001).
- Métodos de Boosting: Como o Gradient Boosting, para melhorar a precisão das previsões (FREUND, SCHAPIRE, 1997).

Os métodos modernos de aprendizado de máquina, especialmente aqueles baseados em redes neurais profundas e algoritmos de ensemble como Florestas Aleatórias e Boosting, têm demonstrado alta eficácia em previsões complexas, lidando bem com grandes volumes de dados e padrões não-lineares. No entanto, exigem maior poder computacional e podem ser menos interpretáveis. Os métodos estatísticos tradicionais, apesar de mais simples, são valiosos pela sua eficiência em contextos com dados limitados e pela facilidade de interpretação, sendo ainda amplamente utilizados em diversas aplicações práticas.

2.4 Avanços na previsão de séries temporais

No âmbito da previsão de séries temporais, observamos avanços significativos e inovadores, conforme ilustrado em estudos recentes. Smyl (2020) propõe uma metodologia híbrida que integra suavização exponencial e redes neurais recorrentes. Esta abordagem representa uma evolução notável frente aos métodos tradicionais, destacando-se na habilidade de capturar padrões complexos e não lineares nas séries temporais, uma capacidade crítica especialmente em cenários de previsão de demanda e análise de tendências de mercado.

Em um avanço paralelo, Flunkert, Salinas e Gasthaus (2020) apresentam o modelo DeepAR, que introduz a previsão probabilística utilizando redes recorrentes autoregressivas. Distinto dos

métodos convencionais que se limitam a previsões pontuais, o DeepAR é capaz de gerar distribuições de previsão, oferecendo uma compreensão mais profunda da incerteza associada às previsões. Tal abordagem é de suma importância em aplicações financeiras e em contextos em que a gestão de riscos é crucial. Além disso, Rangapuram et al. (2018) exploram o potencial dos modelos de espaço de estado profundo para a previsão de séries temporais. Este estudo ressalta a eficácia dos modelos de espaço de estado, enriquecidos com técnicas de *deep learning*, na modelagem de dependências temporais complexas, especialmente em grandes volumes de dados. Esta metodologia é aplicável em uma variedade de campos, desde previsões meteorológicas até análises financeiras e planejamento de inventário.

Deste modo, esses estudos coletivamente representam avanços notáveis na previsão de séries temporais, contribuindo com métodos e insights únicos. A crescente integração de técnicas de *deep learning* e modelagem probabilística reflete a necessidade de abordagens mais sofisticadas e precisas de previsão, dada a complexidade e o volume crescente de dados em diversas áreas. Estas inovações são particularmente valiosas em situações em que a precisão das previsões e a compreensão das incertezas associadas são fundamentais para a tomada de decisões informadas. No entanto, esse estudo se limitará ao uso de métodos estatísticos mais tradicionais e de aprendizagem de máquina puro, para efeitos de comparação e estabelecimento de benchmarks iniciais, abrindo caminho para estudos futuros com métodos híbridos e de *deep learning* que exigem maior poder computacional e elaboração dos parâmetros e hiperparametros.

3. MÉTODO DE PESQUISA

3.1 Descrição e Coleta dos Dados

A base de dados utilizada para a previsão de receitas orçamentárias do Estado do Rio Grande do Sul é composta por um conjunto abrangente e detalhado de informações financeiras. Esses dados são essenciais para entender o fluxo de receitas ao longo do tempo e identificar padrões que podem ajudar na tomada de decisões governamentais.

O dataset compila os valores mensais de diversas fontes de receita do estado, cobrindo o período de janeiro de 2008 a outubro de 2023. Embora alguns dados sejam registrados diariamente, outros são contabilizados apenas no fechamento do mês, razão pela qual optou-se por utilizar uma base de dados mensal para a análise.

As categorias de receita incluídas são:

- ICMS (Imposto sobre Circulação de Mercadorias e Serviços): Este é o principal imposto estadual, representando uma parte significativa da receita. Os dados refletem a arrecadação mensal deste imposto.

- IPVA (Imposto sobre a Propriedade de Veículos Automotores): Reflete a arrecadação mensal do imposto sobre veículos.

- IRRF (Imposto de Renda Retido na Fonte): Registra o montante recolhido mensalmente como parte do imposto de renda retido na fonte.

- ITCD e ITBI (Imposto sobre Transmissão Causa Mortis e Doação e Imposto sobre Transmissão de Bens Imóveis): Estes dados representam as receitas provenientes de transmissões patrimoniais.

- Taxas: Inclui diversas taxas estaduais, refletindo a arrecadação mensal.

- Contribuições Previdenciárias: Montantes recolhidos referentes às contribuições para o sistema previdenciário estadual.

- Outras Contribuições: Abrange outras fontes de contribuições diversas.

- Receita Financeira: Registra os rendimentos de aplicações e outras receitas financeiras do estado.

- Receitas Patrimoniais e Agroindustriais/Serviços: Inclui receitas provenientes de patrimônios e do setor agroindustrial e de serviços.

Foram excluídas da análise os JSCP (Juros sobre Capital Próprio) e os Dividendos, considerando a sua natureza não recorrente e a menor relevância que apresentam para as projeções futuras. Além disso, optou-se por não incluir na avaliação as receitas provenientes de Transferências, Capital e Outras Receitas, dado que estas categorias tendem a apresentar maior estabilidade e, portanto, impactam menos as variáveis que este estudo está focando.

3.2 Pré-processamento dos Dados

3.2.1 Estruturação dos Dados:

Dada a natureza das séries temporais, os dados foram estruturados para refletir a dependência temporal entre as observações, diferenciando-os dos problemas típicos de regressão onde as observações são consideradas independentes.

3.2.2 Transformações e Verificações de Estacionariedade:

A estacionariedade foi avaliada utilizando o teste Dickey-Fuller Aumentado. As séries que não eram estacionárias foram submetidas a transformações, como diferenciação, para estabilizar a média e a variância ao longo do tempo.

3.3 Modelagem e Previsão

3.3.1 ARIMA:

O modelo ARIMA foi empregado por sua robustez e eficácia em modelar tendências e padrões em séries temporais.

3.3.2 SARIMA:

O modelo SARIMA, uma extensão do ARIMA, foi utilizado para capturar tanto a sazonalidade quanto as tendências não sazonais nos dados. Parâmetros sazonais foram cuidadosamente selecionados com base na periodicidade observada nas séries temporais.

3.3.3 Random Forest

A abordagem Random Forest foi aplicada para previsão das receitas. Essa técnica de ensemble utiliza múltiplas árvores de decisão para produzir uma previsão mais estável e robusta, reduzindo o risco de sobreajuste.

3.3.4 XGBoost

O modelo XGBoost, conhecido por sua eficiência e desempenho, foi utilizado devido à sua capacidade de lidar com diferentes tipos de características das séries temporais, incluindo tendências não lineares e interações complexas entre variáveis.

3.4 Seleção e otimização de hiperparâmetros

3.4.1 Hiperparâmetros ARIMA e SARIMA

Ao selecionar hiperparâmetros para modelos ARIMA e SARIMA, duas metodologias principais são frequentemente empregadas: a análise de Função de Autocorrelação (ACF) e Função de Autocorrelação Parcial (PACF), e o Critério de Informação de Akaike (AIC).

A ACF, representada por $\rho(k)=Var(Yt)Cov(Yt,Yt-k)$, e a PACF, denotada por $\phi(k)=Corr(Yt-Yt(k-1),Yt-k-Yt-k(k-1))$, são usadas para identificar visualmente as ordens autorregressivas (AR) e de médias móveis (MA) em modelos ARIMA(p, d, q). Essas funções fornecem uma perspectiva visual da correlação entre observações em lags sucessivos e a correlação de um lag específico isolando os efeitos dos lags intermediários.

Por outro lado, o AIC, calculado por $AIC=2k-2\ln(L)$, onde k representa o número de parâmetros estimados e L a máxima verossimilhança do modelo, oferece uma abordagem quantitativa para seleção de modelos, equilibrando a complexidade do modelo com o ajuste aos dados. Modelos com valores menores de AIC são preferidos por serem mais parcimoniosos e eficazes. Na prática, a combinação de ACF/PACF e AIC é recomendada, começando com ACF e PACF para estimativas iniciais e refinando com AIC para garantir a seleção de um modelo ótimo. Esta abordagem híbrida

une a interpretação visual intuitiva de ACF e PACF com a objetividade e rigor do AIC, fornecendo um equilíbrio na construção de modelos de séries temporais precisos e confiáveis.

Optamos pelo AIC, pois essa abordagem reduz a subjetividade inerente às análises visuais e simplifica o processo de seleção, minimizando o risco de sobre ajuste e assegurando a escolha de um modelo mais eficaz e adequado aos dados em análise.

3.4.2 Hiperparâmetros Random Forest

Na investigação das previsões de receita orçamentária utilizando o modelo Random Forest, adotou-se uma abordagem sistemática para a seleção de hiperparâmetros. Foi estabelecida uma grade de parâmetros, compreendendo variações no número de árvores (n_estimators), com valores de 100, 200 e 500; profundidade máxima das árvores (max_depth), considerando 10, 20 e a ausência de limite; número mínimo de amostras para dividir um nó (min_samples_split), fixado em 2, 5 e 10; e o número mínimo de amostras em um nó folha (min_samples_leaf), com opções de 1, 2 e 4. Esta grade de hiperparâmetros foi utilizada em conjunto com o método GridSearchCV, empregando validação cruzada com três divisões e um *scorer* baseado no erro quadrático médio raiz (RMSE).

3.4.2 Hiperparâmetros XGBoost

A otimização dos hiperparâmetros para o modelo XGBoost foi conduzida através de um procedimento de busca em grade (assim como o usado no Random Forest), utilizando o GridSearchCV. A grade de hiperparâmetros definida incluiu várias iterações para os seguintes parâmetros: número de estimadores (n_estimators), com valores configurados como [100,500,1000] taxa de aprendizado (learning_rate), estabelecida em [0.01,0.1,0.2] profundidade máxima das árvores (max_depth), com opções de [3,5,7]; e a fração de subsample (subsample), definida em [0.7,0.8,0.9], visaando identificar a combinação de parâmetros que minimiza o erro quadrático médio (MSE), uma métrica comum na avaliação de modelos de regressão. A busca em grade foi configurada para realizar validação cruzada com três divisões (cv=3), permitindo uma avaliação robusta e confiável dos modelos candidatos. Este procedimento sistemático e rigoroso assegurou a seleção dos hiperparâmetros mais adequados para o modelo XGBoost.

3.5 Validação e Avaliação de Modelos

A validação dos modelos no estudo foi realizada empregando uma abordagem meticulosa, que incluiu a validação cruzada com divisão temporal, considerando a natureza sequencial dos dados de séries temporais. A função foi desenhada para computar as métricas de erro raiz quadrática média (RMSE), erro absoluto médio (MAE) e erro percentual absoluto médio (MAPE) para cada modelo.

Para a validação cruzada temporal, o conjunto de dados foi dividido em cinco partes sequenciais usando TimeSeriesSplit (n_splits=5). Em cada iteração, uma parte diferente do conjunto de dados foi usada como teste, enquanto as partes anteriores foram utilizadas para treinamento. Este método é essencial para séries temporais, assegurando que a ordem temporal dos dados seja respeitada e evitando o vazamento de informações.

Para modelos baseados em séries temporais, como ARIMA e SARIMA, o ajuste foi realizado diretamente nos valores de série temporal (y), e as previsões foram geradas para o mesmo número de passos à frente quanto o tamanho do conjunto de teste. Para modelos como Random Forest e XGBoost, o ajuste e a previsão foram feitos usando tanto as variáveis explicativas (X) quanto a variável alvo (y).

As métricas de avaliação - RMSE, MAE e MAPE - foram calculadas para cada conjunto de teste. Os resultados para cada modelo foram a média dessas métricas ao longo das cinco divisões da validação cruzada temporal. Este processo de validação forneceu uma avaliação robusta e confiável do desempenho de cada modelo, respeitando a natureza temporal dos dados e permitindo comparações justas entre diferentes abordagens de modelagem.

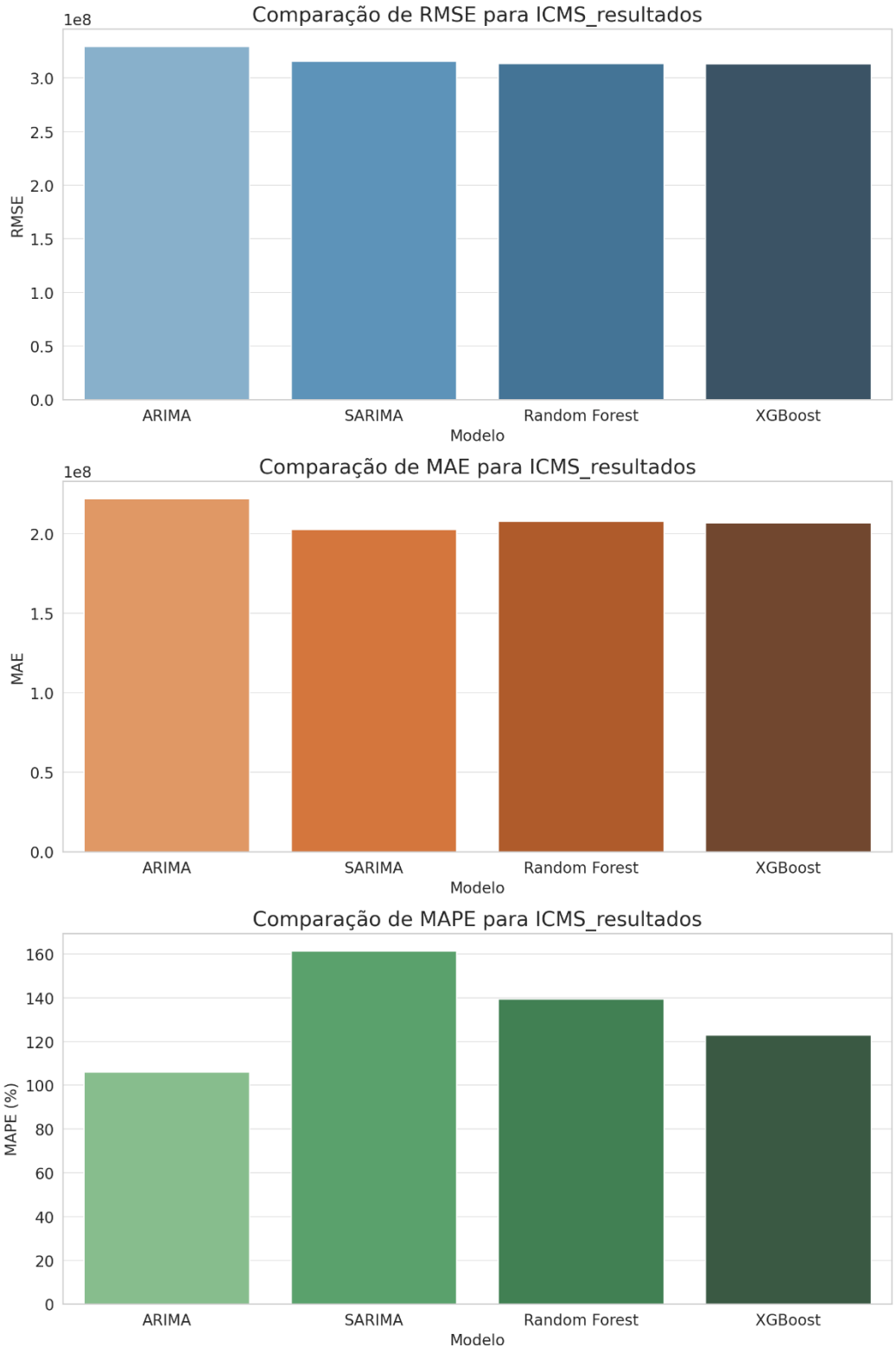
4. RESULTADOS E DISCUSSÃO

A análise dos resultados obtidos para as diversas categorias de receita do Estado do Rio Grande do Sul revelou insights significativos sobre o desempenho dos modelos ARIMA, SARIMA, Random Forest e XGBoost. Cada categoria de receita apresentou características únicas, influenciando a eficácia dos modelos aplicados.

- **ICMS:** O modelo XGBoost destacou-se em termos de RMSE e MAE, demonstrando alta precisão nas previsões. O SARIMA, embora ligeiramente menos preciso, também apresentou resultados competitivos, sugerindo sua adequação para capturar tendências e padrões sazonais nesta categoria.

Modelo	RMSE	MAE	MAPE
ARIMA	3.30e+08	2.22e+08	106.04
SARIMA	3.16e+08	2.03e+08	161.41
Random Forest	3.14e+08	2.08e+08	139.53
XGBoost	3.13e+08	2.07e+08	123.02

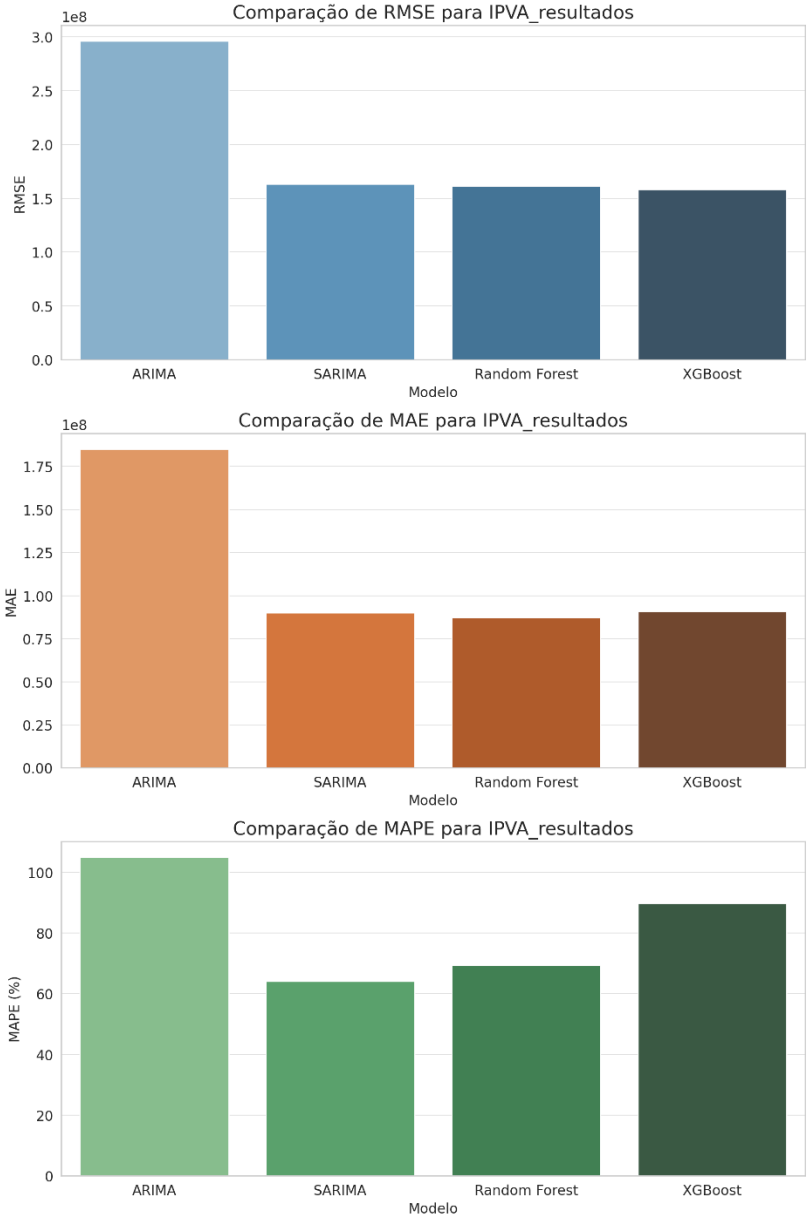
Figura 1. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para ICMS.



- **IPVA:** Nesta categoria, o XGBoost foi superior em RMSE, indicando maior precisão na previsão de grandes variações. Por outro lado, o SARIMA teve um desempenho melhor em MAE, sugerindo uma maior precisão em variações menores e mais frequentes.

Modelo	RMSE	MAE	MAPE
ARIMA	2.96e+08	1.85e+08	104.98
SARIMA	1.63e+08	9.00e+07	64.09
Random Forest	1.62e+08	8.73e+07	69.41
XGBoost	1.58e+08	9.08e+07	89.79

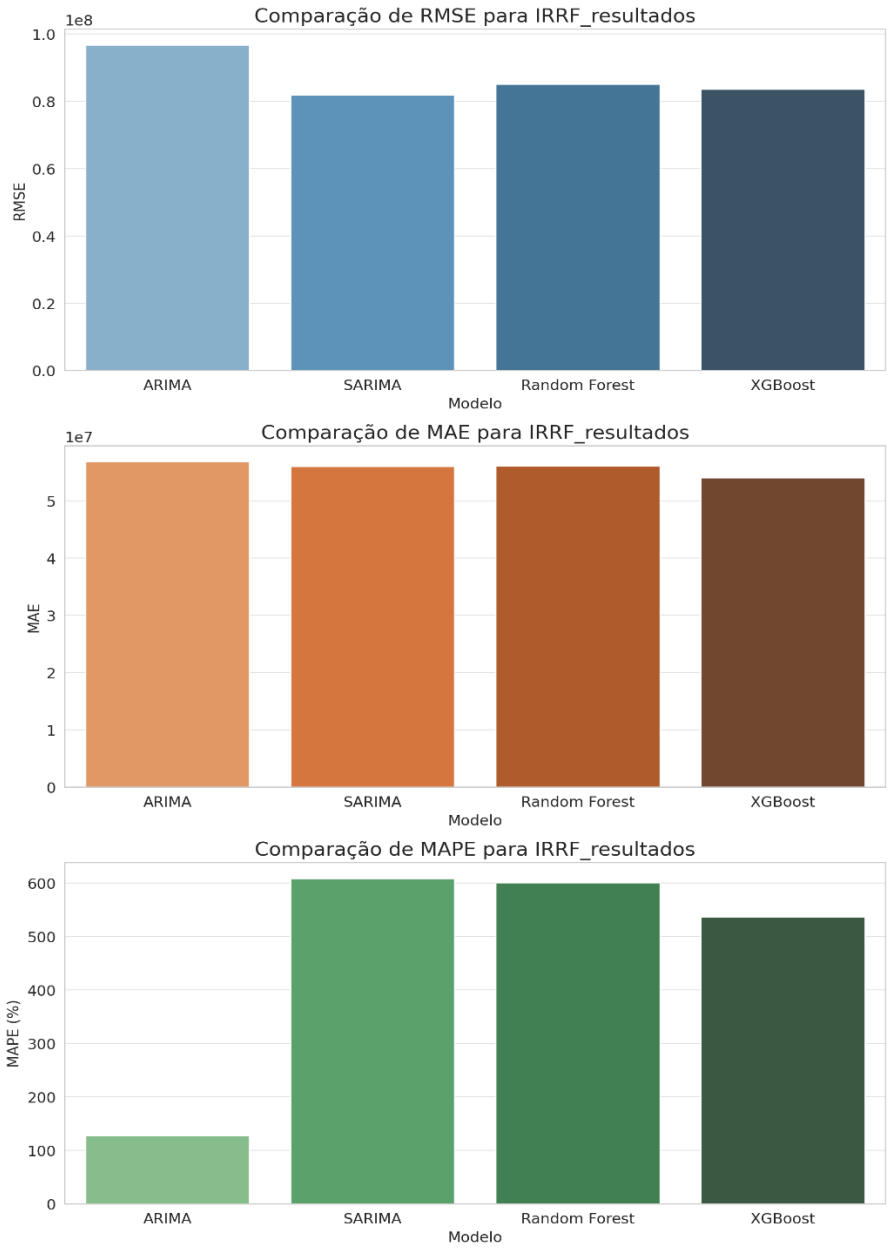
Figura 2 . Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para IPVA.



- **IRRF:** O modelo SARIMA sobressaiu-se com o menor RMSE e MAE, indicando um ajuste superior aos dados em comparação com os outros modelos. Isso pode ser atribuído à sua capacidade de capturar precisamente a sazonalidade e tendências lineares presentes nesta categoria.

Modelo	RMSE	MAE	MAPE
ARIMA	9.67e+07	5.68e+07	127.96
SARIMA	8.18e+07	5.60e+07	608.71
Random Forest	8.51e+07	5.61e+07	600.75
XGBoost	8.36e+07	5.40e+07	536.99

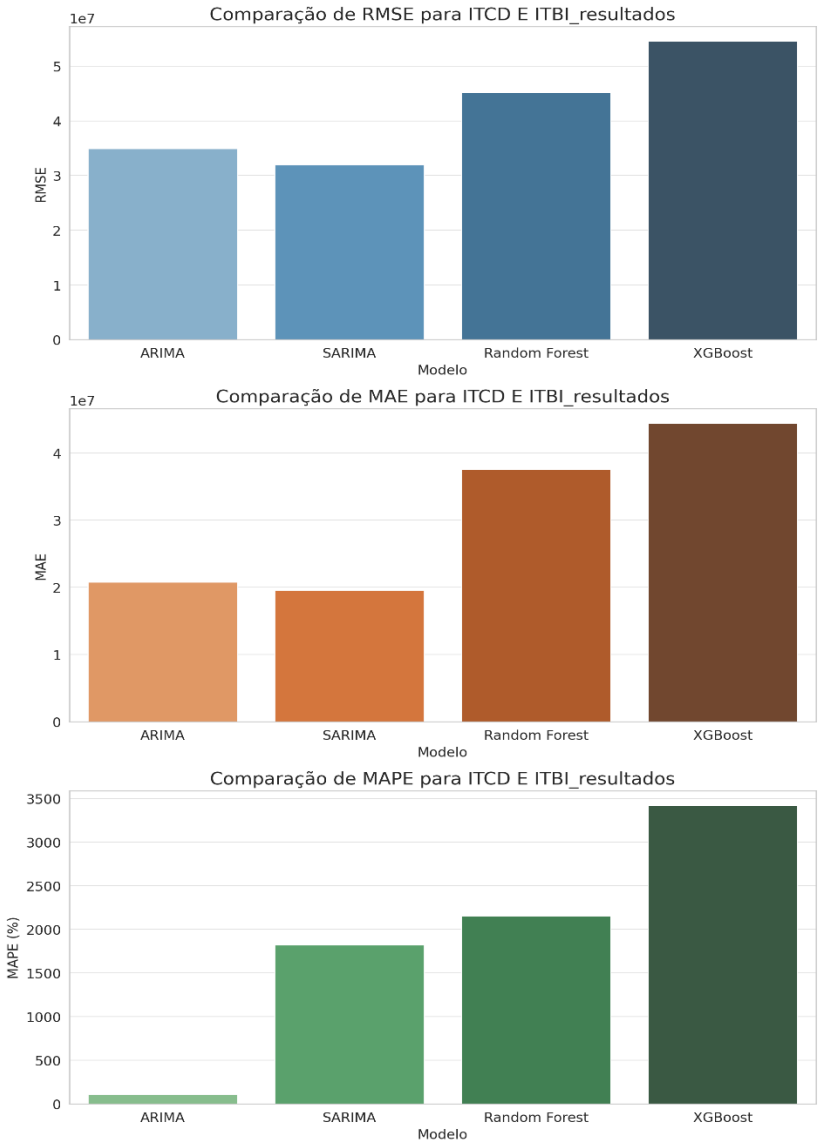
Figura 3. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para IRRF.



- **ITCD e ITBI:** Os modelos tradicionais, ARIMA e SARIMA, tiveram um desempenho superior em termos de RMSE e MAE comparado aos modelos de machine learning. O SARIMA, em particular, mostrou o menor RMSE, evidenciando sua eficácia em séries temporais com padrões mais estáveis e previsíveis.

Modelo	RMSE	MAE	MAPE
ARIMA	3.50e+07	2.08e+07	112.03
SARIMA	3.20e+07	1.96e+07	1823.37
Random Forest	4.52e+07	3.76e+07	2155.01
XGBoost	5.46e+07	4.44e+07	3418.30

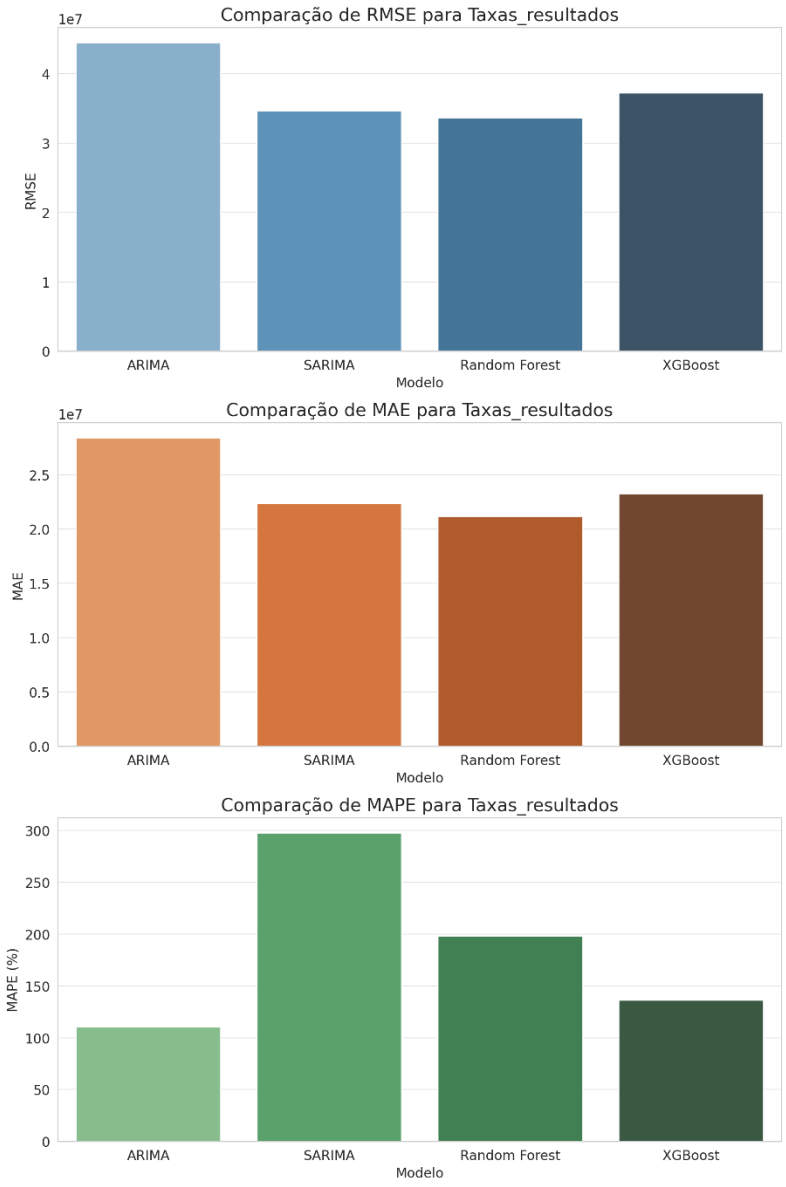
Figura 4. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para ITCD e ITBI.



- **Taxas:** O modelo Random Forest apresentou o menor RMSE, indicando um ajuste mais preciso aos dados. O SARIMA, com o menor MAE, demonstrou ser mais eficaz em prever variações menores.

Modelo	RMSE	MAE	MAPE
ARIMA	4.45e+07	2.84e+07	110.79
SARIMA	3.47e+07	2.24e+07	297.53
Random Forest	3.36e+07	2.12e+07	198.25
XGBoost	3.72e+07	2.33e+07	136.20

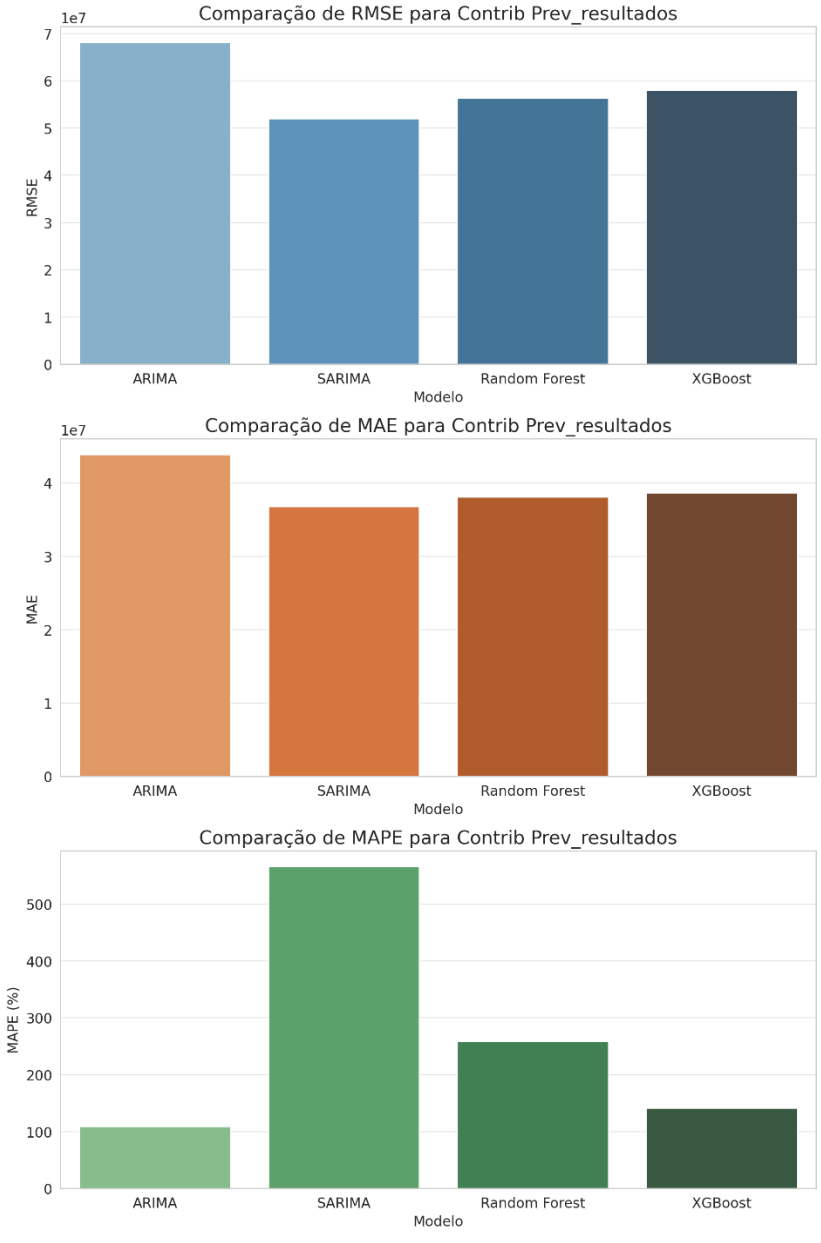
Figura 5. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para Taxas.



- **Contribuição Previdenciária:** O SARIMA mostrou-se mais preciso tanto em RMSE quanto em MAE, superando os modelos de machine learning. Isso pode refletir a presença de padrões sazonais e tendências mais lineares nesta categoria.

Modelo	RMSE	MAE	MAPE
ARIMA	6.81e+07	4.39e+07	109.31
SARIMA	5.20e+07	3.68e+07	565.80
Random Forest	5.63e+07	3.81e+07	258.78
XGBoost	5.80e+07	3.87e+07	141.23

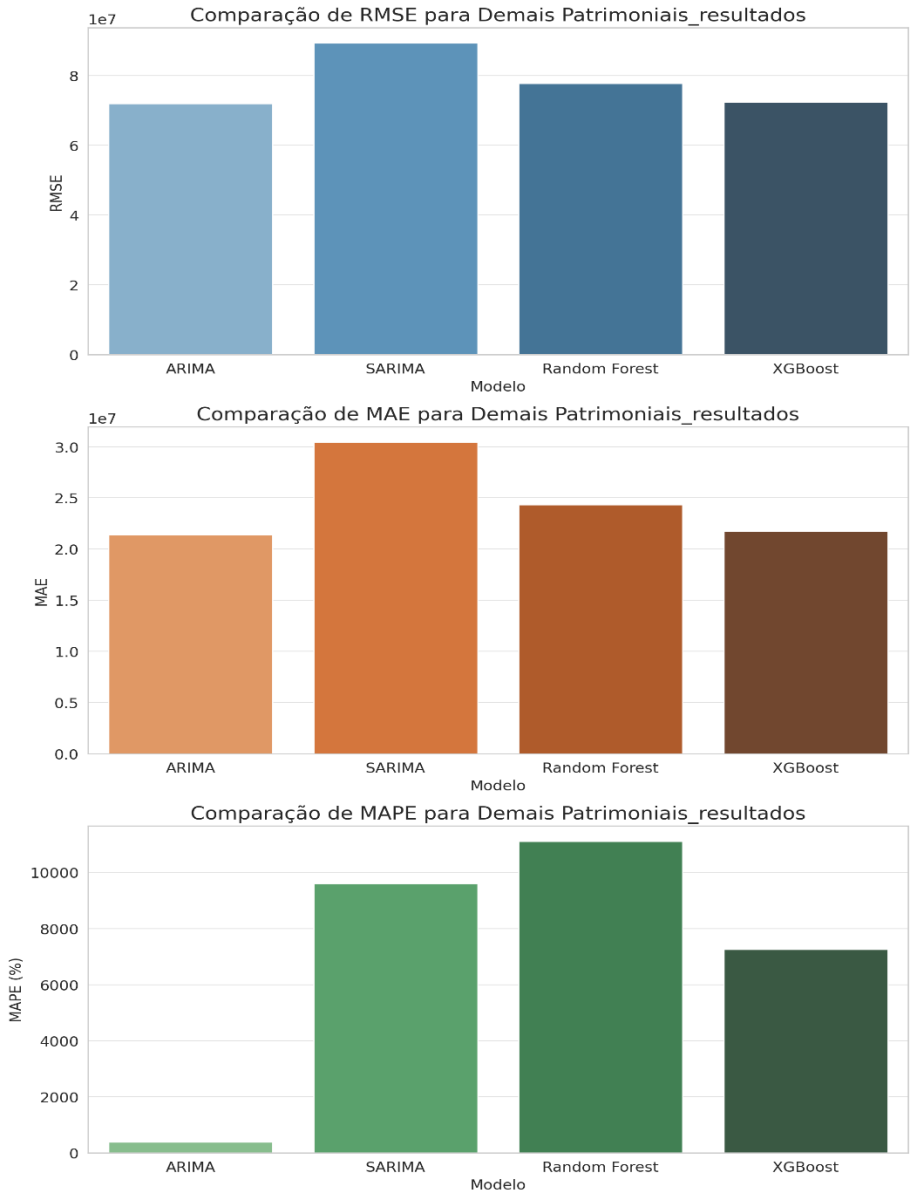
Figura 6. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para contribuição previdenciária.



- **Demais Receitas Patrimoniais:** XGBoost e ARIMA tiveram desempenhos semelhantes em termos de RMSE, mas o XGBoost apresentou um MAE ligeiramente superior, indicando uma precisão um pouco menor em previsões de curto prazo.

Modelo	RMSE	MAE	MAPE
ARIMA	7.21e+07	2.14e+07	400.60
SARIMA	8.94e+07	3.04e+07	9600.04
Random Forest	7.79e+07	2.43e+07	11107.62
XGBoost	7.24e+07	2.17e+07	7266.30

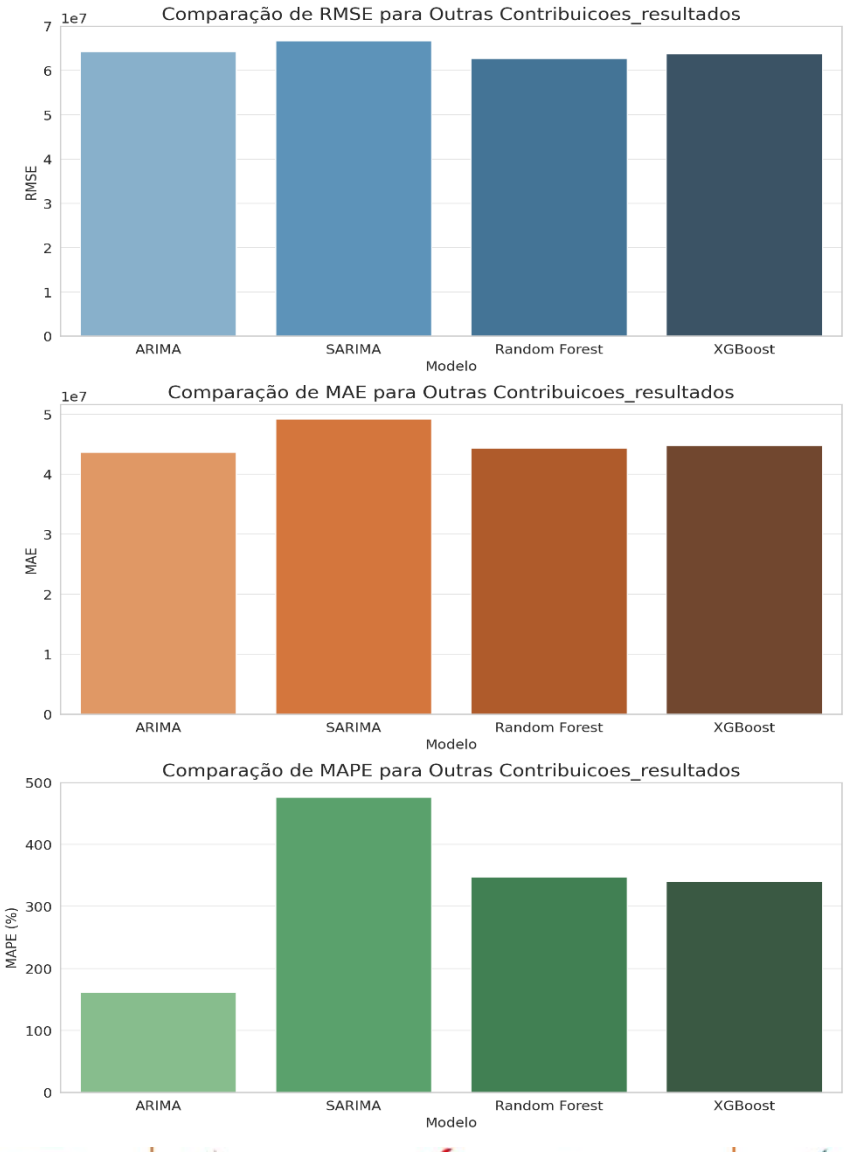
Figura 7. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para demais receitas patrimoniais.



- **Outras Contribuições:** Random Forest e XGBoost foram competitivos, com o Random Forest apresentando o menor RMSE, sugerindo uma melhor capacidade de adaptação a padrões complexos e variáveis.

Modelo	RMSE	MAE	MAPE
ARIMA	6.43e+07	4.37e+07	161.34
SARIMA	6.67e+07	4.92e+07	476.23
Random Forest	6.28e+07	4.44e+07	347.57
XGBoost	6.37e+07	4.49e+07	340.18

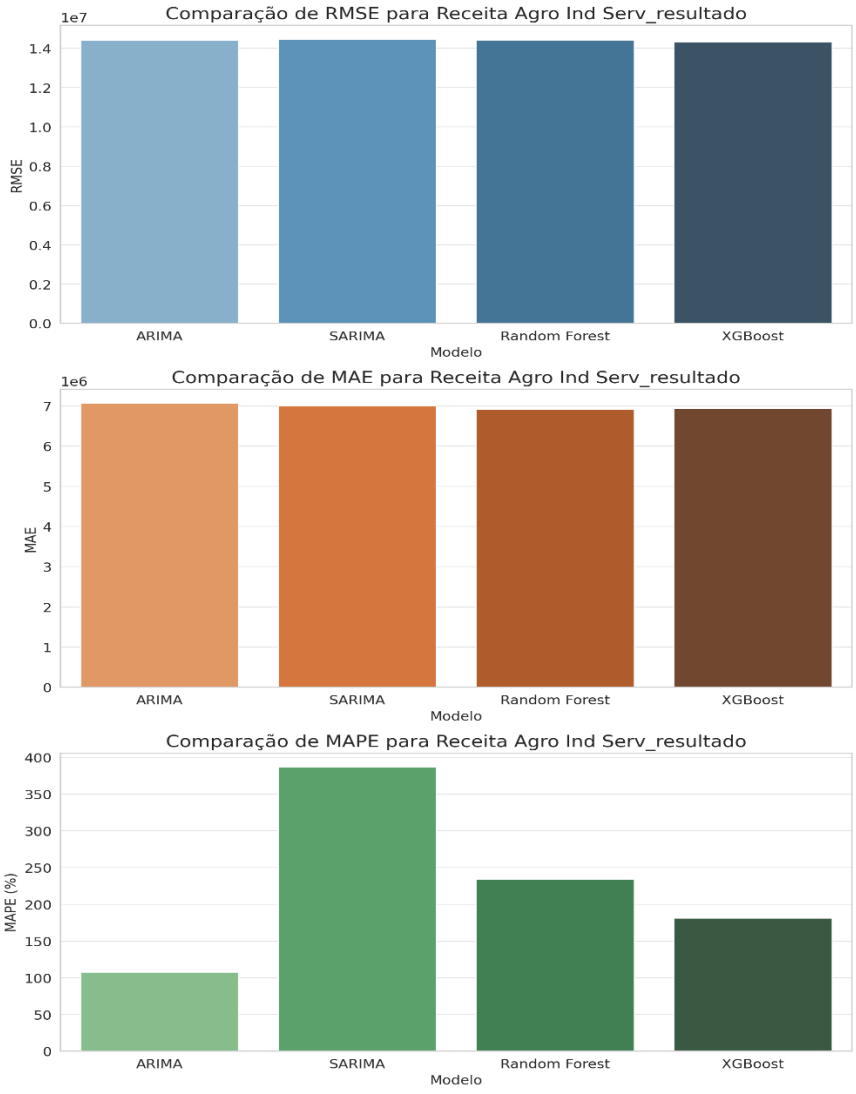
Figura 8. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para outras contribuições.



- **Receita de Agricultura, Indústria e Serviços:** Todos os modelos apresentaram desempenhos próximos, com o XGBoost tendo uma ligeira vantagem em RMSE, possivelmente devido à sua capacidade de lidar com relações não lineares e interações complexas entre variáveis.

Modelo	RMSE	MAE	MAPE
ARIMA	1.44e+07	7.07e+06	107.96
SARIMA	1.45e+07	7.01e+06	386.89
Random Forest	1.44e+07	6.92e+06	234.46
XGBoost	1.43e+07	6.94e+06	181.13

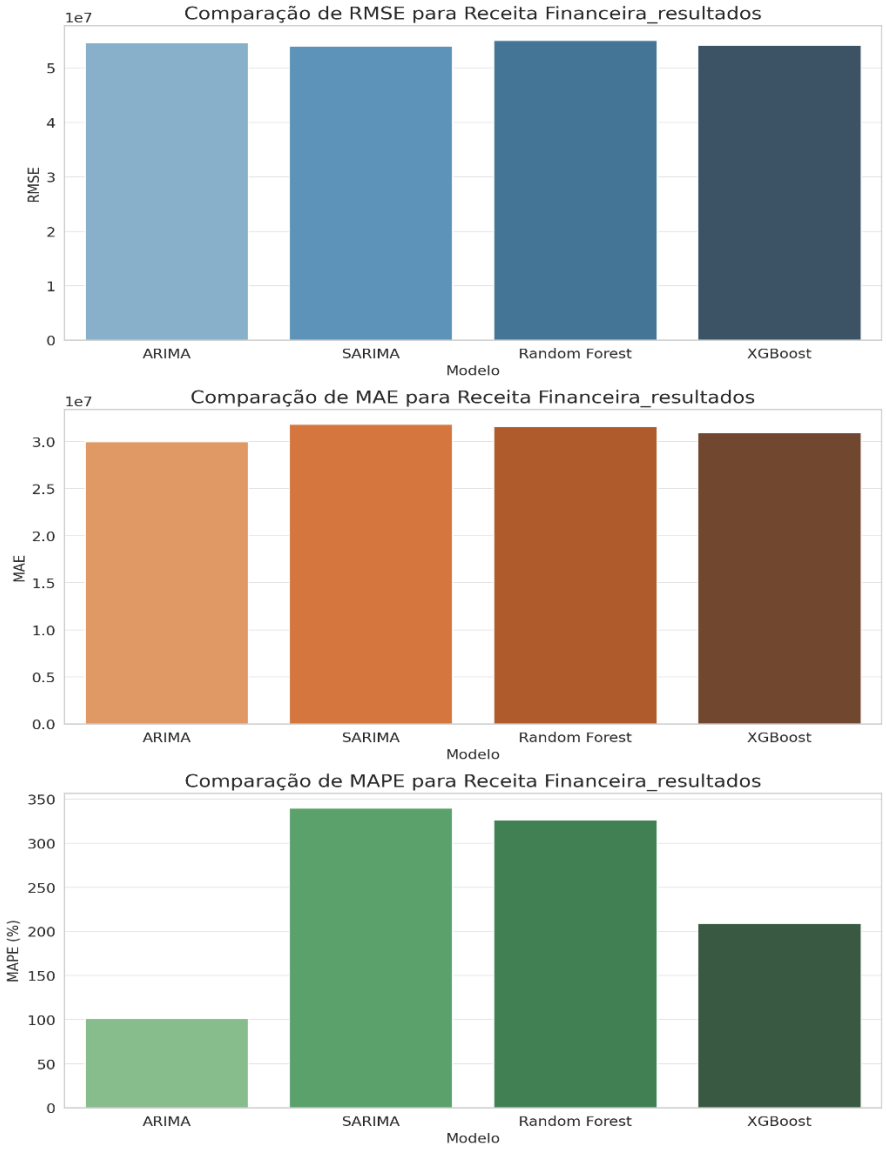
Figura 9. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para receita de agricultura, indústria e serviços.



- **Receita Financeira:** SARIMA e XGBoost foram muito semelhantes em termos de RMSE, com o SARIMA tendo um MAE ligeiramente superior, indicando uma precisão um pouco menor em previsões de curto prazo.

Modelo	RMSE	MAE	MAPE
ARIMA	5.47e+07	2.998e+07	101.19
SARIMA	5.40e+07	3.184e+07	339.80
Random Forest	5.51e+07	3.162e+07	326.34
XGBoost	5.42e+07	3.093e+07	209.43

Figura 9. Análise Comparativa de Métricas de Performace Entre Diversos Modelos de Previsão para receita financeira.



Análise Geral: Os resultados (figura 1) indicam que a escolha do modelo ideal depende significativamente do tipo específico de receita sendo modelada. Modelos de ML, como XGBoost e Random Forest, mostraram-se eficazes em categorias com padrões mais complexos e não lineares. Por outro lado, modelos como ARIMA e SARIMA foram mais adequados para séries com padrões claros de sazonalidade e estacionariedade. Esta análise ressalta a importância de uma abordagem personalizada e a necessidade de testar múltiplas abordagens de modelagem para alcançar as previsões mais precisas.

Desafios e Benefícios: Um desafio notável foi lidar com a não-estacionariedade das séries com alta volatilidade, choques e padrões irregulares, o que, em alguns casos, resultou em previsões menos precisas. Contudo, a aplicação de modelos de aprendizado de máquina em categorias com padrões mais complexos proporcionou insights valiosos, demonstrando a capacidade desses modelos de lidar com a complexidade dos dados financeiros. Outra dificuldade foi a necessidade de trabalhar com séries diferenciadas para lidar com a não-estacionariedade, o que implica em perda de informação da série temporal. Os benefícios para o Rio Grande do Sul incluem a possibilidade de previsões mais precisas e confiáveis de receitas orçamentárias, auxiliando na tomada de decisões fiscais informadas e na alocação eficiente de recursos. A aplicação desses modelos também pode contribuir para uma melhor compreensão das dinâmicas econômicas da região, permitindo uma resposta mais ágil a mudanças no cenário econômico.

Transferência de Tecnologia e Implementação Prática: Para efetivar a transferência de tecnologia deste estudo para o mundo real, é crucial estabelecer um plano de implementação que considere tanto os aspectos técnicos quanto os operacionais. Primeiramente, seria necessário criar um pipeline de dados e MLOps para processar, automatizar e gerar novos modelos que se adaptem a mudanças. Além disso, seria importante desenvolver uma interface de usuário amigável, que permita aos gestores acessar facilmente as previsões, ajustar parâmetros e hiperparâmetros conforme necessário e visualizar os resultados de maneira intuitiva.

Para garantir a sustentabilidade e a evolução contínua do projeto, seria recomendável estabelecer um protocolo de feedback contínuo, onde os usuários finais possam reportar suas experiências, dificuldades e sugestões de melhorias. Isso permitiria ajustes e atualizações regulares dos modelos, assegurando que eles permaneçam precisos e relevantes ao longo do tempo. Por fim, a publicação de estudos de caso e relatórios de impacto demonstrando os benefícios concretos da

implementação desses modelos poderia incentivar a adoção de abordagens semelhantes em outras regiões e setores, ampliando o impacto positivo deste trabalho.

5. CONSIDERAÇÕES FINAIS

Neste estudo, abordamos a complexidade das séries temporais financeiras do Rio Grande do Sul, aplicando e comparando diversos modelos de ML e estatística, como ARIMA, SARIMA, Random Forest e XGBoost. Através desta análise, identificamos que a eficácia de cada modelo varia em função das características específicas de cada categoria de receita orçamentária. Modelos como XGBoost e Random Forest demonstraram maior flexibilidade e precisão em capturar padrões complexos e não lineares, enquanto ARIMA e SARIMA se mostraram mais adequados para séries com padrões claros de sazonalidade e estacionariedade. Esta diferenciação de desempenho ressalta a importância de uma seleção cuidadosa de modelos, adaptada às particularidades de cada série temporal.

A aplicação prática desses achados pode ser um avanço na gestão financeira e na formulação de políticas públicas no Rio Grande do Sul e outros entes federativos. A implementação desses modelos avançados de previsão pode oferecer aos tomadores de decisão insights oportunos, permitindo uma alocação de recursos mais precisos e uma resposta mais ágil às mudanças econômicas. Além disso, a integração dessas técnicas de previsão com as práticas de gestão fiscal pode melhorar significativamente os orçamentos públicos, contribuindo para uma gestão financeira mais robusta e sustentável.

REFERÊNCIAS

AUERBACH, A. J.; GORODNICHENKO, Y. Fiscal Multipliers in Recession and Expansion. In: **Fiscal Policy after the Financial Crisis.** University of Chicago Press, 2012.

BREIMAN, L. Random Forests. **Machine Learning,** 2001.

BISHOP, C. M. **Pattern Recognition and Machine Learning.** Springer, 2006.

BOX, G. E.; JENKINS, G. M. **Time Series Analysis: Forecasting and Control.** John Wiley & Sons, 2015.

CAMPBELL, J. Y.; LO, A. W.; MACKINLAY, A. C. **The Econometrics of Financial Markets.** Princeton University Press, 1997.

CLEVELAND, R.B.; CLEVELAND, W.S.; MCRAE, J.E.; TERPENNING, I. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. **Journal of Official Statistics,** 1990.

DURBIN, J.; KOOPMAN, S.J. Time Series Analysis by State Space Methods. **Oxford University Press,** 2012.

DRUCKER, H.; BURGES, C.J.C.; KAUFMAN, L.; SMOLA, A.; VAPNIK, V. Support Vector Regression Machines. **Advances in Neural Information Processing Systems,** 1997.

FLUNKERT, V.; SALINAS, D.; GASTHAUS, J. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. **International Journal of Forecasting,** 2020.

FREUND, Y.; SCHAPIRE, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. **Journal of Computer and System Sciences,** 1997.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow.** O'Reilly Media, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning.** MIT Press, 2016.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation,** 1997.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning.** Springer, 2013.

LÜTKEPOHL, H. **New Introduction to Multiple Time Series Analysis.** Springer, 2005.

RANGAPURAM, S. S.; SEEGER, M.; GASTHAUS, J.; et al. Deep State Space Models for Time Series Forecasting. **Journal Name,** 2018.

ROMER, D. **Advanced Macroeconomics.** McGraw-Hill Education, 2019.

SMYL, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. **International Journal of Forecasting,** 2020.

TSAY, R. S. **Analysis of Financial Time Series.** Wiley-Interscience, 2005.

ZHANG, G. P. et al. Machine Learning in Time Series Forecasting. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, 2022.

ANEXOS

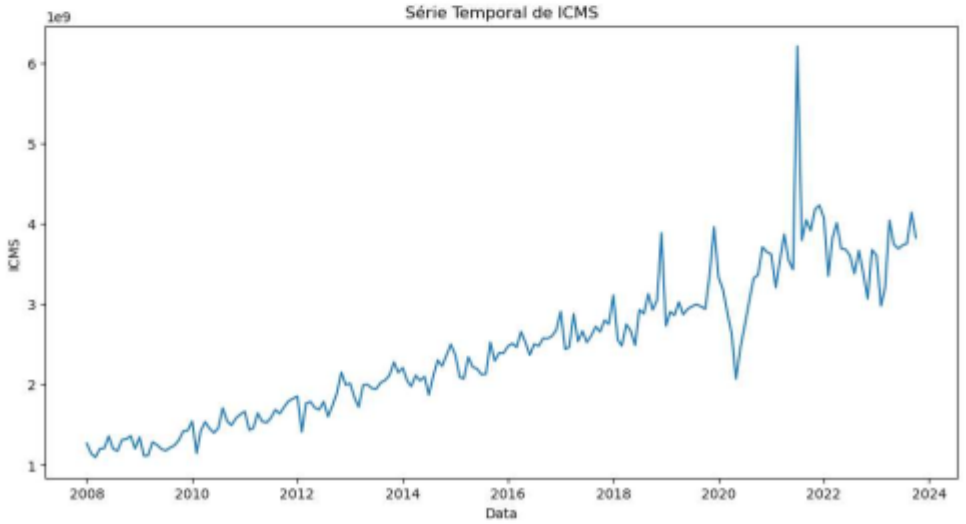
ANEXO 1. Exemplo do Código Fonte de uma das receitas analisadas (ICMS).

In [1]:
```
import pandas as pd
import numpy as np
from scipy.stats import boxcox
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_absolute_error, mean_squared_error
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from math import sqrt
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt
```

In [2]:
```
# Carregando o conjunto de dados
data = pd.read_excel('dataset.xlsx')
```

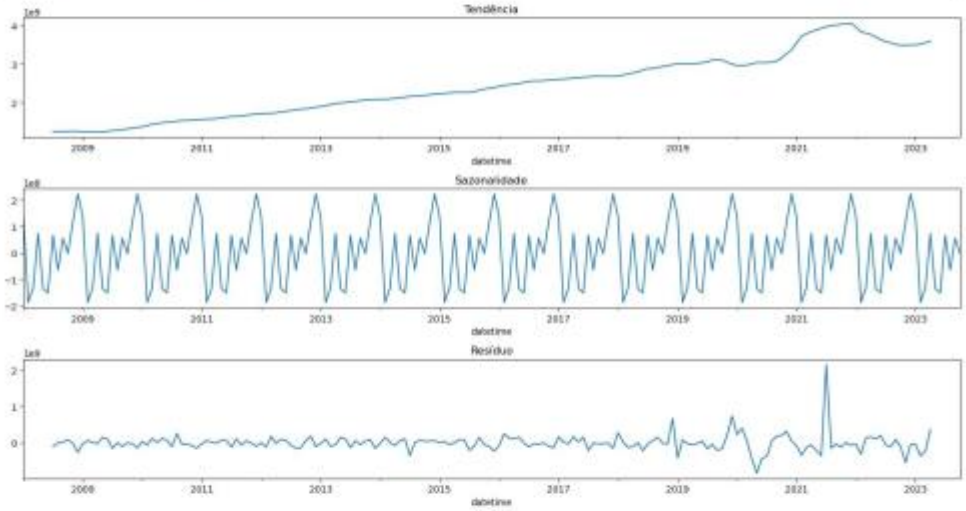
In [3]:
```
 # Configurando a coluna datetime como índice
data['datetime'] = pd.to_datetime(data['datetime'])
data.set_index('datetime', inplace=True)

# Verificando se o índice é um DatetimeIndex
if not isinstance(data.index, pd.DatetimeIndex):
    data.index = pd.DatetimeIndex(data.index.values)

# Tentando definir a frequência
data.index.freq = 'MS'  # Definindo a frequência como início do mês
```

In [4]:
```
# Plotar série temporal
plt.figure(figsize=(12, 6))
plt.plot(data['ICMS'])
plt.title('Série Temporal de ICMS')
plt.xlabel('Data')
plt.ylabel('ICMS')
plt.show()

# Estatísticas Descritivas
print(data['ICMS'].describe())
```



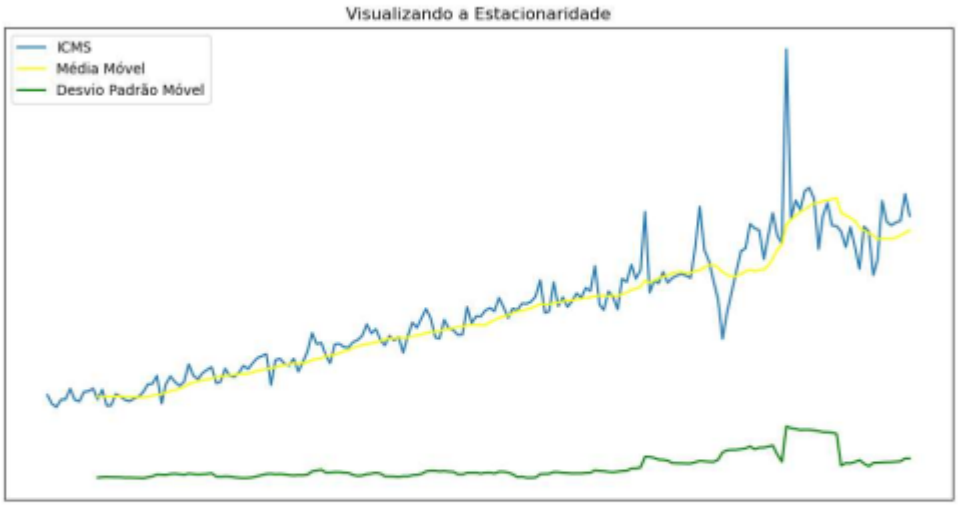
```
count    1.900000e+02
mean     2.439081e+09
std      8.887730e+08
min      1.093966e+09
25%      1.707057e+09
50%      2.389431e+09
75%      2.991068e+09
max      6.204908e+09
Name: ICMS, dtype: float64
```

In [5]:
```python
# Decomposição
decomposicao = seasonal_decompose(data['ICMS'], model='additive')

# Plotar a decomposição
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(15, 8))
decomposicao.trend.plot(ax=ax1)
ax1.set_title('Tendência')
decomposicao.seasonal.plot(ax=ax2)
ax2.set_title('Sazonalidade')
decomposicao.resid.plot(ax=ax3)
ax3.set_title('Resíduo')
plt.tight_layout()
```



In [6]:
```python
# Função para visualizar estacionaridade
def visualiza_estacionaridade(dados_serie):

    # Calcula média e desvio padrão móveis
    rolling_mean = dados_serie.rolling(12).mean()
    rolling_std = dados_serie.rolling(12).std()

    # Plot
    fig = plt.figure(figsize = (12,6))
    time_series = plt.plot(dados_serie, label = 'ICMS')
    mean = plt.plot(rolling_mean, color = 'yellow', label = "Média Móvel")
    std = plt.plot(rolling_std, color = 'green', label = "Desvio Padrão Móvel")

    plt.legend(loc = 'best')
    plt.title("Visualizando a Estacionaridade")
    plt.xticks([])
    plt.yticks([])
    plt.show()
```

In [7]:
```
# Aplica a função
visualiza_estacionaridade(data['ICMS'])
```



In [8]:
```
# Selecionar a série temporal para teste, por exemplo, a coluna ICMS
serie_temporal = data['ICMS']

# Realizar o teste de Dickey-Fuller Aumentado
resultado_adf = adfuller(serie_temporal, autolag='AIC')

print('Estatística ADF:', resultado_adf[0])
print('p-valor:', resultado_adf[1])
print('Valores Críticos:')
for key, value in resultado_adf[4].items():
    print('\t%s: %.3f' % (key, value))

# Interpretando os resultados
if resultado_adf[1] < 0.05:
    print("A série temporal é estacionária")
else:
    print("A série temporal não é estacionária")
```

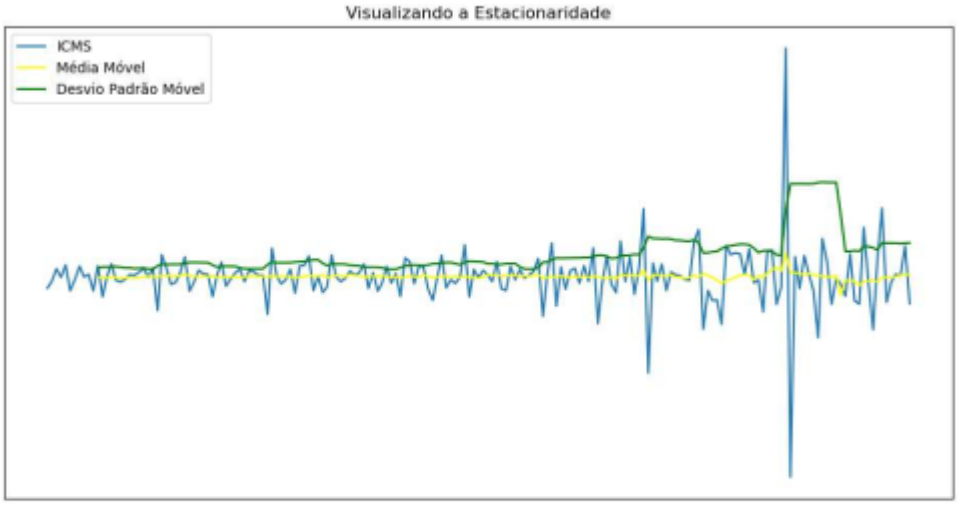
```
Estatística ADF: -0.5509931228068242
p-valor: 0.8816756112091862
Valores Críticos:
        1%: -3.467
        5%: -2.878
        10%: -2.575
A série temporal não é estacionária
```

In [9]:
```
# Diferenciação para tornar a série estacionária
serie_diferenciada = serie_temporal.diff().dropna()

# Novamente, aplicando o teste ADF na série diferenciada
resultado_adf_diferenciada = adfuller(serie_diferenciada, autolag='AIC')
print('Estatística ADF da série diferenciada:', resultado_adf_diferenciada[0])
print('p-valor:', resultado_adf_diferenciada[1])
```

```
Estatística ADF da série diferenciada: -7.769588193903824
p-valor: 8.995329521017501e-12
```

In [10]:
```
# Diferenciando a série temporal
data_dif = data.diff().dropna()
```

In [11]:
```
# Aplica a função
visualiza_estacionaridade(data_dif['ICMS'])
```



In [12]:
```
# Preparando os dados para modelagem
y = data_dif['ICMS']
X = data_dif.drop(['ICMS'], axis=1)
```

In [13]:
```
# Definir a grade de hiperparâmetros para testar
param_grid_xgb = {
    'n_estimators': [100, 500, 1000],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.7, 0.8, 0.9]
}

# Inicializar o modelo
xgb = XGBRegressor(random_state=0)

# Configurar a busca em grade
grid_search_xgb = GridSearchCV(estimator=xgb, param_grid=param_grid_xgb, cv=3, n_jo

# Executar a busca em grade
grid_search_xgb.fit(X, y)

# Melhores parâmetros
print("Melhores hiperparâmetros para XGBoost:", grid_search_xgb.best_params_)
```

```
Melhores hiperparâmetros para XGBoost: {'learning_rate': 0.01, 'max_depth': 3, 'n_
estimators': 100, 'subsample': 0.7}
```

In [14]:
```
# Configuração da validação cruzada temporal
tscv = TimeSeriesSplit(n_splits=5)
```

In [15]:
```
# Função para validar um modelo com gap 1

def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    non_zero_index = y_true != 0
    return np.mean(np.abs((y_true[non_zero_index] - y_pred[non_zero_index]) / y_tru

def validar_modelo(modelo, X, y, is_time_series=False):
    rmse_erros = []
    mae_erros = []
```

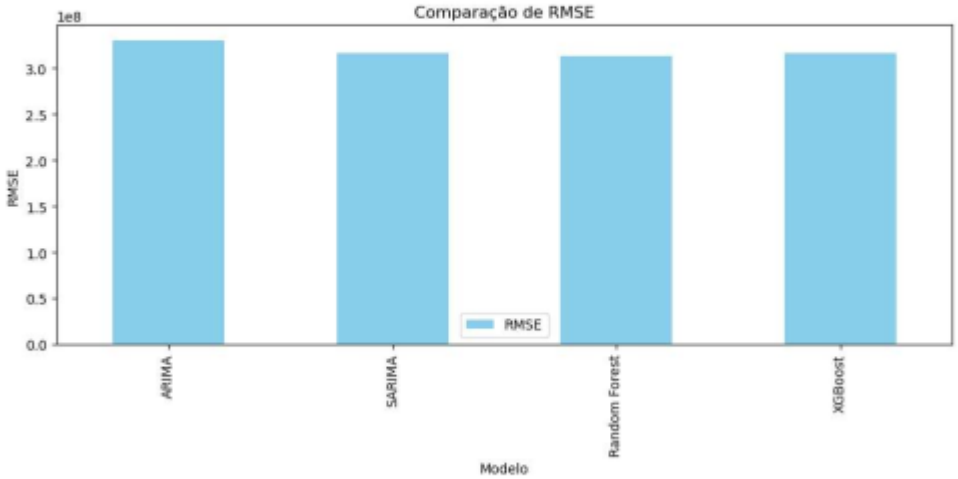
```
    mape_erros = []

    tscv = TimeSeriesSplit(n_splits=5)

    for train_index, test_index in tscv.split(X):
        X_treino, X_teste = X.iloc[train_index], X.iloc[test_index]
        y_treino, y_teste = y.iloc[train_index], y.iloc[test_index]

        if is_time_series:
            modelo_fit = modelo(y_treino).fit()
            previsoes = modelo_fit.forecast(steps=len(y_teste))
        else:
            modelo.fit(X_treino, y_treino)
            previsoes = modelo.predict(X_teste)

        rmse_erros.append(sqrt(mean_squared_error(y_teste, previsoes)))
        mae_erros.append(mean_absolute_error(y_teste, previsoes))
        mape_erros.append(mean_absolute_percentage_error(y_teste, previsoes))

    return np.mean(rmse_erros), np.mean(mae_erros), np.mean(mape_erros)
```

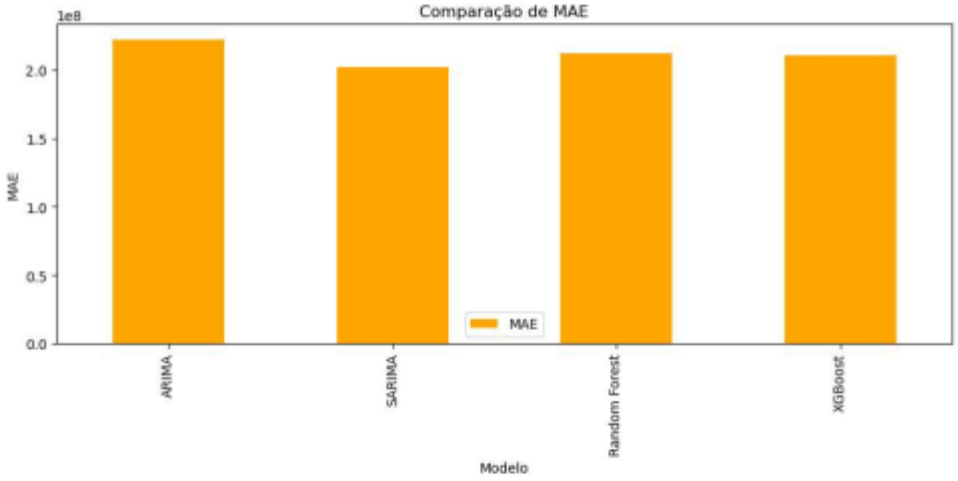
In [16]:
```
# Validação cruzada para cada modelo
erro_arima = validar_modelo(lambda y: ARIMA(y, order=(1, 0, 1)), X, y, is_time_seri
erro_sarima = validar_modelo(lambda y: SARIMAX(y, order=(1, 0, 1), seasonal_order=(
erro_rf = validar_modelo(RandomForestRegressor(n_estimators=100, random_state=0), X
erro_xgb = validar_modelo(XGBRegressor(n_estimators=100, learning_rate=0.01, max_de
```

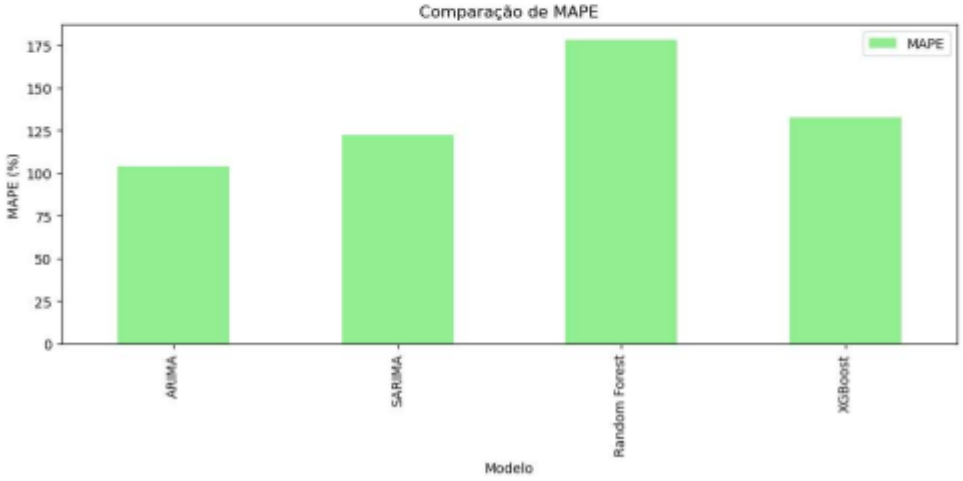
```
C:\ProgramData\anaconda3\Lib\site-packages\statsmodels\tsa\statespace\sarimax.py:9
78: UserWarning: Non-invertible starting MA parameters found. Using zeros as start
ing parameters.
  warn('Non-invertible starting MA parameters found.'
C:\ProgramData\anaconda3\Lib\site-packages\statsmodels\tsa\statespace\sarimax.py:8
66: UserWarning: Too few observations to estimate starting parameters for seasonal
ARMA. All parameters except for variances will be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
```

In [17]:
```
# Resultados da validação cruzada
resultados = {
    'Modelo': ['ARIMA', 'SARIMA', 'Random Forest', 'XGBoost'],
    'RMSE': [erro_arima[0], erro_sarima[0], erro_rf[0], erro_xgb[0]],
    'MAE': [erro_arima[1], erro_sarima[1], erro_rf[1], erro_xgb[1]],
    'MAPE': [erro_arima[2], erro_sarima[2], erro_rf[2], erro_xgb[2]]
}

df_resultados = pd.DataFrame(resultados)

# Criar gráficos de barras para cada métrica
fig, axs = plt.subplots(3, 1, figsize=(10, 15))

# RMSE
df_resultados.plot(x='Modelo', y='RMSE', kind='bar', ax=axs[0], color='skyblue')
axs[0].set_title('Comparação de RMSE')
axs[0].set_ylabel('RMSE')

# MAE
df_resultados.plot(x='Modelo', y='MAE', kind='bar', ax=axs[1], color='orange')
axs[1].set_title('Comparação de MAE')
axs[1].set_ylabel('MAE')

# MAPE
df_resultados.plot(x='Modelo', y='MAPE', kind='bar', ax=axs[2], color='lightgreen')
axs[2].set_title('Comparação de MAPE')
axs[2].set_ylabel('MAPE (%)')
```

```
plt.tight_layout()
plt.show()

# Imprimir a tabela com os resultados
print("Resultados da Validação Cruzada:")
print(df_resultados)
```







```
Resultados da Validação Cruzada:
           Modelo          RMSE           MAE        MAPE
0           ARIMA  3.300821e+08  2.223214e+08  103.833911
1          SARIMA  3.162627e+08  2.021772e+08  122.179364
2   Random Forest  3.127896e+08  2.118552e+08  178.220361
3         XGBoost  3.163583e+08  2.105517e+08  132.783534
```

In [ ]: