

DESENVOLVIMENTO DE UM SOFTWARE DE PREVISÃO DE DEMANDA DE PRODUTOS LÁCTEOS

Luan Dutra

Trabalho orientado pelo Prof. Dr. Ailton Luiz Dias Siqueira Junior

Resumo - A previsão de demanda é crucial para que as empresas se planejem e garantam recursos suficientes para atender aos pedidos, evitando excessos que possam levar ao vencimento de produtos em estoque e o desabastecimento, o que poderia resultar em perda de market share para concorrentes. Muitas vezes esse processo é feito manualmente, baseados em estimativas humanas, que consomem tempo e esforço. Recentemente têm surgido modelos estatísticos, que realizam essa tarefa de forma automatizada a partir de dados históricos. Esse trabalho visa desenvolver um programa para analisar o histórico de vendas de produtos lácteos e prever a demanda futura usando modelos estatísticos e inteligência artificial. Foi utilizada a linguagem Python, que oferece diversas ferramentas para a criação de gráficos e dashboards, além de bibliotecas para análise estatística e ciência de dados. Os modelos de previsão que serão comparados são chamados de Prophet, NeuralProphet, ARIMA e SARIMA, que terão sua precisão analisada em 3 produtos diferentes.

Palavras-Chave - Análise de dados, Ciência de dados, estatística, inteligência artificial, otimização de recursos, previsão de demanda.

DEVELOPMENT OF A DAIRY PRODUCT DEMAND FORECASTING SOFTWARE

Abstract - Demand forecasting is crucial for companies to plan and guarantee sufficient resources to meet orders, avoiding excesses that could lead to products in stock expiring and shortages, which could result in loss of market share to competitors. This process is often done manually, based on human estimates, which consume time and effort. Recently, statistical models have emerged that perform this task in an automated way based on historical data. This work aims to develop a program to analyze the history of dairy product sales and predict future demand using statistical models and artificial intelligence. The Python language was used, which offers several tools for creating graphs and dashboards, as well as libraries for statistical analysis and data science. The prediction models that will be compared are called Prophet, NeuralProphet, ARIMA and SARIMA, which will have their accuracy analyzed in 3 different products.

Keywords - Artificial intelligence, Data analysis, Data science, Demand forecasting, Resource optimization, Statistics

1 Introdução

O mercado de alimentos frescos no Brasil tem passado por uma transformação significativa, com os consumidores demonstrando forte preferência por marcas locais, especialmente em produtos perecíveis como leite, queijos e iogurte, segundo estudos da AC Nielsen®.(MILKPOINT, 2024). Essa escolha é impulsionada pela busca por frescor, qualidade e segurança, com metade

dos consumidores optando por produtos de origem regional, o que impõe uma responsabilidade maior às empresas para atender à demanda local. Influências externas, como flutuações econômicas e mudanças nos padrões de consumo, também desempenham um papel crucial. A pandemia de COVID-19, por exemplo, alterou a dinâmica de compra, levando os consumidores a priorizar produtos de fácil acesso e maior segurança sanitária onde muitas pessoas se mantiveram fiéis as marcas escolhidas nesse período.

Entretanto, o setor enfrenta desafios significativos, como a volatilidade nos preços dos insumos e a necessidade de uma logística eficiente para garantir a entrega de produtos frescos. As empresas locais precisam lidar com a incerteza na demanda, que pode variar drasticamente de uma semana para outra. Isso torna ainda mais essencial o desenvolvimento de modelos de previsão de demanda que possam se adaptar rapidamente a essas mudanças e oferecer previsões mais precisas(MILKPOINT, 2024).

Historicamente, o setor utilizou modelos tradicionais de previsão, como a Média Móvel Simples (MMS), a Média Móvel Ponderada (MMP), a Suavização Exponencial Simples (SES), o modelo ARIMA, entre outros. Esses métodos, embora úteis, muitas vezes apresentam limitações em termos de precisão e capacidade de adaptação a padrões complexos de consumo. A transição para modelos mais avançados é uma necessidade premente, pois as empresas buscam maneiras de otimizar seus processos e atender melhor às expectativas dos consumidores(FILHO, 2018).

Recentemente surgiram novos modelos como o Prophet, desenvolvido pelo Facebook que é projetado para lidar com dados sazonais e eventos de tendência, oferecendo flexibilidade e facilidade de uso. Já o NeuralProphet, uma extensão do Prophet, incorpora redes neurais para melhorar a precisão das previsões, permitindo a captura de padrões mais complexos. Ambos podem representar uma evolução significativa na previsão de demanda, prometendo resultados mais robustos e confiáveis em comparação aos métodos tradicionais(RAFFERTY, 2021).

Neste contexto, o objetivo deste artigo é analisar os novos modelos de previsão de séries temporais, Prophet e NeuralProphet contra os modelos ARIMA e SARIMA . A análise busca destacar as vantagens e limitações de cada abordagem, contribuindo assim para uma compreensão mais profunda das melhores práticas em previsão de demanda no setor de alimentos frescos.

2 Metodologia

2.1 Prophet

O Prophet é uma ferramenta avançada de previsão de séries temporais projetada para lidar com dados que apresentam tendências e sazonalidades complexas. Este modelo matemático analisa e separa diferentes componentes que influenciam a série, permitindo que cada um seja estimado de forma independente. A combinação dessas estimativas resulta em previsões mais precisas e robustas (RAFFERTY, 2021). A Equação 1 que fundamenta essa abordagem é:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (1)$$

Onde:

- $y(t)$ - Valor previsto no tempo t .
- $g(t)$ - Componente de tendência geral.
- $s(t)$ - Componente de sazonalidade.
- $h(t)$ - Componente de feriados/eventos.
- ϵ_t - Termo de erro/Variabilidade

No Prophet, a tendência pode ser calculada de duas formas principais: por meio de uma função de crescimento linear, onde a série cresce ou decresce a uma taxa constante ao longo do tempo, ou por meio de uma função de crescimento logístico, que é ideal para séries que possuem uma capacidade máxima ou limite de saturação. Na função logística, a tendência segue uma curva do tipo sigmóide e se aproxima de um limite ao crescer, ajustando-se ao comportamento real de séries com restrições de crescimento. Ambas as opções permitem capturar diferentes tipos de variação de longo prazo na série (RAFFERTY, 2021).

A sazonalidade no Prophet é calculada por meio de uma série de Fourier, que permite capturar padrões recorrentes em diferentes períodos, como diários, semanais ou anuais. Essa abordagem utiliza funções trigonométricas para representar flutuações cíclicas nos dados, permitindo que o modelo identifique e ajuste as variações sazonais de forma precisa. Por exemplo, ela pode detectar variações nas vendas relacionadas à temperatura ao longo das estações do ano. Essa flexibilidade torna o Prophet eficaz para lidar com padrões sazonais que podem ocorrer em diversas escalas de tempo (RAFFERTY, 2021).

O componente de feriados no Prophet é modelado para capturar o impacto de datas es-

pecíficas que podem causar variações significativas nos dados, como eventos pontuais como feriados nacionais. Esse componente utiliza uma abordagem que inclui uma função indicadora para identificar a presença de feriados, permitindo ao modelo ajustar as previsões para picos ou quedas temporárias associados a esses dias. Por exemplo, pode-se observar um aumento nas vendas durante o Natal ou uma diminuição durante feriados prolongados. Ao incorporar esses efeitos pontuais, o Prophet melhora a precisão das previsões ao considerar influências externas que podem alterar o comportamento normal da série temporal (RAFFERTY, 2021).

2.2 NeuralProphet

O NeuralProphet é uma extensão do modelo Prophet, que combina as técnicas tradicionais de previsão de séries temporais com a abordagem de aprendizado profundo. Essa ferramenta foi desenvolvida para melhorar a precisão das previsões, incorporando redes neurais, enquanto mantém a interpretabilidade e a flexibilidade que caracterizam o Prophet (TRIEBE et al., 2021). A Equação 2 utilizada no NeuralProphet é semelhante à do Prophet, mas com a adição de uma rede neural para modelar padrões complexos nos dados:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t + n(t) \quad (2)$$

Onde:

- $y(t)$ - Valor previsto no tempo t .
- $g(t)$ - Componente de tendência geral.
- $s(t)$ - Componente de sazonalidade.
- $h(t)$ - Componente de feriados/eventos.
- ϵ_t - Termo de erro/Variabilidade.
- $n(t)$ - Componente adicional gerado pela rede neural.

A tendência no NeuralProphet é similar à do Prophet, permitindo o uso de crescimento linear ou logístico. No entanto, o NeuralProphet introduz flexibilidade adicional ao modelar a tendência, utilizando redes neurais para capturar mudanças abruptas e não lineares na série temporal, adaptando-se melhor a padrões complexos (TRIEBE et al., 2021).

A sazonalidade é abordada de maneira semelhante, mas com uma estrutura neural que aprimora a detecção de padrões mais sutis e dinâmicos, além de permitir sazonalidades personalizadas, oferecendo a capacidade de modelar flutuações em diferentes escalas com mais detalha-

mento (TRIEBE et al., 2021).

O componente de feriados é integrado de forma análoga ao Prophet, porém com a possibilidade de ajustar automaticamente as influências desses dias com base em dados históricos e padrões aprendidos pela rede neural. Isso resulta em previsões que consideram tanto o impacto imediato dos feriados quanto interações mais complexas ao longo do tempo (TRIEBE et al., 2021).

Em suma, o NeuralProphet combina a robustez da modelagem clássica de séries temporais com a adaptabilidade das redes neurais, oferecendo uma solução poderosa para previsões em larga escala, mantendo a transparência e a explicabilidade dos resultados (TRIEBE et al., 2021).

2.3 ARIMA

O ARIMA (AutoRegressive Integrated Moving Average) é um dos modelos mais tradicionais e amplamente utilizados para previsão de séries temporais. Sua abordagem é baseada em três componentes principais: autoregressivo (AR), média móvel (MA) e diferenciação (I), que permitem capturar as dependências temporais e as características subjacentes dos dados. O modelo ARIMA é especialmente eficaz para séries temporais que apresentam padrões de tendência e ruído, sem a presença de sazonalidades muito complexas. (BOX et al., 2015) A Equação 3 que descreve o modelo ARIMA é dada por:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (3)$$

Onde:

- y_t - Valor observado no tempo t .
- ϕ_1, \dots, ϕ_p - Parâmetros autoregressivos (AR).
- $\theta_1, \dots, \theta_q$ - Parâmetros de média móvel (MA).
- ϵ_t - Termo de erro/resíduo no tempo t .
- p - Número de lags autoregressivos (AR).
- q - Número de lags de média móvel (MA).
- d - Número de diferenciações para tornar a série estacionária.

O componente autoregressivo (AR) modela a dependência de uma variável com seus próprios valores passados, ou seja, a relação entre o valor da série no tempo t e os valores anteriores. O componente de média móvel (MA) ajuda a reduzir grandes saltos de variação mas acaba

por ser redundante pois essa propriedade já foi adicionada manualmente em todos os modelos para terem seus resultados comparados. Já a diferenciação (d) é aplicada para tornar a série temporal estacionária, removendo tendências e tornando os dados mais adequados para modelagem.(BOX et al., 2015)

ARIMA é um modelo flexível para previsão de séries temporais, especialmente quando as séries não apresentam sazonalidade ou quando essa sazonalidade é tratada separadamente. Contudo, sua limitação está na dificuldade de lidar com dados altamente sazonais, o que levou ao desenvolvimento de modelos como SARIMA (ARIMA sazonal). Ainda assim, o ARIMA continua sendo uma abordagem robusta para uma grande variedade de dados temporais(BOX et al., 2015)

2.4 SARIMA

O modelo SARIMA (Seasonal AutoRegressive Integrated Moving Average) é uma extensão do ARIMA desenvolvida para lidar explicitamente com séries temporais que apresentam sazonalidade. Enquanto o ARIMA modela tendências e ruídos por meio de componentes autoregressivos, de média móvel e diferenciação, o SARIMA acrescenta termos sazonais que permitem capturar padrões que se repetem em intervalos regulares, como semanas, meses ou anos (BOX et al., 2015).

O modelo é formalmente representado como $SARIMA(p, d, q)(P, D, Q)_s$, onde:

- p, d, q representam, respectivamente, os parâmetros autoregressivo, de diferenciação e de média móvel para a série não sazonal.
- P, D, Q são os mesmos parâmetros, mas aplicados ao componente sazonal.
- s corresponde ao período da sazonalidade (por exemplo, $s = 12$ para dados mensais com sazonalidade anual).

A equação geral do SARIMA pode ser expressa da seguinte forma:

$$\Phi_P(L^s)\phi_p(L)(1-L)^d(1-L^s)^D y_t = \Theta_Q(L^s)\theta_q(L)\epsilon_t \quad (4)$$

Onde:

y_t	- Valor observado no tempo t .
$\phi_p(L)$	- Polinômio autoregressivo não sazonal de ordem p .
$\theta_q(L)$	- Polinômio de média móvel não sazonal de ordem q .
$\Phi_P(L^s)$	- Polinômio autoregressivo sazonal de ordem P .
$\Theta_Q(L^s)$	- Polinômio de média móvel sazonal de ordem Q .
$(1 - L)^d$	- Operador de diferenciação não sazonal de ordem d .
$(1 - L^s)^D$	- Operador de diferenciação sazonal de ordem D .
ϵ_t	- Termo de erro/resíduo no tempo t .

O grande diferencial do SARIMA é a capacidade de modelar séries em que os padrões sazonais têm impacto significativo, como ocorre nas vendas de produtos lácteos, que podem variar em função de períodos festivos ou estações do ano. Assim, o SARIMA consegue representar tanto os efeitos de curto prazo quanto os ciclos sazonais de longo prazo.

Embora o modelo ofereça maior flexibilidade, sua aplicação exige uma escolha criteriosa dos parâmetros, o que pode aumentar a complexidade computacional. Para isso, são utilizados critérios estatísticos como o AIC e o BIC, que auxiliam na seleção do conjunto de parâmetros mais adequado para capturar as características da série (BOX et al., 2015).

2.5 Auto_ARIMA

O Auto_ARIMA é uma extensão dos modelos ARIMA e SARIMA que automatiza o processo de identificação dos melhores parâmetros para o modelo, ou seja, os valores de p , d e q . Tradicionalmente, a escolha desses parâmetros exige análises manuais de autocorrelação (ACF) e autocorrelação parcial (PACF), além de múltiplos testes para encontrar a combinação que minimize critérios estatísticos como AIC (Akaike Information Criterion) e BIC (Bayesian Information Criterion) (HYNDMAN; ATHANASOPOULOS, 2018).

O Auto_ARIMA executa esses passos de forma automática, avaliando várias combinações de parâmetros e selecionando aquela que apresenta o melhor ajuste estatístico à série temporal. Isso torna o modelo ARIMA mais acessível, principalmente quando se trabalha com grandes volumes de dados ou múltiplos produtos, reduzindo o tempo necessário para ajuste manual.

No presente trabalho, o Auto_ARIMA foi utilizado para determinar os parâmetros iniciais do ARIMA e SARIMA, sendo posteriormente aplicados testes de previsão sobre os dados de vendas de produtos lácteos.

2.6 Pré-processamento

Durante o pré-processamento, serão aplicadas diversas alterações para melhorar a qualidade e a consistência dos dados históricos. Primeiramente, serão identificadas e removidas as linhas duplicadas que possuam a mesma data, evitando que informações redundantes interfiram nas previsões. Em seguida os nomes das colunas serão ajustados para tornar a interpretação dos dados mais intuitiva, e as datas serão formatadas corretamente, padronizando-as para facilitar o processamento posterior.

Para realizar essas modificações será utilizada a biblioteca Pandas em Python para manipulação e análise de dados. O Pandas oferece estruturas de dados flexíveis, como DataFrames, que possibilitam filtrar, corrigir e transformar dados de forma eficiente. Com o uso de suas funções, como `drop_duplicates` para remover duplicatas, conseguimos automatizar e agilizar o pré-processamento, assegurando que os dados de entrada estivessem adequados para as etapas de modelagem (MCKINNEY; TEAM, 2015).

2.7 Streamlit

Será utilizada a biblioteca streamlit para a criação da interface gráfica do programa, essa biblioteca foi desenvolvida para criar interfaces simples, rápidas e interativas a partir de scripts Python. Com apenas algumas linhas de código, é possível transformar visualizações de dados, gráficos, tabelas e até modelos de machine learning em aplicativos web com uma interface gráfica. Além disso, a biblioteca permite a integração com bibliotecas populares, como Pandas, Matplotlib, Plotly, TensorFlow e PyTorch, tornando-a uma ótima opção para quem deseja compartilhar resultados de maneira dinâmica e interativa sem se preocupar com configurações complexas de servidores ou front-end.

2.8 Validação

Para cada um dos modelos será criada uma aba no programa para avaliar sua taxa de acerto de sua previsão em relação aos dados do histórico, para isso 95% dos dados serão utilizados como entrada para ambos os modelos e o programa realizará a previsão até a data dos 5% faltantes e terá sua previsão comparada aos valores reais. Essa comparação será feita por meio de ferramentas estatísticas como o erro médio absoluto (MAE), erro médio percentual (MAPE) e desvio padrão (RMSE). (DEVELOPERS, 2024) A principal medida que será

analisada será o erro médio percentual que não deverá ser maior do que 10%, podendo ser considerado um sucesso caso a previsão consiga atingir esse requisito.

A Equação 5 define o Erro Médio Absoluto (MAE), que calcula a média dos erros absolutos entre os valores previstos e os valores reais:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

Onde:

- MAE - Erro médio absoluto.
- n - Número total de observações.
- y_i - Valor real observado no ponto i .
- \hat{y}_i - Valor previsto no ponto i .

A Equação 6 define o Erro Médio Percentual (MAPE), que calcula a média dos erros percentuais entre os valores previstos e reais:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (6)$$

Onde:

- $MAPE$ - Erro médio percentual.
- n - Número total de observações.
- y_i - Valor real observado no ponto i .
- \hat{y}_i - Valor previsto no ponto i .

A Equação 7 define a Raiz do Erro Quadrático Médio (RMSE), que calcula a raiz quadrada da média dos quadrados dos erros:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

Onde:

- $RMSE$ - Raiz do erro quadrático médio.
- n - Número total de observações.
- y_i - Valor real observado no ponto i .
- \hat{y}_i - Valor previsto no ponto i .

3 Resultados

A forma de obtenção dos dados vem por meio de uma planilha do excel com as datas e os respectivos valores de venda, e com esses dados é obtido o primeiro gráfico na Figura 1 mostrando o histórico de valores.

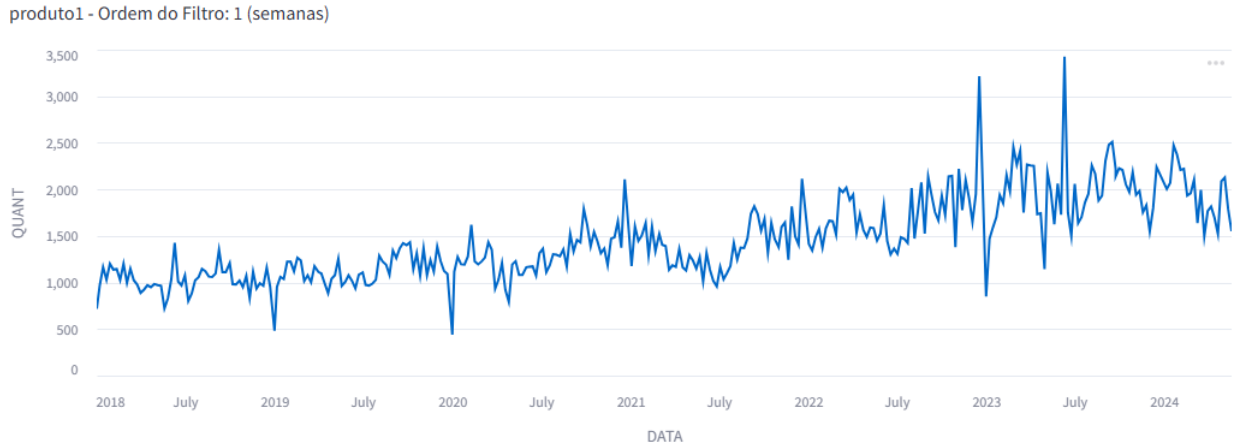


Figura 1: Histórico de valores.

3.1 Média móvel

A Equação 8 define a Média Móvel (Moving Average), que calcula a média de um conjunto de observações em uma janela de tempo ou número de períodos fixos. Essa técnica é amplamente utilizada para suavizar séries temporais e reduzir a variação nos dados, ajudando na identificação de tendências subjacentes.

$$MA_t = \frac{1}{k} \sum_{i=t-k+1}^t y_i \quad (8)$$

Onde:

- MA_t - Média móvel no tempo t .
- k - Número de períodos na janela de média.
- y_i - Valor real observado no ponto i .
- t - Ponto de tempo atual.

A média móvel é calculada tomando a média dos valores em uma janela de tempo k , onde k pode ser determinado com base no contexto da análise (por exemplo, 7 dias, 30 dias, etc.). Esse método é muito utilizado para suavizar os dados e remover flutuações de curto prazo, permitindo que tendências de longo prazo se tornem mais evidentes.(BOX et al., 2015)

É utilizado uma barra de arrastar no programa que permite suavizar esse gráfico com o histórico de valores utilizando-se de uma média móvel, que vai transformar o valor de uma data para ser a média dos n números anteriores sendo n um número de 1 a 52 semanas que foram os valores definidos pelo próprio programa. Um exemplo do gráfico com média móvel com valor 10 na Figura 2:

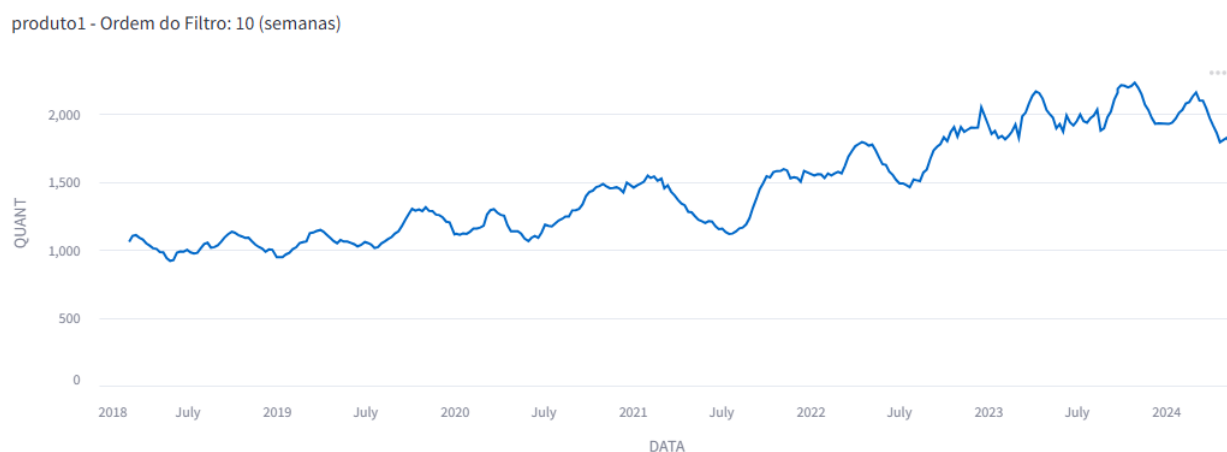


Figura 2: Histórico de valores com média móvel (10 períodos).

Essa média móvel serve para dar uma maior consistência aos dados fornecidos removendo maiores picos e deixando mais observável a tendência geral de vendas. Ao alterar o valor da média móvel, tanto o gráfico inicial como todos os outros subsequentes terão seus valores alterados de acordo com os novos dados

3.2 Boxplot das vendas anuais

Esse tipo de gráfico da Figura 3 consiste em dividir os valores em quartis tendo um traço na mediana e 25% dos dados em cada quartil, sendo útil para determinar a variação e tendência geral dos dados, e foi implementado com uma análise para cada ano:

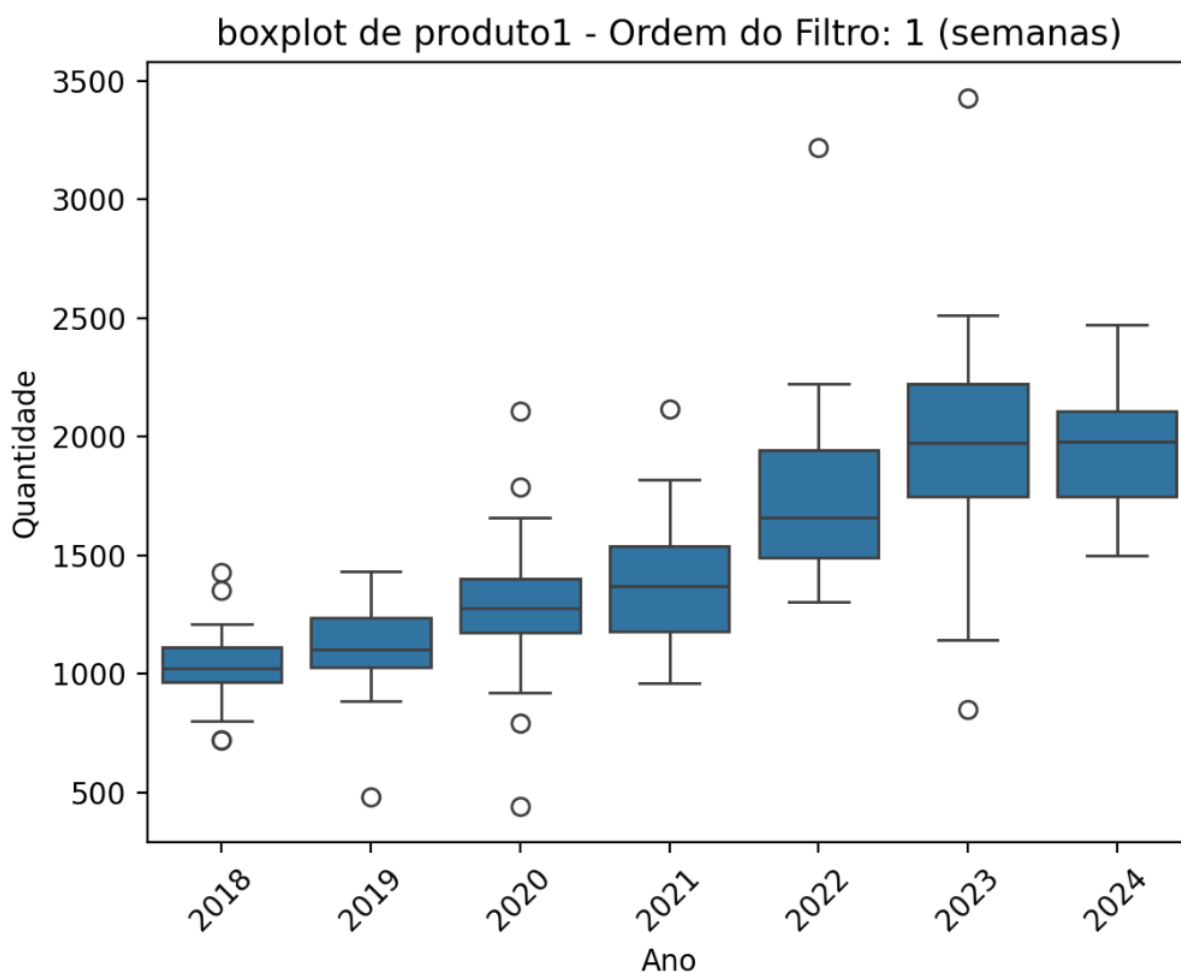


Figura 3: Boxplot das vendas anuais.

3.3 Remoção de outliers - IQR

Durante a análise do boxplot, observou-se a presença de valores atípicos (outliers), que poderiam distorcer o comportamento da série temporal e comprometer a qualidade das previsões. Para tratar esse problema, foi adotado o método do Intervalo Interquartil (IQR).

O IQR corresponde à diferença entre o terceiro quartil (Q_3) e o primeiro quartil (Q_1), representando a faixa central onde se encontram 50% dos dados. Sua fórmula é dada por:

$$IQR = Q_3 - Q_1$$

Com base nesse intervalo, definem-se os limites inferior e superior para detecção de outliers:

$$LimiteInferior = Q_1 - 1.5 \times IQR$$

$$LimiteSuperior = Q_3 + 1.5 \times IQR$$

Valores que se encontram abaixo do limite inferior ou acima do limite superior são considerados outliers. Assim, foram removidos pontos extremos que poderiam representar erros de registro ou eventos não recorrentes, garantindo maior consistência na série utilizada para modelagem.

3.4 Modelo Prophet

Utilizando-se dos valores mostrados no histórico de valores e da biblioteca Prophet, é possível apresentar o primeiro modelo na Figura 4. Esse modelo possui os valores do histórico como pontos pretos, a estimativa de previsão mínima e máxima entre a faixa azul claro e a previsão média (linha azul escuro)

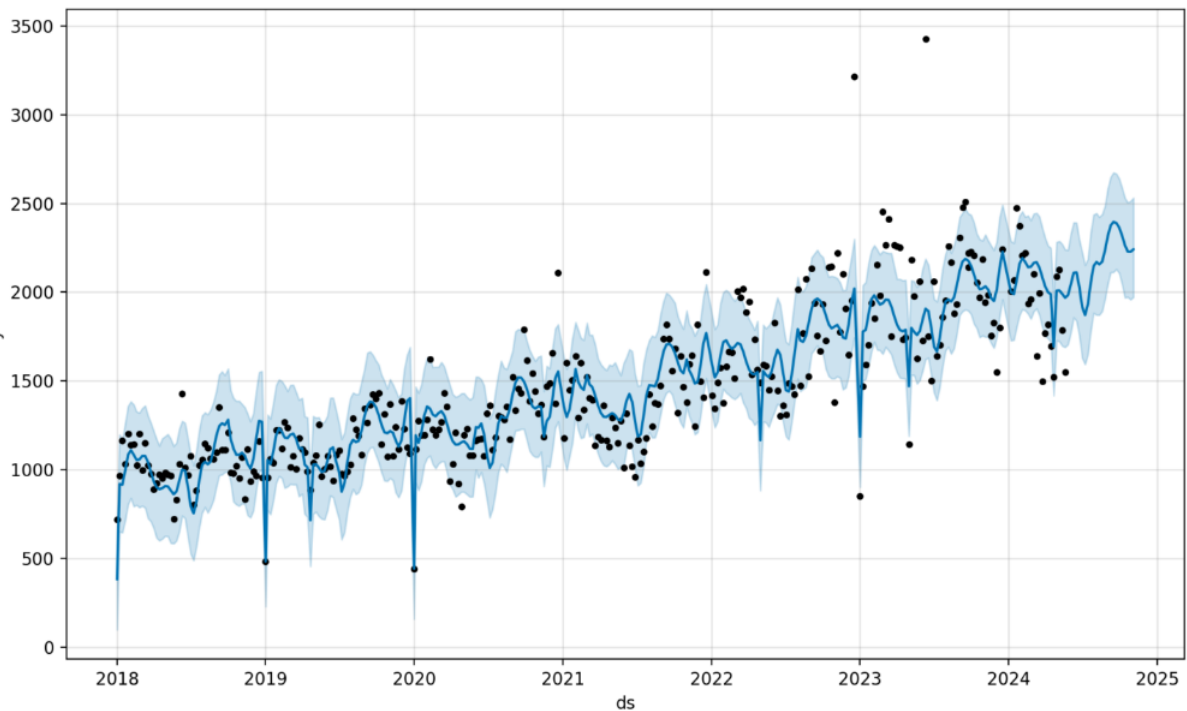


Figura 4: Previsão utilizando o modelo Prophet.

Também é possível mostrar os componentes dessa previsão como a tendência geral, tendência anual e a influência dos feriados respectivamente na Figura 5:

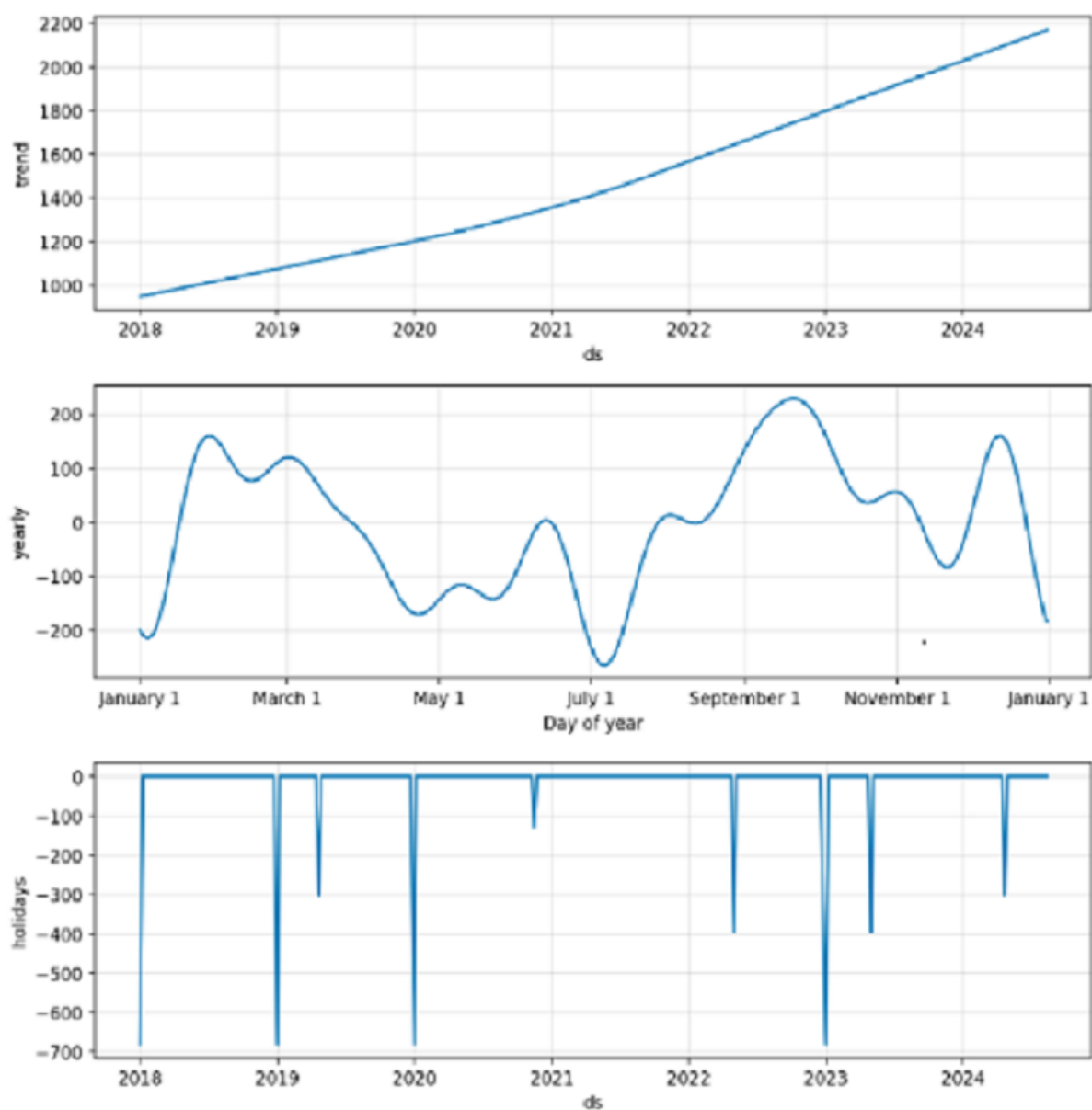


Figura 5: Componentes do modelo Prophet.

3.5 Modelo NeuralProphet

Nesse modelo da Figura 6 é possível perceber uma semelhança entre o Prophet, mas esse não possui uma previsão mínima e máxima como o anterior, os componentes também podem ser apresentados sendo idênticos ao da Figura 5.

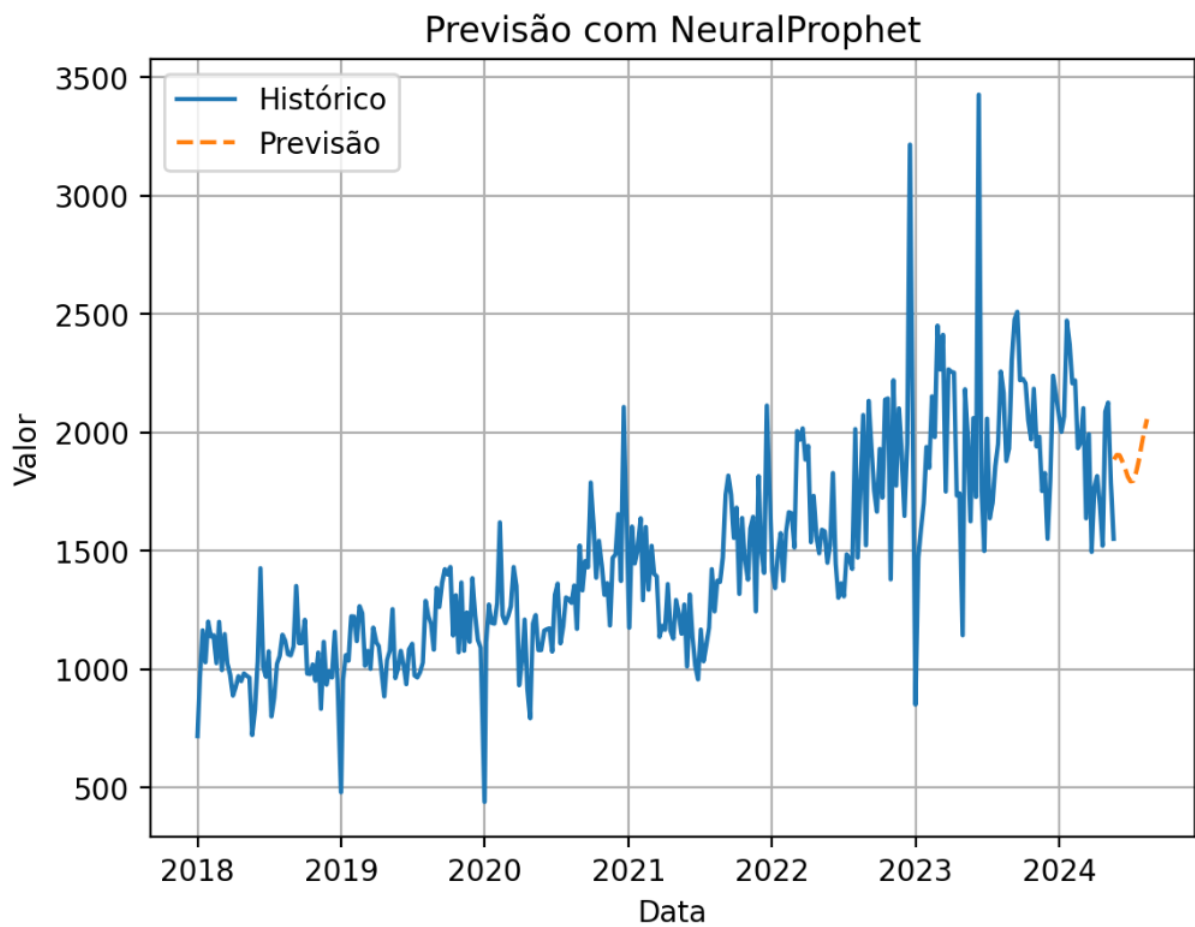


Figura 6: Previsão utilizando o modelo NeuralProphet.

3.6 Modelo ARIMA/SARIMA

Esse modelo da Figura 7 visualmente é muito parecido com o NeuralProphet, mas a forma do ARIMA de estimar é bem diferente pois não captura as sazonalidades e o SARIMA que não segue tão bem quanto o Prophet e acaba também fazendo quase uma linha reta ao invés de curvas como o NeuralProphet também podendo ser representado pela Figura 7.

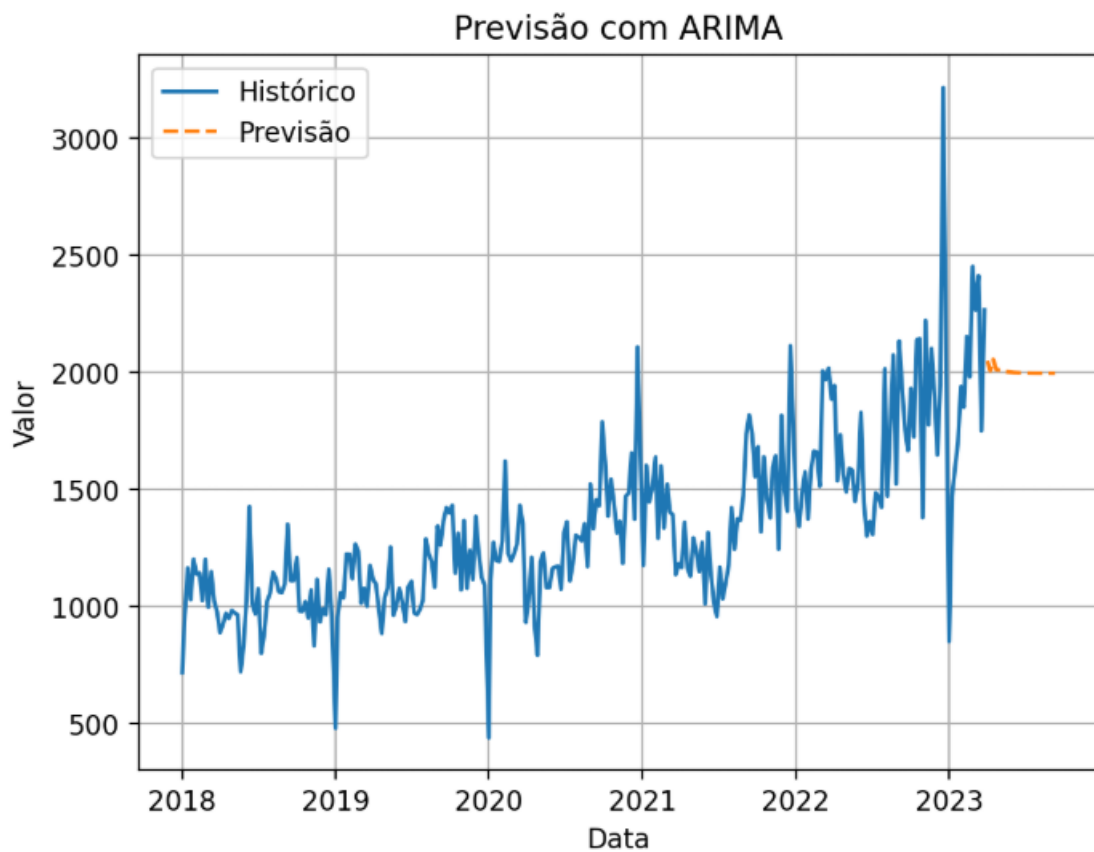


Figura 7: Previsão utilizando o modelo ARIMA.

3.7 Resultados primeiro produto

Utilizando $IQR=1.5$ e média móvel com 2 semanas em todos os modelos (utilizando o `auto_arima` para os outros parâmetros do ARIMA e SARIMA) foram obtidas as seguintes previsões na Figura 8, o valor real sem média móvel se refere ao programa ter feito a previsão em cima da média móvel mas está sendo comparado com o valor inadulterado de vendas:

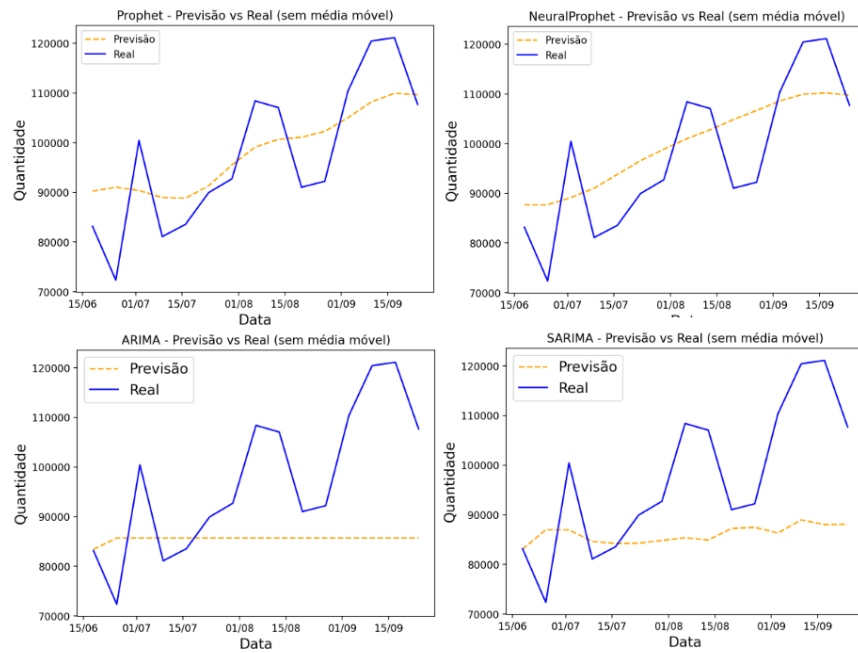


Figura 8: Previsões comparadas dos modelos — Produto 1.

No primeiro produto o Prophet obteve um MAPE de 8,43%, o NeuralProphet obteve 9,02%, o ARIMA obteve 15,32% e o SARIMA obteve 14,53% com as previsões da Figura 8.

E para cada um dos modelos foi criado um histograma demonstrando a frequência de erros, o que permitem avaliar de forma visual como os erros dos modelos estão distribuídos, oferecendo informações importantes sobre o comportamento da previsão. Por meio desse tipo de gráfico, é possível identificar se os erros se concentram próximos de zero — indicando previsões mais precisas — ou se há uma dispersão maior, sugerindo instabilidade no modelo.

Além disso, o histograma evidencia se há tendência sistemática de superestimação ou subestimação, observada pela assimetria da distribuição dos erros. Distribuições mais simétricas indicam que o modelo erra para mais e para menos com a mesma frequência, enquanto distribuições inclinadas revelam um padrão consistente de erro. A presença de barras muito afastadas do centro revela erros extremos, que podem indicar sensibilidade do modelo a picos ou quedas abruptas na série temporal. O primeiro produto obteve os seguintes resultados na Figura 9

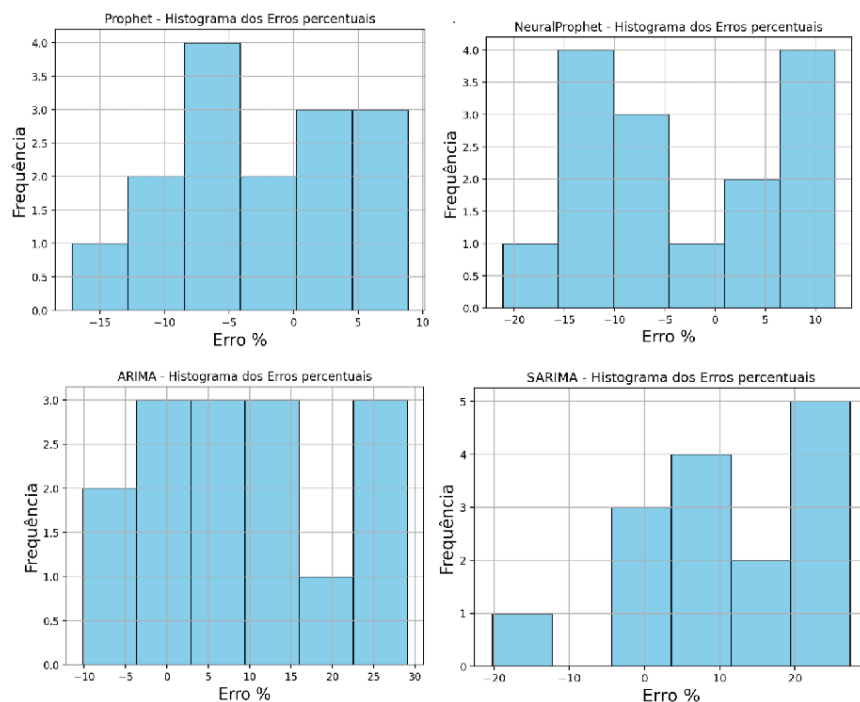


Figura 9: Histograma dos erros percentuais — Produto 1.

3.8 Resultados segundo produto

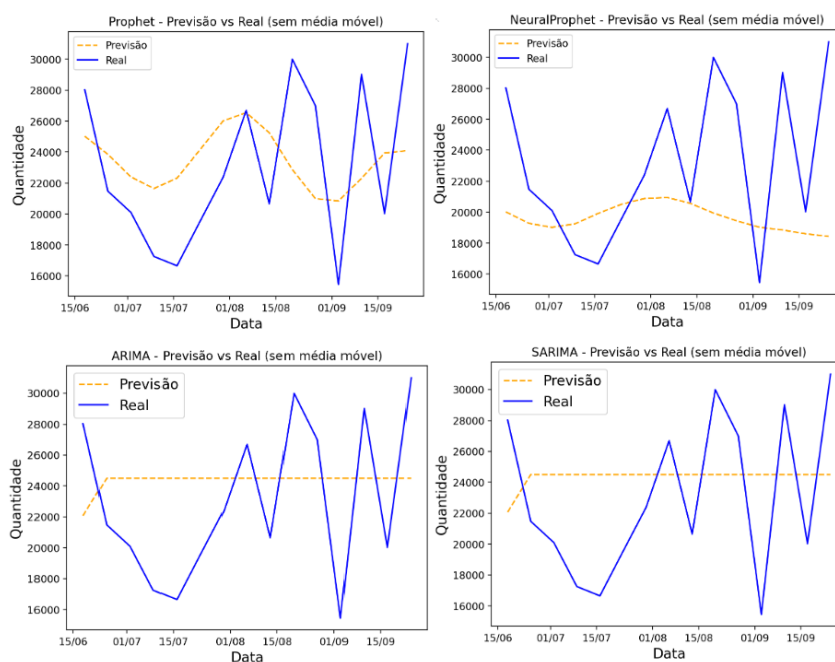


Figura 10: Previsões comparadas dos modelos — Produto 2.

Nesse produto as previsões da Figura 10 o Prophet obteve um MAPE de 20,26%, o NeuralProphet obteve 21,33%, já os modelos ARIMA e SARIMA obtiveram 22,53% resultando no

histograma da Figura 11:

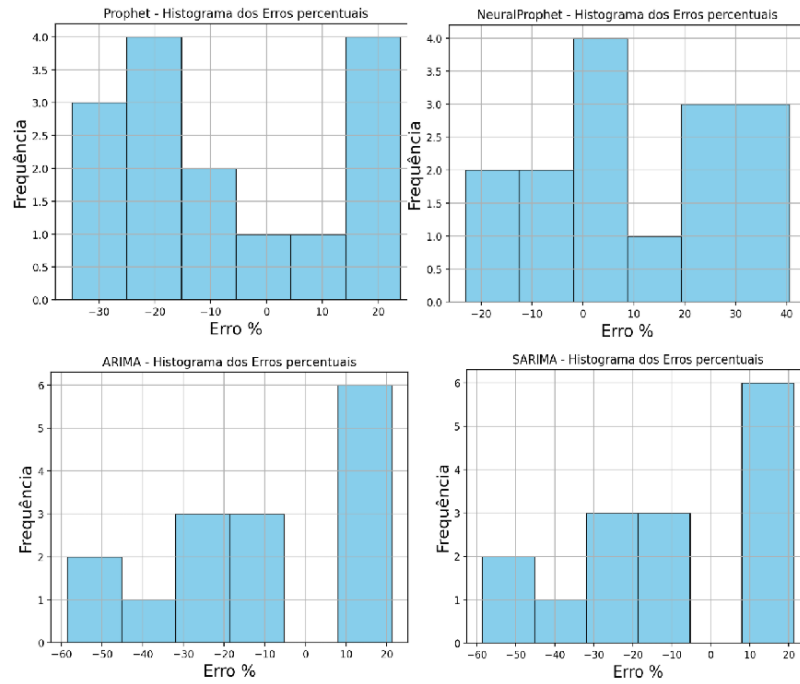


Figura 11: Histograma dos erros percentuais — Produto 2.

3.9 Resultados terceiro produto

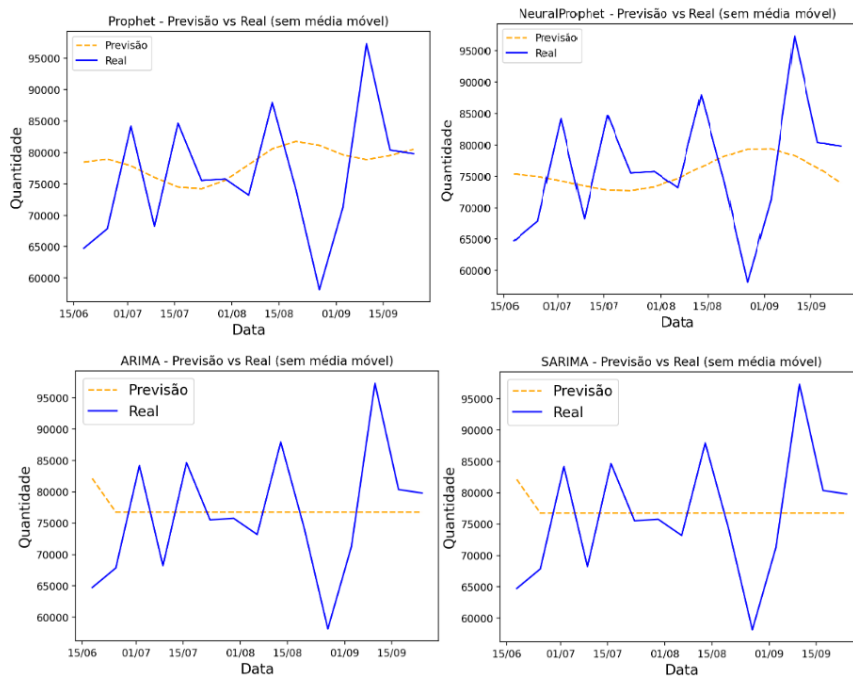


Figura 12: Previsões comparadas dos modelos — Produto 3.

No terceiro produto as previsões da Figura 12 o Prophet obteve um MAPE de 10,17%, o NeuralProphet obteve 11,12%, já os modelos ARIMA e SARIMA obtiveram 10,76% resultando nos erros percentuais da Figura 13:

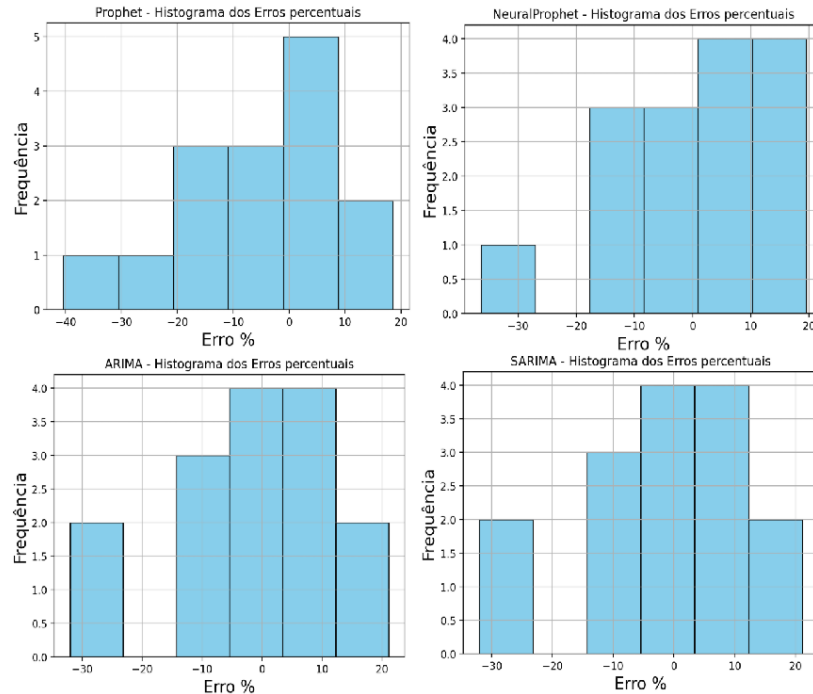


Figura 13: Histograma dos erros percentuais — Produto 3.

4 Discussão e Conclusão

A variação do MAPE dos modelos ficou entre 8% e 22%, indicando que os modelos já podem ser utilizados com alguns produtos, mas ainda pode precisar de aprimoramentos para obter uma previsão suficiente para aplicação comercial em produtos com menor consistência.

O Prophet foi o modelo com melhores resultados em todos os produtos, além de ter obtido um tempo de execução muito superior ao NeuralProphet e SARIMA.

O neuralProphet chegou próximo da precisão do Prophet mas seu componente de redes neurais fere o tempo de execução, possivelmente não sendo viável caso os dados históricos cheguem a milhares de valores.

ARIMA obteve um resultado levemente inferior aos outros, mas é bem eficiente ao ser utilizado com bases de dados grandes pelo seu tempo de execução.

O SARIMA obteve tecnicamente uma melhora em precisão quanto ao ARIMA, porém o tempo que leva para encontrar os parâmetros corretos pelo `auto.arima` é extremamente alto pela quantidade de combinações, e em 2 dos produtos ainda acabou encontrando os mesmos

parâmetros do ARIMA zerando os parâmetros de sazonalidade. Ele pode ser uma melhoria quanto ao outro modelo mas a utilização do `auto_arima` fica a questão do tamanho da base de dados utilizado.

Os resultados obtidos do Prophet e NeuralProphet nesses testes foram um pouco divergentes aos mostrados no artigo (TRIEBE et al., 2021), mostraram uma diferença mínima entre ambos os modelos mas nesse caso favorável ao NeuralProphet. Quanto ao tempo de execução do NeuralProphet mostraram ser 5 vezes maior o que não é um problema com a quantidade de dados utilizada onde ainda está em uma questão de minutos, mas essa diferença cresce exponencialmente com o tamanho da entrada o tornando inviável em certas situações.

Entre as opções de melhorias na precisão dos modelos que poderiam ser implementadas, utilizar-se de uma junção do histórico de vendas com as datas de descontos ou promoções no respectivo produto, isso poderia ajudar a prever a demanda e a entender a influência dos descontos nas vendas dando uma maior peso a quantidade de vendas em promoções anteriores. Uma outra forma de previsão que poderia ser utilizada e analisada são as redes neurais pois poderiam apresentar-se bem competitivas com o Prophet e Arima.

Referências

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015.

DEVELOPERS, S. learn. *Regression metrics*. 2024. Accessed: 2024-12-05. Disponível em: https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics.

FILHO, J. d. D. H. *Métodos quantitativos de previsão de vendas: estudo de caso em uma indústria Cearense de alimentos*. Fortaleza, CE, Brasil, 2018. Orientador: Prof. Dr. Cláudio Bezerra Leopoldino. Disponível em: https://repositorio.ufc.br/bitstream/riufc/37152/1/2018_tcc_jdholandafilho.pdf.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: Principles and Practice*. 2nd. ed. OTexts, 2018. Disponível em: <https://otexts.com/fpp3/>.

MCKINNEY, W.; TEAM, P. Pandas-powerful python data analysis toolkit. *Pandas—Powerful Python Data Analysis Toolkit*, v. 1625, 2015.

MILKPOINT. *Brasileiros preferem marcas locais de laticínios, aponta Nielsen*. 2024. Accessed: 2024-12-05. Disponível em: <https://www.milkpoint.com.br/noticias-e-mercado/giro-noticias/nielsen-brasileiros-preferem-marcas-locais-de-laticinios-108742n.aspx>.

RAFFERTY, G. *Forecasting Time Series Data with Facebook Prophet: Build, improve, and optimize time series forecasting models using the advanced forecasting tool*. [S.l.]: Packt Publishing Ltd, 2021.

TRIEBE, O. et al. *NeuralProphet: Explainable Forecasting at Scale*. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2111.15397>.