

PREVISÃO DO ÍNDICE DE VENDAS DO SETOR VAREJISTA BRASILEIRO UTILIZANDO DOS MÉTODOS HOLT-WINTERS E ARIMA: UM ESTUDO COM PYTHON

Renzo Paranaíba Mesquita (INATEL, UNIFEI)
renzokuken31@gmail.com

Frâneli Eduarda Rezende Costa (UNIFEI)
franelirezende@gmail.com

Pedro Paulo Balestrassi (UNIFEI)
pedro@unifei.edu.br

Fabiano Leal (UNIFEI)
fleal@unifei.edu.br



O varejo é o setor responsável pelo fornecimento de produtos e serviços ao consumidor final e tem sido protagonista no impulso do PIB brasileiro. Entender o comportamento do índice de vendas deste setor no Brasil pode auxiliar instituições a compreenderem melhor como ele vem se comportando ao longo dos anos. Além disso, por meio de dados passados e da aplicação de técnicas adequadas, é possível até mesmo buscar prever o comportamento do setor em curto ou médio prazo, a fim de auxiliar as instituições na tomada de decisões. Este trabalho tem como objetivo avaliar, por meio do uso da linguagem Python, a aplicação de dois métodos de previsão poderosos da literatura, no caso, Holt-Winters e ARIMA, para prever o índice do volume de vendas no setor de varejo brasileiro entre outubro de 2022 a setembro de 2023. Os dados foram coletados da base de dados do Banco Central do Brasil de janeiro de 2000 a setembro de 2022. Os resultados ilustram que o método ARIMA atinge melhor acurácia na previsão de acordo com o erro percentual absoluto médio (MAPE), por isso, ele foi escolhido para fazer a previsão no intervalo mencionado.

Palavras-chave: Varejo, Previsão, Holt-Winters, ARIMA, Python

1. Introdução

O varejo é um modelo de negócio baseado na venda de bens e serviços diretamente ao consumidor final, para uso pessoal e não comercial, independente se são vendidos pessoalmente, pela *Internet*, por *vending machines*, etc (KOTLER, P.; KELLER, K. L., 2012). Segundo dados da Sociedade Brasileira de Varejo e Consumo (SBVC), o valor agregado do varejo brasileiro corresponde a aproximadamente 11.04% do PIB do país (SBVC, 2021). Isso ilustra a importância deste setor no Brasil, e por isso, é importante que iniciativas públicas e privadas acompanhem seu comportamento com frequência para que possam melhor se moldarem aos desafios iminentes do setor.

Atualmente, a análise de dados por séries temporais tem sido largamente empregada para compreender e até mesmo buscar prever o comportamento de fenômenos em diferentes áreas. Uma série temporal é um conjunto de dados em que as observações são registradas na ordem que elas ocorrem ao longo do tempo. Em um gráfico deste tipo de série, o eixo vertical (y) denota um valor observado de uma variável x, que se encontra no eixo horizontal e possui um valor temporal correspondente, que pode ser em minutos, dias, meses, anos etc. A simples plotagem de uma série temporal já pode apresentar informações importantes para um determinado fenômeno, como ser capaz de ilustrar tendências, sazonalidades e outros tipos de características que dificilmente poderiam ser identificadas sem o auxílio de uma representação visual (MONTGOMERY, D. C.; RUNGER, G. C., 2013).

O Banco Central do Brasil (BCB) possui um Sistema Gerenciador de Séries Temporais (SGS) que tem como objetivo tornar disponíveis dados econômico-financeiros não só do Brasil, como também de cunho internacional. São mais de 18000 séries temporais que podem ser consultadas e baixadas gratuitamente de forma individual, em grupos ou em listas personalizadas por assunto (BCB, 2022). Para o setor de varejo, em particular, o SGS permite a consulta e utilização de dados relacionados ao volume de vendas por subsetor (Ex: Combustível e lubrificantes; hipermercados e supermercados; têxteis, vestuário e calçados etc.) ou de forma total, que podem servir de insumo para a realização de análises e previsões importantes para o setor.

Baseado nos dados fornecidos pelo BCB e fazendo uso da linguagem Python, este trabalho tem como objetivo avaliar a aplicação de dois modelos de previsão de séries temporais para melhor compreender de forma total o comportamento do setor varejista brasileiro entre outubro de 2022 e setembro de 2023. Os modelos utilizados foram: o método de alisamento exponencial Holt-Winters e o Box-Jenkins ARIMA, dois métodos objetivos, porém poderosos, e que têm

contribuído para a previsão de séries temporais em diferentes áreas. Além disso, o trabalho também visa ilustrar como a linguagem Python pode ser estratégica para se trabalhar com métodos de previsões, uma vez que fornece um conjunto poderoso de bibliotecas para este fim, com grande quantidade de recursos, mas com comandos objetivos.

Nas próximas seções este trabalho está organizado da seguinte forma: a seção 2 apresenta um referencial teórico, que busca ilustrar recursos importantes da linguagem Python para previsão de séries temporais, assim como a apresentação dos princípios fundamentais de funcionamento dos dois modelos a serem utilizados. A seção 3 apresenta qual foi o método de pesquisa utilizado para se conduzir este trabalho. Na seção 4 são apresentados detalhes a respeito da definição do problema, coleta e análise de dados. Na seção 5 são descritos os procedimentos para a seleção, validação e execução do modelo, e por fim, na seção 6, são apresentadas as conclusões dos autores e sugestões de trabalhos futuros.

2. Referencial teórico

Nos subtópicos a seguir serão apresentados recursos importantes da linguagem Python para previsão de séries temporais e conceitos importantes de dois modelos de previsão a serem utilizados neste trabalho, no caso, o método de alisamento exponencial Holt-Winters e o Box-Jenkins ARIMA, respectivamente.

2.1. A linguagem Python

O Python é uma linguagem de programação de propósito geral, diferente de outras linguagens de uso mais específico como o R e o Matlab. Isso significa que ela possibilita realizar pesquisas e o desenvolvimento de sistemas de *softwares* complexos em um mesmo lugar. De acordo com o índice TIOBE (2022), atualmente se destaca como a linguagem de programação mais utilizada no planeta.

Na área de ciência de dados, em particular, se destaca como a linguagem protagonista pela grande quantidade de bibliotecas disponíveis (BROWNLEE, J., 2017). Dentre as inúmeras, pode-se citar as seguintes utilizadas neste trabalho:

- NumPy: é o pacote fundamental para computação científica em Python. Ele oferece uma estrutura de dados chamada NumPy Array (ndarray) que permite o armazenamento de dados homogêneos e que podem ser estruturados em diferentes dimensões (como arrays unidimensionais, matrizes etc.). Além disso, oferece uma grande quantidade de funções matemáticas, lógicas e estatísticas pré-prontas para serem utilizadas sobre esta estrutura (NUMPY, 2022);

- Pandas: biblioteca que utiliza o NumPy como base e oferece ferramentas de alta performance para carregamento, manipulação e plotagem de dados. Ela fornece duas estruturas de dados fundamentais, mas poderosas, para lidar com dados em uma e duas dimensões, no caso, as *Series* e os *DataFrames*, respectivamente. Além disso, fornece recursos especiais para se trabalhar com séries temporais, como permitir a manipulação de índices temporais, realizar com facilidade processos de *lagging* nos dados, entre outros (PANDAS, 2022);
- Statsmodels: biblioteca que fornece um pacote completo para realização de testes e exploração de dados estatísticos, além de ferramentas dedicadas a análise de séries temporais que também podem ser usadas para previsões. Como exemplo, permite a aplicação e avaliação de vários modelos lineares como autorregressão (AR), média móvel (MA), métodos de suavização como Holt, Holt-Winters, entre outros (STATSMODELS, 2022);
- Pmdarima: biblioteca que usa o Statsmodels como base. Além de possuir todos os recursos fundamentais para se explorar séries temporais no Python, destaca-se a presença da função *auto_arima* (muito popular na linguagem R), que é capaz de retornar o melhor modelo ARIMA a ser utilizado nos dados de acordo com valores de *Akaike information criterion* (AIC) ou *Bayesian information criterion* (BIC). A função realiza uma pesquisa automatizada sobre os melhores parâmetros a serem utilizados dentro das restrições fornecidas (PMDARIMA, 2022).

2.2. O método de Holt-Winters

Este método, também chamado de método de Winters ou suavização exponencial sazonal, é uma extensão sofisticada da metodologia de suavização exponencial, uma vez que ele é capaz de lidar com dados que apresentam tendência e sazonalidade. Em teoria, a tendência é o aumento ou diminuição de valores na série, que podem acontecer de forma linear, exponencial ou amortecida. Já a sazonalidade é representada por um comportamento que se repete ao longo da série a cada x períodos (ŞAHINLI, 2020).

Para prever dados de séries temporais que possuam tanto tendência quanto sazonalidade, ele considera α , γ e δ como os três parâmetros de suavização (com valores entre 0 e 1) e p como o número de observações por ciclo sazonal. É parecido com o método tradicional de Holt (método mais simplificado de suavização), porém, com uma equação adicional para lidar com a sazonalidade dos dados (DA VEIGA et al., 2014).

Existem dois métodos diferentes de Holt-Winters, o método Aditivo e o Multiplicativo, que são diferenciados sobre como a sazonalidade é tratada. O método Aditivo é melhor para dados com tendência e sazonalidade que não aumentam com o tempo. Já o método multiplicativo é melhor para dados com tendência e sazonalidade que aumentam com o tempo (ŞAHINLI, 2020).

Maiores detalhes de operação e equações deste método podem ser encontrados em Montgomery et al. (2011).

2.3. O método ARIMA

A abreviação ARIMA significa *Autoregressive (AR) Integrated (I) Moving Average (MA)*. Também conhecido como modelo de Box-Jenkins fazendo uma alusão aos seus criadores, este método busca utilizar de dados do passado para prever o futuro por meio da integração de duas técnicas: autocorrelação e médias móveis (SILVA et al., 2021).

Um modelo ARIMA tradicional ou não sazonal é classificado como um modelo ARIMA (p,d,q) , em que p significa a ordem do modelo autorregressivo, d o grau de diferenciação para deixar a série estacionária (característica fundamental para se aplicar um modelo ARIMA) e q é a ordem do modelo de média móvel (ŞAHINLI, 2020).

Modelos ARIMA também são capazes de modelar séries que apresentam um componente sazonal. Neste caso, temos uma variação chamada de *Seasonal Autoregressive Integrated Moving Average (SARIMA)* ou ARIMA $(p, d, q) \times (P, D, Q)S$ em que o segundo componente faz referência à parte sazonal e S significa a duração de um ciclo sazonal (JAIN et al., 2021).

Os modelos ARIMA e SARIMA podem ser aplicados apenas em séries temporais univariadas. Porém, muitos problemas podem apresentar algum tipo de relacionamento entre variáveis. Neste caso, deve-se usar uma variação do ARIMA mais adequada para esta situação, no caso, o modelo *Seasonal Auto-Regressive Integrated Moving Average with exogenous factors (SARIMAX)*. Em suma, este modelo pode levar em consideração variáveis exógenas, ou seja, usar dados externos na previsão (JAIN, 2021).

Como mencionado, na linguagem Python, a escolha dos melhores parâmetros e a melhor variação do modelo ARIMA a ser utilizado pode ser feito por meio do uso da função `auto_arima`, disponível no pacote `pmdarima`. De acordo com Yermal (2017), o uso desta função poupa o desenvolvedor de executar e analisar manualmente testes implícitos à execução do modelo ARIMA, como a execução de funções de autocorrelação (ACF), autocorrelação parcial (PACF), testes de estacionariedade como o *Augmented Dickey Fuller (ADF)*, etc.

Maiores detalhes de operação e equações do método ARIMA e suas variações podem ser encontradas em Montgomery et al. (2011).

3. Método de pesquisa

A fim de melhor definir as etapas de desenvolvimento deste trabalho, os autores se inspiraram na sequência de passos que um processo de previsão deve seguir segundo Montgomery et al. (2011). Em suma, ele apresenta um processo para condução de projetos de previsão de séries temporais que segue uma sequência de 7 etapas, são elas:

- Definição do problema: busca compreender como a previsão poderá ser útil para atender às expectativas de possíveis clientes;
- Coleta de dados: consiste em obter dados históricos relevantes da(s) variável(is) que se pretende fazer previsões;
- Análise de Dados: A plotagem da série temporal dos dados coletados deve ser feita para se realizar inspeções visuais e começar a compreender possíveis padrões destes dados como possíveis tendências, sazonalidades, entre outros. Esta etapa é de grande importância para que se possa fazer boas decisões nas etapas seguintes;
- Seleção do Modelo e *Fitting*: consiste em escolher um ou mais modelos de previsão e seus parâmetros a serem utilizados se baseando no padrão dos dados observados na etapa de Análise de Dados;
- Validação do Modelo: consiste na avaliação dos modelos de previsão escolhidos para avaliar quão viável eles são para se realizar o processo de previsão. Cálculos dos erros de previsão podem ser utilizados nesta etapa;
- Execução do modelo de previsão: realizar o processo de previsão de dados futuros;
- Monitoramento da performance do modelo de previsão: deve ser uma atividade contínua após a implantação do modelo para garantir que ele ainda possa operar de forma satisfatória.

4. Definição do problema, Coleta e Análise de Dados

O varejo tem grande importância no panorama empresarial brasileiro. Para se fazer investimentos e empreender neste ramo, é importante que iniciativas tanto públicas quanto privadas acompanhem o comportamento dele, para que possam melhor se moldarem aos desafios do mercado, traçando planos de curto e médio prazo que os permitam melhor se adequarem à realidade do setor.

Os modelos de previsão de séries temporais podem ser grandes aliados neste processo, por isso, este trabalho busca avaliar a aplicação de dois modelos que possam ajudar a entender o

comportamento do setor entre os meses de outubro de 2022 e setembro de 2023. Para isso, dados temporais do índice de volume de vendas no setor varejista entre janeiro de 2000 e setembro de 2022 foram extraídos do SGS do BCB.

A Figura 1 ilustra o código utilizado para se iniciar a análise de dados. Ao se carregar a base de dados, os valores dos índices de vendas precisaram inicialmente ser transformados em números de ponto flutuante (*float*), pois naturalmente vieram em formato do tipo texto. Em seguida, foi necessário deixar claro que os dados coletados são de caráter mensal, por isso, a função *asfreq()* com o parâmetro “MS” foram utilizados. Por fim, a série temporal foi traçada para se começar a observar suas primeiras características.

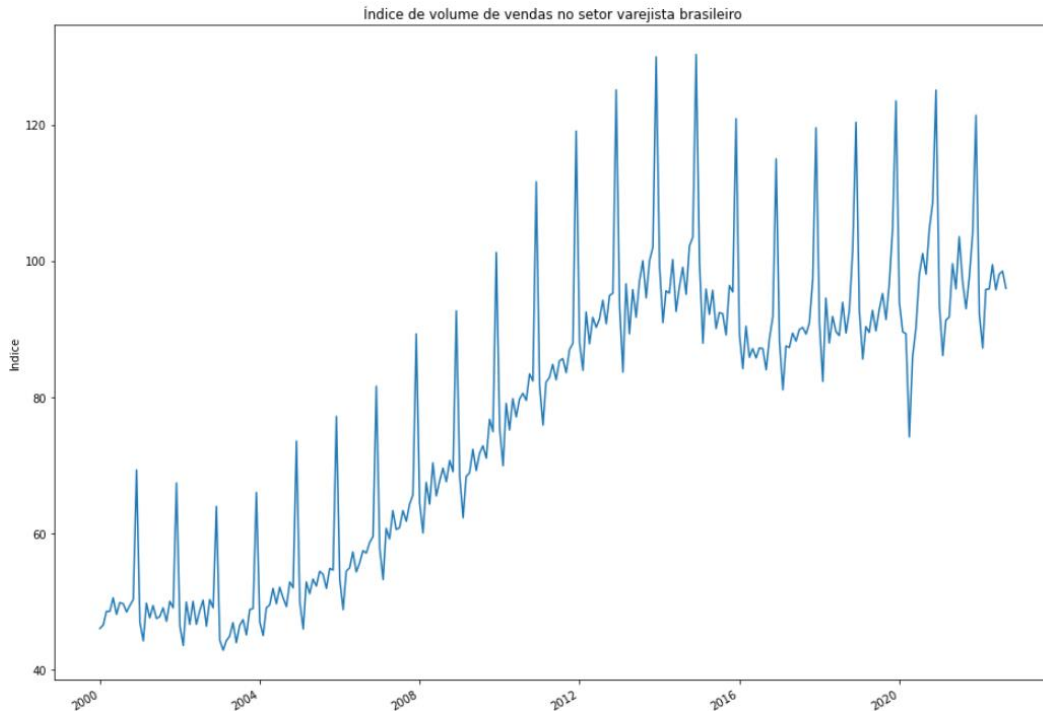
Figura 1 – Importando e Carregando a Base de Dados, Transformando os Dados, Determinando a Frequência da série e Plotando o Gráfico

```
1 # Importando Bibliotecas
2 import pandas as pd
3 import numpy as np
4
5 # Carregando o dataset
6 dataset = pd.read_csv('salesRetailIndexBr.csv', delimiter=';',
7                       index_col='date', parse_dates=True)
8
9 # Transformando os dados de y (sales_index) em float
10 dataset['sales_index'].astype(float)
11
12 # Convertendo a série para a frequência especificada
13 dataset = dataset.asfreq('MS')
14
15 # Plotando o gráfico
16 dataset['sales_index'].plot(figsize = (16, 12),
17                             title='Índice de volume de vendas no setor varejista brasileiro',
18                             xlabel = 'Data', ylabel = 'Índice', x_compat=True)
```

Fonte: o próprio autor.

A Figura 2 ilustra a série temporal dos dados extraídos do BCB. É possível observar que até o ano de 2015 a série possui uma tendência positiva, e nos anos seguintes, uma leve queda, seguida de um comportamento estacionário. Em todos os anos, é possível observar claramente a sazonalidade, com o volume de vendas aumentando de forma intensa nos finais de ano.

Figura 2 – Índice de volume de vendas no setor varejista brasileiro



Fonte: o próprio autor.

5. Seleção, Validação e Execução do Modelo

Os comportamentos da série temporal traçada possuem sinergia com os modelos que serão utilizados neste trabalho. Segundo Da Veiga et al. (2014), o Holt-Winters é adequado para ser utilizado em séries que apresentam tendência e sazonalidade. Já o ARIMA também possui variações capazes de lidar com séries que possuam este tipo de comportamento (JAIN et al., 2021).

No total, a base de dados utilizada possui 273 meses (de janeiro de 2000 a setembro de 2022). A fim de validar os modelos escolhidos, 218 meses foram utilizados para treinamento (~80% dos dados) e 55 meses (~20% dos dados) para testes. Esta configuração na linguagem Python pode ser observada na Figura 3.

Figura 3 – Organização dos dados para treino e teste dos modelos

```
1 # No total o Dataset possui 273 meses (De jan 2000 a set 2022)
2 # 218 são usados para treinamento e 55 para testes
3 test_months = 55
4 training_set = dataset.iloc[:-test_months, :]
5 test_set = dataset.iloc[-test_months:, :]
```

Fonte: o próprio autor.

Após a organização dos dados para treino e teste dos modelos, inicialmente o modelo Holt-Winters foi executado utilizando das configurações apresentadas na Figura 4. Como os dados

no geral possuem tendência que aumenta com o tempo e sazonalidade (de período = 12), o método multiplicativo deste modelo foi escolhido para ser utilizado.

Figura 4 – Aplicando o Modelo Holt-Winters

```

1 # Importando Bibliotecas
2 from statsmodels.tsa.holtwinters import ExponentialSmoothing
3
4 # Aplicando o Modelo com os dados de treinamento
5 model = ExponentialSmoothing(endog=training_set.sales_index,
6                               trend='mul', seasonal='mul',
7                               seasonal_periods=12).fit()
8
9 # Realizando as Previsões do mesmo tamanho dos dados de teste
10 forecasting_hw = model.forecast(steps = len(test_set))
11
12 # Mostrando Dados de Treinamento
13 training_set['sales_index']['2015-01-01:'].plot(figsize=(16,12), legend = True,
14                                                xlabel = 'Data', ylabel = 'Índice',
15                                                label = 'Dados de Treinamento')
16
17 # Mostrando Dados de Teste
18 test_set['sales_index'][:].plot(legend = True, label = 'Dados de Teste')
19
20 # Mostrando Previsões
21 forecasting_hw.plot(legend = True, label = 'Previsão com Holt-Winters')

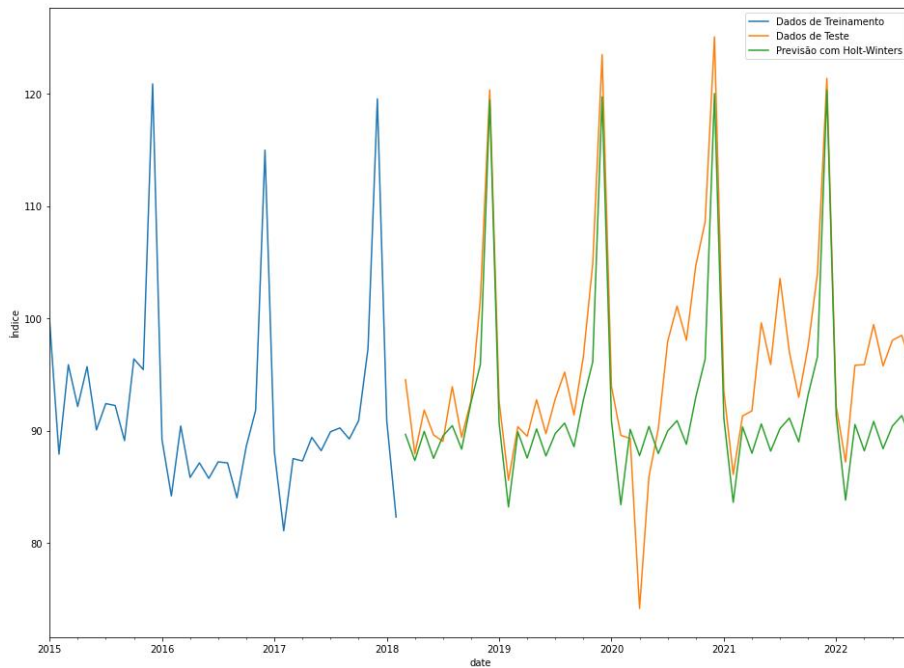
```

Fonte: o próprio autor.

A fim de deixar a visualização dos dados de teste e das previsões para este período mais explícitos, a plotagem da série foi limitada apenas ao ano de 2015 adiante, conforme ilustra a Figura 5. Apesar das previsões em relação aos dados de teste não terem ficado precisas, os resultados ilustram que o modelo respeitou principalmente a sazonalidade que vinha acontecendo nos anos anteriores. Vale ressaltar que as discrepâncias presentes nos anos de 2020 e 2021 são compreensíveis devido à pandemia de COVID-19, que acabou causando grandes impactos no setor de varejo.

Em seguida, a fim de identificar qual modelo ARIMA poderia ser o mais adequado para ser aplicado sobre os dados coletados, a função *auto_arima()* com parâmetro $m = 12$ foi executada, conforme ilustra a Figura 6. Caso a série possua sazonalidade, como é o caso da série deste trabalho, é importante que ela seja explicitada por meio do parâmetro m , pois isso pode auxiliar a função a melhor identificar o modelo adequado. Como pode ser visto ainda na Figura 6, após a execução da função *auto_arima()*, o método ARIMA (1, 1, 3)(0, 1, 2)₁₂ foi o melhor encontrado por ele. Como o resultado possui uma parte sazonal (*seasonal_order*), este método é equivalente à variação do ARIMA chamada SARIMA.

Figura 5 – Plotagem da Previsão do Modelo Holt-Winters em relação aos dados de teste



Fonte: o próprio autor.

Figura 6 – Aplicando a função `auto_arima()`

```

1 # Importando Bibliotecas
2 from pmdarima.arima import auto_arima
3
4 # Aplicando o Modelo com os dados de treinamento
5 model = auto_arima(y = training_set['sales_index'], m = 12)
6
7 # Mostrando resultado do melhor modelo ARIMA detectado
8 print(model)

ARIMA(order=(1, 1, 3), scoring_args={}, seasonal_order=(0, 1, 2, 12),
      suppress_warnings=True, with_intercept=False)

```

Fonte: o próprio autor.

Após a aplicação do `auto_arima()`, em seguida o método ARIMA detectado foi utilizado para se realizar as previsões sobre os dados de teste, conforme ilustra a Figura 7. As configurações de plotagem foram as mesmas utilizadas no método Holt-Winters a fim de deixar a visualização dos dados de teste e das previsões para este período mais explícitos.

A Figura 8 ilustra os resultados produzidos pelo método ARIMA. Como pode ser visto, o resultado parece similar ao produzido pelo modelo de Holt-Winters.

Figura 7 – Aplicando o Modelo ARIMA

```

1 # Realizando as Previsoes
2 forecasting_arima = pd.Series(model.predict(n_periods=len(test_set)))
3
4 # Setando os indices dos dados previstos igual aos indices dos dados de teste
5 forecasting_arima.index = test_set.index
6
7 # Mostrando Dados de Treinamento
8 training_set['sales_index']['2015-01-01:'].plot(figsize=(16,12), legend = True,
9                                     xlabel = 'Data', ylabel = 'Índice',
10                                    label = 'Dados de Treinamento')
11
12 # Mostrando Dados de Teste
13 test_set['sales_index'][:].plot(legend = True, label = 'Dados de Teste')
14
15 # Mostrando Previsões
16 forecasting_arima.plot(legend = True, label = 'Previsão com ARIMA')

```

Fonte: o próprio autor.

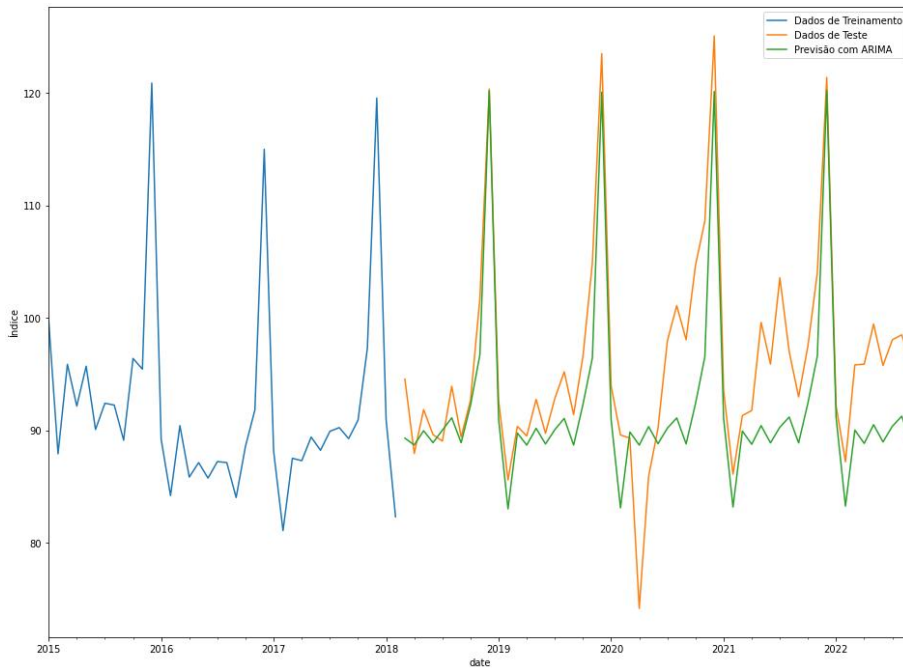
Porém, a fim de identificar aquele que poderia produzir melhor acurácia na previsão, o teste de acurácia chamado de Erro Percentual Absoluto Médio (MAPE, do inglês *Mean Absolute Percentage Error*), um dos critérios mais populares utilizados para auxiliar na seleção de modelos segundo Da Veiga et al. (2014), foi utilizado a fim de identificar aquele que possui o menor erro, conforme ilustra a Figura 9.

Após o cálculo do MAPE para ambos os modelos, o método ARIMA apresentou o menor valor. Por conta disso, será o método escolhido para finalmente prever dados que extrapolam o período coletado, no caso, compreender o comportamento do setor varejista brasileiro entre outubro de 2022 e setembro de 2023.

A Figura 10 ilustra o procedimento que foi utilizado para realizar a previsão com ARIMA para o período futuro mencionado. Neste momento, toda a base de dados foi utilizada para treinamento do modelo, e logo em seguida, foi definida a previsão de 12 meses adiante a partir de setembro de 2022 (fim da base coletada do BCB).

O resultado da previsão pode ser visto na Figura 11. As configurações de plotagem foram reajustadas para ilustrar do ano de 2019 adiante, a fim de deixar mais explícito a previsão do período futuro. Como pode ser visto, segundo as previsões, o cenário futuro parece promissor para o setor varejista, uma vez que a previsão mostrou a alta de vendas para o fim do ano de 2022 em relação ao ano de 2021, e até mesmo o intervalo entre janeiro a setembro de 2023 se mostrou estável, sem quedas bruscas.

Figura 8 – Plotagem da Previsão do Modelo ARIMA em relação aos dados de teste



Fonte: o próprio autor.

Figura 9 – Procedimento para cálculo dos MAPEs

```

1 # Função do MAPE
2 def MAPE(y_true, y_pred):
3     y_true, y_pred = np.array(y_true), np.array(y_pred)
4     return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
5
6 # Cálculo dos MAPEs
7 print('MAPE Holt-Winters:', MAPE(test_set['sales_index'], forecasting_hw))
8 print('MAPE ARIMA:', MAPE(test_set['sales_index'], forecasting_arima))

```

MAPE Holt-Winters: 4.960289376513722
 MAPE ARIMA: 4.872926621845155

Fonte: o próprio autor.

Figura 10 – Procedimento para realizar a previsão com ARIMA para o período futuro

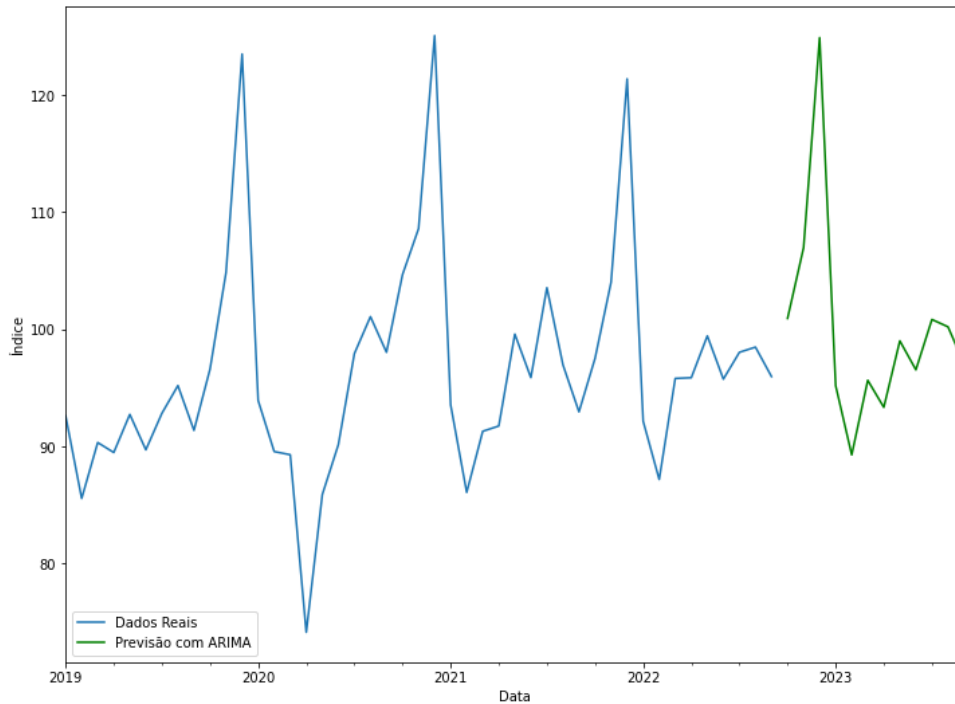
```

1 # Aplicando o Modelo com os dados completos
2 model = auto_arima(y = dataset['sales_index'], m = 12)
3
4 # Realizando as Previsões
5 forecasting_arima = pd.Series(model.predict(n_periods=12))
6
7 # Mostrando Dados
8 dataset['sales_index']['2019-01-01:'].plot(figsize=(12,9), legend = True,
9                                           label = 'Dados Reais')
10
11 # Mostrando Previsões
12 forecasting_arima.plot(legend = True, label = 'ARIMA Forecasting',
13                        color='green')

```

Fonte: o próprio autor.

Figura 11 – Previsão com ARIMA para o período futuro



Fonte: o próprio autor.

6. Conclusão

Conforme dados ilustrados neste estudo, o setor varejista possui alta relevância na economia brasileira. Compreender o comportamento deste setor de forma contínua pode ser um procedimento estratégico para estudiosos e investidores, tanto de iniciativas públicas quanto privadas, a fim de que tomem decisões sobre o setor baseadas em fatos, no caso, por meio de dados (*data-driven decisions*).

Este trabalho teve como objetivo ilustrar como dois métodos de previsão poderosos, no caso, o método de suavização exponencial Holt-Winters e o ARIMA, podem ser facilmente utilizados na linguagem Python para prever o comportamento do setor varejista. São inúmeros os recursos no Python que podem facilitar este processo, mas destacam-se as bibliotecas Statsmodels e Pmdarima, que possibilitam trabalhar mais precisamente com séries temporais.

Após a avaliação de qual modelo utilizar por meio de testes e cálculo do MAPE, o método ARIMA foi selecionado para se prever os dados futuros do setor varejista. Como pode ser visto, o setor se mantém promissor para o período de outubro de 2022 a setembro de 2023, sem apresentar comportamentos discrepantes.

Apesar deste trabalho ter produzido *insights* para prever o comportamento do setor varejista por meio de métodos de previsão poderosos, seu estudo ainda é inicial. Neste estudo foi utilizada uma série temporal univariada para se realizar o processo de previsão, porém, muitos

problemas podem utilizar de dados externos na previsão (variáveis exógenas) para se produzir *insights* ainda mais precisos. Por isso, sugere-se como trabalhos futuros investigar outras possíveis variáveis que poderiam amadurecer este processo de previsão.

Com este trabalho, também foi possível perceber como a linguagem Python pode ser uma grande aliada no estudo de séries temporais. A quantidade de bibliotecas, funções e parâmetros disponíveis para se aplicar e refinar os modelos são inúmeras, e neste trabalho, em particular, praticamente se utilizou dos parâmetros padrões para executá-los. Por isso, sugere-se também como trabalhos futuros, investigar de forma mais profunda as configurações avançadas que as bibliotecas podem oferecer a fim de obter resultados mais precisos.

REFERÊNCIAS

BCB. **Sistema Gerenciador de Series Temporais**. BCB, 2022. Disponível em:

<<https://www3.bcb.gov.br/sgspub/localizarseries/localizarSeries.do?method=prepararTelaLocalizarSeries>> Acesso em: 14 nov. 2022.

BROWNLEE, J. **Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future**. Machine Learning Mastery, 2017.

DA VEIGA, C.P., DA VEIGA, C.R.P., CATAPAN, A., TORTATO, U. and DA SILVA, W.V. **Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models**. WSEAS transactions on business and economics, 11(1), p. 608-614, 2014.

JAIN, A., SUKHDEVE, T., GADIA, H., SAHU, S. P., & VERMA, S. **COVID19 prediction using time series analysis**. International Conference on Artificial Intelligence and Smart Systems (ICAIS). p. 1599-1606, 2021.

KOTLER, P.; KELLER, K. L. **Administração de Marketing**. Pearson Education do Brasil, 2012.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied statistics and probability for engineers**. John Wiley & Sons, 2013.

MONTGOMERY, D.C.; JENNINGS, C.L.; KULAHCI, M. **Introduction to Time Series Analysis and Forecasting**. Wiley, 2011.

NUMPY. **NumPy official website**. NumPy, 2022. Disponível em: < <https://numpy.org/> > Acesso em: 15 nov. 2022.

PANDAS. **Pandas official website**. Pandas, 2022. Disponível em: <<https://pandas.pydata.org/>> Acesso em: 15 nov. 2022.

PMDARIMA. **Pmdarima official repository**. PyPi, 2022. Disponível em: <<https://pypi.org/project/pmdarima/>> Acesso em: 15 nov. 2022.

ŞAHINLI, M.A. **Potato Price Forecasting with Holt-Winters and ARIMA Methods: A Case Study**. Am. J. Potato Res. 97, p. 336–346, 2020.

SBVC. **O Papel do Varejo na Economia Brasileira**. SBVC, 2021. Disponível em: <http://sbvc.com.br/wp-content/uploads/2021/04/O-Papel-do-Varejo-na-Economia-Brasileira_2021-SBVC-4.pdf> Acesso em: 11 nov. 2022.

SILVA, A.B.D.S., ARAÚJO, A.C.D.M., FRIAS, P.G.D., VILELA, M.B.R. and BONFIM, C.V.D. **Modelo Autorregressivo Integrado de Médias Móveis (ARIMA): aspectos conceituais e metodológicos e sua aplicabilidade na mortalidade infantil**. Revista Brasileira de Saúde Materno Infantil, 21, p. 647-656, 2021.

STATSMODELS. **Statsmodels official website**. Statsmodels, 2022. Disponível em: <<https://www.statsmodels.org/stable/index.html>> Acesso em: 15 nov. 2022.

TIOBE. **TIOBE Index for November 2022**. TIOBE the software quality company, 2022. Disponível em: <<https://www.tiobe.com/tiobe-index/>> Acesso em: 14 nov. 2022.

YERMAL, L; BALASUBRAMANIAN, P. **Application of auto arima model for forecasting returns on minute wise amalgamated data in nse**. IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC), p. 1-5, 2017.