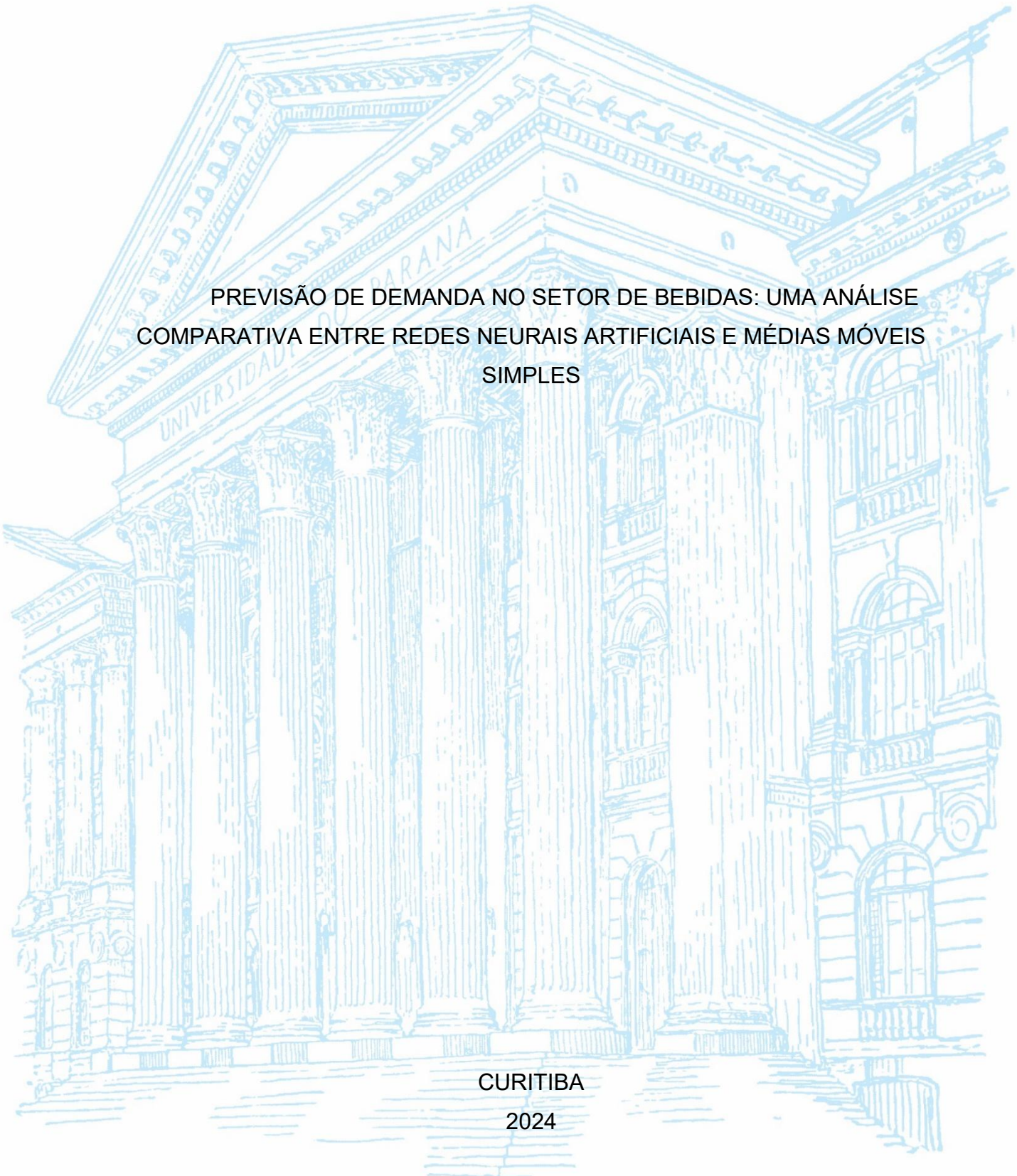
UNIVERSIDADE FEDERAL DO PARANÁ

GABRIELLE ANTUNES



PREVISÃO DE DEMANDA NO SETOR DE BEBIDAS: UMA ANÁLISE COMPARATIVA ENTRE REDES NEURAIS ARTIFICIAIS E MÉDIAS MÓVEIS SIMPLES

CURITIBA

2024

Gabrielle Antunes

PREVISÃO DE DEMANDA NO SETOR DE BEBIDAS: UMA ANÁLISE COMPARATIVA ENTRE REDES NEURAIS ARTIFICIAIS E MÉDIAS MÓVEIS SIMPLES

Trabalho de conclusão de curso apresentado ao curso de Graduação em Engenharia de Produção, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia de Produção.

Orientadora: Profa. Dra. Mariana Kleina

CURITIBA

2024

Dedico à minha família que esteve ao meu lado em cada etapa desta jornada acadêmica, seu apoio inabalável, incentivo constante e amor incondicional foram fundamentais para que eu chegasse até aqui, cada palavra deste trabalho é também um reflexo do seu carinho e suporte. À Mariana Kleina, que desde o início da graduação tem sido minha inspiração. Este trabalho é uma pequena homenagem à sua influência positiva em minha vida.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todas as pessoas que contribuíram de forma significativa para a conclusão deste trabalho. As palavras de incentivo, apoio emocional e colaboração foram essenciais para minha jornada acadêmica.

Em primeiro lugar, agradeço a Deus por ter me concedido saúde e sabedoria durante toda a minha graduação. Sua orientação e proteção foram fundamentais em cada passo que dei ao longo deste caminho.

À minha orientadora, Professora Mariana Kleina, minha mais sincera gratidão pelo seu apoio, orientação e expertise. Sua dedicação e comprometimento foram verdadeiramente inspiradores e contribuíram significativamente para meu crescimento acadêmico e profissional.

Aos meus queridos familiares e amigos, expresso minha imensa gratidão por estarem sempre presentes e por seu apoio incondicional ao longo desta jornada. Obrigada por estarem ao meu lado, por me ouvirem, me motivarem e por nunca deixarem que eu desistisse.

Ao Henrique Salmon, muito obrigada por seu amor, apoio incondicional e compreensão ao longo de toda essa jornada. Obrigada por estar sempre comigo, por me incentivar e trazer leveza e alegria em todos os momentos. Você é minha inspiração e, se cheguei até aqui, saiba que não teria conseguido sem você.

À minha amiga Laura Rodrigues, sua amizade foi um presente precioso durante todo esse caminho. Os momentos que compartilhamos serão lembrados com carinho para sempre, mais especificamente, aqueles que saímos tomar café para escrever este trabalho.

A todos vocês, que de alguma forma contribuíram para a conclusão deste trabalho, meu mais sincero obrigado.

RESUMO

A previsão de vendas é essencial para a tomada de decisões estratégicas e operacionais, permitindo antecipar desafios e otimizar recursos. Este estudo investiga e compara o desempenho de dois métodos preditivos na previsão de demanda no mercado de bebidas, utilizando os dados históricos de vendas de uma marca específica de bebidas, para previsão dos próximos dozes dias de vendas. A primeira metodologia empregada é a das Médias Móveis Simples, enquanto a segunda utiliza Redes Neurais Artificiais. O desempenho das previsões é avaliado por meio da métrica Raiz do Erro Quadrático Médio. Os resultados mostram que, embora as Redes Neurais Artificiais sejam métodos mais sofisticados, as Médias Móveis Simples tiveram melhor desempenho na série de dados, a qual apresenta períodos de estabilidade moderada. Este estudo destaca a importância de escolher o método preditivo adequado conforme a natureza dos dados e as condições do mercado. Suas conclusões fornecem *insights* valiosos para a indústria de bebidas, ajudando as empresas a tomar decisões mais assertivas e estratégicas.

Palavras-chave: métodos preditivos; tomada de decisão; mercado de bebidas; planejamento.

ABSTRACT

Sales forecasting is essential for strategic and operational decision-making, enabling the anticipation of challenges and the optimization of resources. This study investigates and compares the performance of two predictive methods in forecasting demand in the beverage market, using historical sales data from a specific beverage brand to predict the next twelve days of sales. The first methodology employed is Simple Moving Averages, while the second uses Artificial Neural Networks. Prediction performance is evaluated using the Root Mean Squared Error metric. The results show that, although Artificial Neural Networks are more sophisticated methods, Simple Moving Averages performed better on the data series, which exhibits periods of moderate stability. This study highlights the importance of choosing an appropriate predictive method based on the nature of the data and market conditions. Its conclusions provide valuable insights for the beverage industry, helping companies make more informed and strategic decisions.

Keywords: predictive methods; decision-making; beverage market; planning.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS OU SIGLAS

RNA	- Redes Neurais Artificiais
MLP	- *Multilayer Perceptron*
ABRABE	- Associação Brasileira de Bebidas
RMSE	- Raiz do erro quadrático médio

LISTA DE EQUAÇÕES

SUMÁRIO

1 INTRODUÇÃO

No contexto empresarial atual, a previsão de vendas é uma ferramenta essencial para a tomada de decisões estratégicas e operacionais. Um planejamento bem definido e alinhado com os colaboradores, é fundamental para o bom funcionamento de uma organização, porque é por meio desse planejamento que a empresa pode antecipar desafios e enfrentar eficazmente os problemas do dia a dia.

De acordo com Tubino (2007), a previsão de demanda é um dos pontos chave para o planejamento estratégico, pois permite que a empresa antecipe suas ações. Essa afirmação é reforçada por Gerber *et al.* (2013), que apontam a previsão de demanda como um ponto inicial do planejamento em empresas de bens de consumo, permitindo o planejamento adequado da produção.

Além disso, a previsão de demanda pode ser considerada uma vantagem competitiva, uma vez que possibilita a determinação dos recursos necessários para as operações (VEIGA e DUCLÓS, 2010), e auxilia a empresa a evitar perdas financeiras significativas devido ao excesso de estoque, vendas perdidas e ineficiências no controle da produção (MIRANDA *et al.,* 2011).

A indústria de bebidas, em particular, apresenta um cenário dinâmico e diversificado. Segundo a Associação Brasileira de Bebidas (ABRABE, 2014), o setor gera empregabilidade e possui relevância para o crescimento da economia brasileira. Após a pandemia de Covid-19, o mercado de bebidas alcoólicas tem mostrado crescimento, especialmente durante períodos de calor, o que torna o cenário ainda mais competitivo (KANTAR, 2024). Isso ressalta a importância de um bom planejamento para essas empresas.

O presente trabalho visa realizar a previsão de demanda de uma empresa de bebidas e responder à seguinte pergunta: Qual método, entre Médias Móveis Simples e Redes Neurais Artificiais, oferece previsões mais precisas e confiáveis para o volume de vendas de bebidas?

Para responder à pergunta proposta, serão consideradas as seguintes hipóteses:

I) As Redes Neurais Artificiais apresentarão uma exatidão de previsão superior em comparação com as Médias Móveis Simples devido à sua capacidade de capturar relações complexas entre variáveis;

II) As Médias Móveis Simples podem fornecer previsões mais estáveis e robustas em períodos de volatilidade moderada, enquanto as Redes Neurais Artificiais podem se destacar em períodos de mudanças bruscas ou não lineares nas tendências de vendas;

III) A eficácia relativa das Médias Móveis Simples e das Redes Neurais Artificiais na previsão de volume pode variar de acordo com a disponibilidade e qualidade dos dados, bem como com a natureza específica do mercado de bebidas.

Este estudo analisará os dados de vendas de uma empresa de bebidas, desde 2019 até 2023, cujo nome será mantido em sigilo e os dados modificados para preservar a confidencialidade da organização.

Ao final deste estudo, espera-se responder à pergunta proposta e verificar a validade das hipóteses, fornecendo à empresa conhecimento sobre esses métodos para apoiar decisões mais assertivas.

1.1 OBJETIVOS DO TRABALHO

1.1.1 Objetivo geral

O objetivo deste estudo é aplicar dois métodos quantitativos de previsão de demanda: Método das Médias Móveis Simples e Redes Neurais Artificias, em uma série temporal sobre demanda de bebidas, e verificar, a partir de uma medida de erro, qual modelo se adequou melhor para a previsão de demanda.

1.1.2 Objetivos específicos

Como objetivos específicos deste trabalho, destacam-se:

- Entender os conceitos sobre previsão de demanda, e os métodos a serem aplicados neste trabalho;
- Definir uma base histórica que possua dados diários de demanda de bebidas, para a utilização nessa pesquisa;
- Aplicar o Método da Médias Móveis Simples e o de Redes Neurais Artificiais, o *Multilayer Perceptron*, para fazer a previsão de demanda

de 12 dias à frente (dia a dia), utilizando o *software* R, que possui funções prontas para serem aplicadas;

- Definir o nível de significância, e calcular os intervalos de confiança em torno das previsões de cada método;
- Comparar o resultado dos dois modelos, a partir do cálculo da Raiz do Erro Quadrático Médio, e selecionar o que possui o menor erro de previsão.

2 REVISÃO DE LITERATURA

Esse capítulo irá abordar os conceitos de cada um dos modelos que serão aplicados, e como será feita a comparação dos resultados obtidos.

2.1 PREVISÃO DE DEMANDA

A previsão de demanda tem como objetivo identificar padrões de comportamento em séries históricas e antecipar seu comportamento futuro, além de estudar os fatores causais que influenciam esse comportamento (ACKERMANN e SELLITTO, 2022)

Arvan *et al*. (2019) ainda complementam que a previsão de demanda tem um papel muito importante em diversas áreas de uma organização, principalmente porque ela afeta a estratégia, produção e venda dos produtos da empresa.

A previsão de demanda pode objetivar dois horizontes de planejamento: visando o curto e médio prazo, ou visando o longo prazo (TUBINO, 2007):

a) Previsão de demanda em curto ou médio prazo: utilizada no planejamento mestre da produção. Mensurar os recursos disponíveis, o que está faltando, definições de armazenagem, compras, estoques e sequenciamento da produção, é o que é feito nesta fase;
b) Previsão de demanda em longo prazo: é utilizada no planejamento estratégico, pois define os produtos a serem fabricados, verifica as instalações da empresa, qualificação da mão-de-obra, entre outros.

Pode-se dividir os modelos de previsão de demanda em duas categorias: qualitativa e quantitativa, uma baseada no julgamento e outra mais matemática, respectivamente (TUBINO, 2007).

Segundo Armstrong (1983), os métodos qualitativos, ou como o autor chama, métodos subjetivos, são utilizados quando não há um processo bem definido para a análise de dados. Além disso, este método é muito utilizado quando as empresas estão lançando um novo produto, e não sabem como o consumidor irá se comportar. Com isso, as previsões são baseadas em opiniões dos especialistas da área, com o contexto envolvido (TUBINO, 2007), resultando em uma previsão mais subjetiva.

Por outro lado, os métodos quantitativos (ou objetivos) são aplicados quando os processos estão bem definidos, e que uma outra pessoa consegue replicar, ou até, pode ser programado no computar e ser aplicado sem a interação com um humano (ARMSTRONG, 1983), retornando uma previsão de demanda mais objetiva.

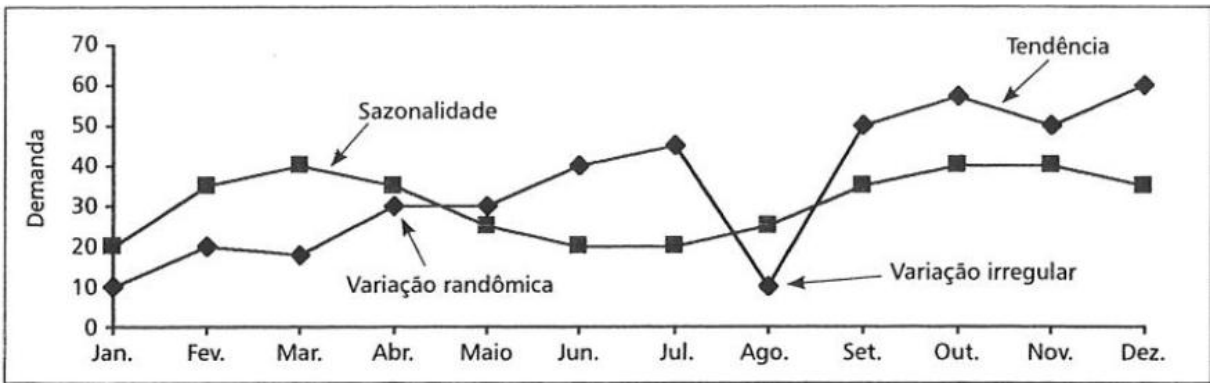
2.2 SÉRIES TEMPORAIS

As séries temporais são uma sequência de dados, obtidos em um intervalo de tempo específico (BROCKWELL e DAVIS, 2002), que podem ser coletados por meio de observações regulares da área de interesse (CARDOSO e LATORRE, 2001). Esses dados podem representar diversas áreas da economia, finanças, entre outras e auxiliam no entendimento da estrutura das informações ao longo do tempo.

A utilização da técnica de séries temporais parte do princípio de que a demanda futura será um reflexo das demandas passadas (TUBINO, 2007). Então, é analisado o padrão de comportamento daquele produto específico nos períodos anteriores e espera-se que mantenha o comportamento.

Para fazer essa análise, é necessário conhecer as quatro possíveis componentes que podem afetar o comportamento da demanda, sendo elas: efeito de tendência, efeito sazonal, ciclo de negócios e variações irregulares (FIGURA 1). A primeira traz um movimento crescente ou decrescente ao longo do tempo, a segunda mostra padrão de comportamento em épocas específicas do ano, a terceira tem relação com a economia, que pode flutuar em periodicidade indefinida. Por último, as variações irregulares e aleatórias são quando tem natureza aleatória e não são previstas (TUBINO, 2007).

FIGURA 1 - FATORES QUE INFLUENCIAM AS SÉRIES TEMPORAIS



FONTE: Tubino (2007).

Além disso, Stock e Watson (2004) trazem a distinção de séries temporais não estacionárias (FIGURA 2) e estacionárias (FIGURA 3), a primeira é descrita como uma série que suas características estatísticas (média e variância) são constantes ao longo do tempo, e a segunda sendo o contrário da primeira. O retorno de ações e crescimento do PIB são exemplos de séries estacionárias, enquanto índices de preço e nível de produto são exemplos de séries não estacionárias (BUENO, 2011).

FIGURA 2 - SÉRIE TEMPORAL ESTACIONÁRIA

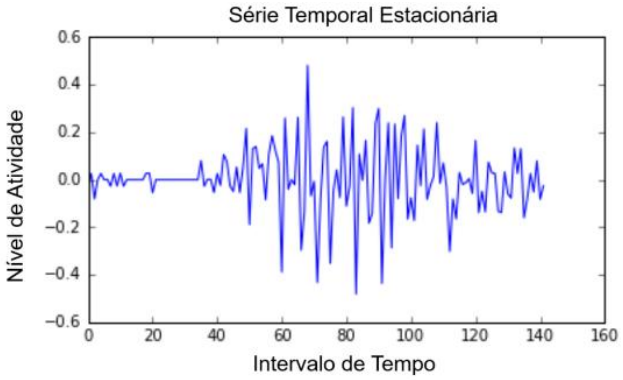
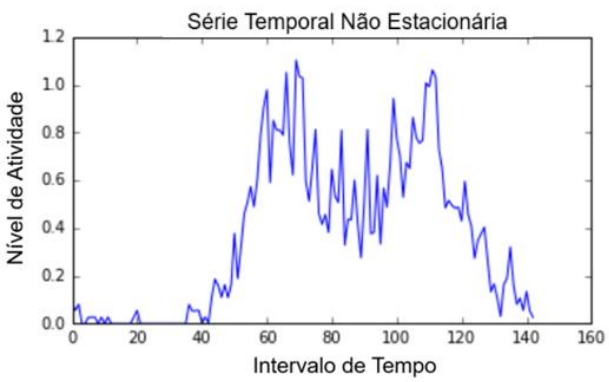


FIGURA 3 - SÉRIE TEMPORAL NÃO ESTACIONÁRIA



FONTE: Adaptado de Sultan *et al*. (2018).

Guajarati e Porter (2011) ressaltam que quando uma série temporal é não estacionária, não é possível generalizar para outros períodos, então aquela previsão, fica restrita àquela série temporal analisada.

Guajarati e Porter (2011) ressaltam que quando uma série temporal é não estacionária, não é possível generalizar para outros períodos, então aquela previsão, fica restrita àquela série temporal analisada.

E para verificar a estacionariedade, Reimbold *et al.* (2017) indicam dois testes populares para a verificação da estacionariedade de uma série temporal, o ADF (*Augmented Dickey & Fuller*):

H_0 : tem raíz unitária (não é estacionária);

H_1 : não tem raiz unitária (é estacionária).

E o KPSS (*Kwiatkowski Philips Schmidt & Shin*) considera a hipótese nula (H_0) uma série que não tem raiz unitária e, portanto, é estacionária (REIMBOLD *et al.*, 2017).

Após aplicar os testes de estacionariedade, se constatada a não estacionariedade, é possível aplicar métodos para torná-las estacionárias, sendo o mais comum o método de diferenciação, que consiste em analisar as primeiras diferenças das séries temporais (GUAJARATI e PORTER, 2011). A equação (1) define como deve-se tomar as primeiras diferenças da série temporal.

$$\Delta Z_t = Z_t - Z_{t-1} \quad (1)$$

Sendo Z_t a série temporal no período t, Z_{t-1} a série temporal no período $t-1$ e ΔZ_t a diferença da série temporal entre os períodos t e $t-1$. Se o processo de diferenciação ocorrer uma vez e a série resultante já for estacionária, diz-se que a série temporal original é integrada de ordem 1, caso a série diferenciada uma vez não se torne estacionária, pode-se diferenciar mais vezes, até que ela se torne estacionária.

2.3 MÉDIAS MÓVEIS SIMPLES

A definição de Médias Móveis Simples dada por Morettin e Toloi (2004) é o cálculo da média aritmética dos valores do período observado, ou seja, a previsão futura será dada pela última média móvel calculada. A peculiaridade deste método é, que quando uma nova observação é feita, ela substitui a observação mais antiga, por isso o termo “móvel”.

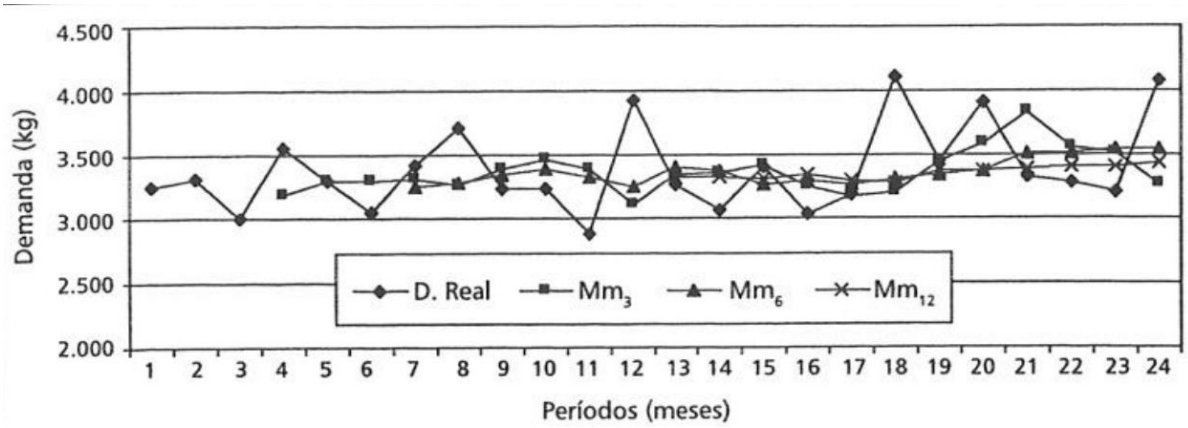
O cálculo matemático para utilização desse método é dado pela equação (2).

$$\hat{Y}_{(t+1)} = \frac{Y_t + Y_{t-1} + \cdots + Y_{t-k+1}}{k} \quad (2)$$

O valor de k na equação (2) é a quantidade de observações do passado que serão incluídas na análise, a variável Y_t é o valor da observação no período t, e $\hat{Y}_{(t+1)}$ é o valor previsto da série no período $t+1$. A Figura 4 apresenta a aplicação do método de média móvel para três valores diferentes de k: três, seis e doze meses. É importante ressaltar que quanto menor o período a análise se torna mais sensível às mudanças, e já com intervalos de períodos muito grandes, o modelo trata as médias de uma forma mais homogênea (TUBINO, 2007), deixando a informação muito

genérica e podendo atrapalhar o desempenho. Portanto, o valor do parâmetro k deve ser escolhido de modo que capture as características da série temporal em questão.

FIGURA 4 - DEMANDA REAL E PREVISTA PELA MÉDIA MÓVEL SIMPLES PARA DIFERENTES VALORES DE k



FONTE: Tubino (2007).

A vantagem deste modelo é que, segundo Ragsdale (2021), é o modelo mais fácil de usar para fazer previsão de demanda, já que o cálculo base é um cálculo simples de média aritmética. Porém, ele não é aplicável para alto volume de dados, e só consegue fazer a previsão do período posterior àquele analisado, se tornando um método com limitações (TUBINO, 2007). Esse método é idealmente aplicado em séries temporais estacionárias, visto que a presença de tendências ou sazonalidades podem afetar a precisão e a interpretação das previsões (HYNDMAN e ATHANASOPOULOS, 2018).

2.4 REDES NEURAIS ARTIFICIAIS

2.4.1 HISTÓRIA

O início das Redes Neurais Artificiais (RNA) se deu por volta de 1943, com Warren McCulloch e Walter Pitts (McCULLOCH e PITTS, 1943), quando eles criaram o primeiro modelo de rede neural, conhecida como MCP (McCulloch-Pitts), onde existe um número n de entradas, que se multiplicam por alguns pesos e são comparados.

Este primeiro modelo trouxe muito reconhecimento para a área de redes neurais artificiais, e influenciou outros estudos, como o de Frank Rosenblatt em 1958, que criou um modelo utilizando neurônios *Perceptrons* de Múltiplas Camadas ou *MultiLayer Perceptron* (MLP) do seu termo em inglês, e a arquitetura do modelo composta por duas camadas (ROSENBLATT, 1969).

Em 1969, Minsky e Papert publicaram um estudo trazendo as desvantagens do modelo *perceptron*, já que ele não garantia a solução de problemas não-linearmente separáveis (MINSKY e PAPERT, 1969). Isso fez com que os estudos sobre RNA tenham enfraquecido bastante durante a época.

Porém, em 1982, Hopfield trouxe novamente os estudos de Redes Neurais Artificiais, com a criação de um sistema neurocomputacional baseado no sistema neural de uma lesma, reacendendo o interesse no tema.

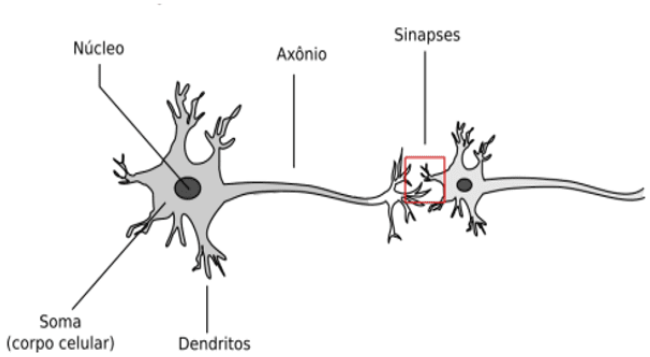
Com o avanço dos estudos, o problema citado por Minsky e Papert (1969) foi resolvido com o método *back-propagation*, proposto por Rumelhart, em que os pesos sinápticos podem ser atualizados em diversas camadas e se torna possível resolver problemas não linearmente separáveis (SILVA *et al*., 2010).

2.4.2 NEURÔNIO BIOLÓGICO

Como as redes neurais são baseadas em neurônios biológicos, se faz necessária uma explicação sobre como funciona o sistema neural.

Os neurônios são células nervosas que determinam o raciocínio e o funcionamento do corpo humano. Os elementos principais dos neurônios são o axônio, dendritos e corpo celular (FIGURA 5). Os dendritos recebem as informações, que são armazenadas no corpo celular e depois repassadas pelo axônio, até outro neurônio, e essa passagem de informação de um neurônio para outro é chamada de sinapse.

FIGURA 5 - NEURÔNIO BIOLÓGICO



FONTE: Azevedo (2016).

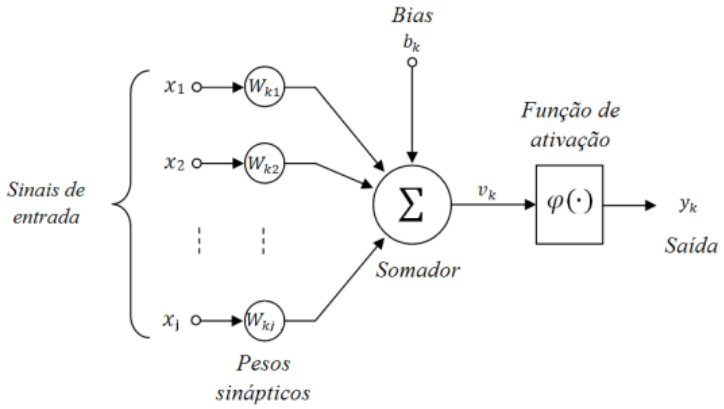
2.4.3 NEURÔNIO ARTIFICIAL

Haykin (2001) observa a relação entre os neurônios biológicos e os neurônios artificiais pelo processo de aprendizado, visto que, para os dois tipos de neurônios, essa aprendizagem é armazenada e, para os neurônios artificiais, é armazenada pelos pesos sinápticos. Ludwig Jr. e Costa (2007) complementam a definição de neurônio artificial como unidades de processamento distribuídas de forma paralela, visto que os neurônios recebem informações simultaneamente.

Haykin (2001) explica alguns elementos principais dos neurônios artificiais (FIGURA 6):

a) Sinais de entrada: valores passados para o modelo;
b) Pesos sinápticos: valores que representam a relevância do valor de entrada;
c) Somador: faz a soma das entradas recebidas com seus respectivos pesos;
d) *Bias*: uma incremental na função, para variar a entrada da função de ativação;
e) Função de ativação: gerar uma limitação do valor de saída do neurônio.

FIGURA 6 - NEURÔNIO ARTIFICIAL



FONTE: Haykin (2001).

Em resumo, o processamento do neurônio se dá da seguinte forma: com os valores de entradas e seus respectivos pesos, o corpo do neurônio realizará a soma da multiplicação de cada neurônio por seu peso sináptico. Após isso, é possível acrescentar a etapa de *bias*, pois ela agrega no resultado, visto que dá maior liberdade nos valores de entrada, aproxima a rede e faz com que nenhuma saída do neurônio seja nula (LUDWIG JR. e COSTA, 2007). Todo esse processamento pode ser resumido pela equação (3).

$$y_k = \sum_{i=1}^{n} w_{ik} \cdot x_i + b_k \quad (3)$$

O valor de y_k é a saída do corpo celular no neurônio k, e que está pronto para entrar na função de ativação, responsável por criar uma limitação nos valores de saída de uma camada do neurônio para outra. A variável w_{ik} é o peso sináptico de cada valor de entrada i, este representado pela variável x_i e o valor de b_k representa o *bias*. (LUDWIG JR. e COSTA, 2007; SILVA *et al*., 2010).

O valor de y_k passa então por uma função de ativação φ, para limitar a saída de um neurônio e também introduzir a não linearidade ao modelo (HAYKIN, 2001). Há diversas funções de ativação, que podem ser aplicadas, e o que vai indicar qual a melhor função, serão as características do modelo (MUNAKATA, 2008). As principais funções de ativação estão descritas no Quadro 1.

QUADRO 1 – PRINCIPAIS FUNÇÕES DE ATIVAÇÃO

Nome da função	Equação
Sigmoide Logística	$\varphi(y_k) = \frac{1}{1 + e^{-y_k}}$
Tangente Hiperbólica	$\varphi(y_k) = \frac{e^{y_k} - e^{-y_k}}{e^{vy_k} + e^{-y_k}}$
Gaussiana	$\varphi(y_k) = e^{\frac{-(y_k - \mu)^2}{2\sigma^2}}$
Linear	$\varphi(y_k) = \alpha y_k , \alpha \in \mathbb{R}$

FONTE: A Autora (2024)

Quanto à arquitetura das redes neurais, Silva *et al*. (2010, p. 45) afirmam que "a arquitetura de uma rede neural artificial, define a forma como diversos neurônios estão arranjados, ou dispostos, uns em relação aos outros". Visto isso, Grzeidak (2016) aponta dois grandes grupos de arquitetura de redes neurais: *feedforward networks* (estáticas ou não recorrentes) e *feedback networks* (dinâmicas ou recorrentes). O primeiro grupo abrange as redes neurais que não possuem laços, e os neurônios de uma camada só estão conectados aos neurônios da camada seguinte. Já o segundo grupo, percorre mais do que um caminho, e por conta disso, as saídas podem ser realimentadas como entradas para a mesma ou camadas anteriores, criando ciclos.

A definição da arquitetura de rede neural, é o ponto principal para definir o processo de aprendizagem do modelo (HAYKIN, 2001). E Munakata (2008, p.11) define o processo de aprendizagem como: "uma rede neural aprende padrões, ajustando os pesos".

Há três tipos de algoritmos para treinamento das redes neurais: o supervisionado, o não-supervisionado e o de reforço. Conforme Grzeidak (2016), para utilizar o treinamento supervisionado, é necessário que tenha um supervisor, e este irá fornecer a saída desejada do algoritmo. O segundo e o terceiro são explicados por Castro (2006), sendo que o não-supervisionado não é passado uma saída para um algoritmo, e ele vai se adaptando e agrupando os dados. Já o de reforço tem um diferencial, em que o algoritmo aprende com a interação com o ambiente, com o acompanhamento de um indicador que avalia a performance da rede neural.

Neste trabalho, será utilizado a arquitetura MLP (*Perceptron* de Múltiplas Camadas), que faz parte do grupo de redes neurais estáticas, e como algoritmo de

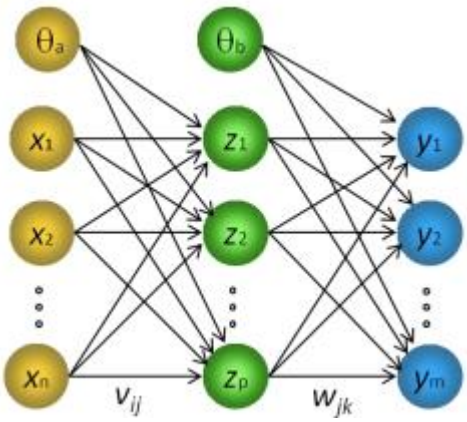
aprendizagem, será utilizado o *backpropagation,* que faz parte da aprendizagem supervisionada.

2.4.4 PERCEPTRON DE MÚLTIPLAS CAMADAS

Esse modelo de rede neural é caracterizado por ter uma ou mais camadas intermediárias entre a entrada e a saída, que são responsáveis por parte do processamento da rede e não interagem diretamente com o ambiente externo.

Na Figura 7, Siqueira (2021) mostra as camadas de entrada, camadas ocultas e camadas de saída de uma rede neural artificial. A rede apresenta n neurônios na entrada, p na camada oculta e m na camada de saída. Todos os neurônios da camada de entrada se conectam com todos os neurônios da camada oculta, assim como, todos os neurônios da camada oculta se conectam com todos os neurônios da camada de saída. Essa conexão das camadas se dá por conta dos pesos sinápticos, representados pelas letras v e w, respectivamente.

FIGURA 7 - EXEMPLO DE UMA RNA COM UMA CAMADA OCULTA



FONTE: Siqueira (2021)

Por conta disso, vale ressaltar a importância da definição da quantidade de neurônios e camadas da rede neural. Para alguns problemas, duas ou mais camadas ocultas facilitam o treinamento da rede, porém quando existem camadas em excesso, as mesmas podem interferir no cálculo do erro, não sendo possível utilizá-las de forma precisa e assertiva (BRAGA *et al.,* 2000).

Braga *et al.* (2000) também complementam que, a quantidade de neurônios na camada oculta da rede apresenta dois problemas que podem ocorrer com o

excesso ou redução dos neurônios, *overfitting* e *underfitting*, respectivamente. O primeiro ocorre quando a rede possui uma alta precisão nos dados de treinamento, mas baixo nos dados de treino, ou seja, não consegue generalizar sua aprendizagem. Já o segundo, o modelo não consegue representar de maneira adequada a relação entre as variáveis de entrada e saída, o que impede de fazer boas previsões.

A rede MLP possui três principais elementos: os pesos sinápticos, os *bias,* e a função da ativação, já explicados na seção 2.4.3 deste trabalho. Durante o treinamento, os pesos sinápticos são atualizados, de acordo com o valor do erro daquela iteração, até que atinja um critério de parada, podendo ser o número de iterações ou o valor do erro mínimo.

2.4.5 PROCESSO DE APRENDIZAGEM

Como citado anteriormente, este trabalho irá abordar o *backpropagation* como algoritmo de treinamento da rede. De acordo com Costa Jr. *et al.* (1998), o principal foco desse algoritmo é minimizar a função do erro, por meio do gradiente descendente, dessa forma, é possível ajustar os pesos para achar um valor satisfatório para o erro, ou então, ajustar a arquitetura da rede e acrescentar mais neurônios.

O treinamento desse algoritmo se divide em duas fases: propagação *forward* e propagação *backward.* De acordo com Braga *et al.* (2000), nessa primeira fase os pesos e os *bias* não sofrem alterações, já na segunda fase há a atualização dos pesos sinápticos e dos *bias*. Utilizando de exemplo uma arquitetura de rede que possui uma camada de entrada com n neurônios, uma oculta com p neurônios e uma camada de saída com um neurônio, tem-se o seguinte processo, onde $i = 1, \dots, n$, $j = 1, \dots, p$ e $k = 1, \dots, m$:

1. *Forward:*
    a. Definir pesos aleatórios entre (-1,1) para os pesos sinápticos da camada de entrada para a camada oculta (v_{ij} e θ_a) e da camada oculta para a camada de saída (w_{jk} e θ_b).
    b. Realizar a etapa de somatório da multiplicação do valor do neurônio de entrada com o seu respectivo peso de entrada. Esse processo deve ser feito para todos os n neurônios de entrada. A equação (4)

representa como deve ser feita essa ativação da camada oculta para cada neurônio k dessa camada.

$$z_j^* = \sum_{i=1}^{n} v_{ij} * x_i + \theta_a \quad (4)$$

c. Aplicar a função de ativação para cada saída z_j^* dos neurônios p da camada oculta, como apresentado na equação (5) utilizando, por exemplo, uma das funções de ativação citadas na seção 2.4.3 deste estudo.

$$z_j = \varphi(z_j^*) \quad (5)$$

d. Realizar a etapa de somatório da multiplicação do valor do neurônio da camada oculta z_j com seu respectivo peso de saída. Esse processo deve ocorrer como mostra a equação (6).

$$y_k^* = \sum_{j=1}^{p} w_{jk} \cdot z_j + \theta_b \quad (6)$$

e. Para finalizar a fase *forward*, aplicar novamente a função de ativação, agora para os neurônios da camada de saída, como está representado na equação (7).

$$y_k = \varphi(y_k^*) \quad (7)$$

2. *Backward*

a. Com o resultado da camada de saída, descoberto na fase anterior, é necessário calcular o erro, para identificar a performance do modelo. A equação (8) representa o cálculo do erro para um determinado padrão, sendo que d_k é o valor esperado e y_k é o valor do neurônio da camada de saída.

$$E_k = \frac{1}{2}(d_k - y_k)^2 \quad (8)$$

b. Após isso, calcula-se o ajuste dos pesos a partir da equação (10), onde α é a taxa de aprendizagem (9), e atualizam-se os pesos sinápticos na camada de saída em direção a camada oculta, como mostra a equação (11).

$$\alpha = 0{,}95 * \alpha \quad (9)$$

$$\Delta w_{jk} = \alpha y_k(1 - y_k) * (d_k - y_k) * z_j \quad (10)$$

$$w_{jk} = w_{jk} + \Delta w_{jk} \quad (11)$$

c. Também se calcula o ajuste dos *bias* da camada saída para a camada oculta, e atualiza-se seus valores, conforme equação (12) e (13), respectivamente.

$$\Delta\theta_b = \alpha y_k(1 - y_k) * (d_k - y_k) \quad (12)$$

$$\theta_b = \theta_b + \Delta\theta_b \quad (13)$$

d. Analisando o processo da camada oculta em direção à camada de entrada, também é necessário fazer os mesmos passos anteriores, ajustar os pesos (14) e os *bias* (15) conforme o erro calculado e atualizá-los (16) e (17), respectivamente, onde $\varphi'(*)$ é a derivada da função de ativação.

$$\Delta v_{ij} = \alpha \sum_{k=1}^{m} \left[(d_k - y_k)\varphi'(y_k^*)w_{jk}\right]\varphi'\left(z_j^*\right)x_i \quad (14)$$

$$v_{ij} = v_{ij} + \Delta v_{ij} \quad (15)$$

$$\Delta\theta_a = \alpha(d_k - y_k) * y_k * (1 - y_k) * w_{jk} * z_j * \left(1 - z_j\right) \quad (16)$$

$$\theta_a = \theta_a + \Delta\theta_a \quad (17)$$

2.4.6 MODOS DE TREINAMENTO

Haykin (2001) aborda os modos de treinamento de uma rede neural artificial da seguinte forma: dados os valores de entrada no seguinte formato $\left((x_1, d_1), (x_2, d_2), \ldots, (x_N, d_N)\right)$, onde x_i é o valor de entrada é d_i é a resposta para a *i-ésima* observação do conjunto de dados de tamanho N.

O treinamento pode ser de forma sequencial ou por lote. No modelo sequencial, cada subconjunto é testado, e seus pesos sinápticos e *bias* são atualizados, então realizam-se as fases de *forward* e *backward* N vezes, para cada iteração.

Já para o modelo por lote, a atualização dos pesos sinápticos e *bias* ocorre somente uma vez, pois é passado o subconjunto inteiro em uma única iteração e as fases *forward* e *backward* são realizadas somente uma vez (HAYKIN, 2001).

Apesar de existir esses dois modelos, Haykin (2001) deixa claro que é preferível utilizar o modo de treinamento sequencial e pontua algumas vantagens para esse modo:

a. É computacionalmente mais rápido;
b. Armazena menos dados para cada neurônio;
c. Proporciona dinamismo para o modelo;
d. Evita que algoritmo *backpropagation* estacione em um mínimo local.

2.4.7 CRITÉRIOS DE PARADA

Levando em consideração o objetivo principal do algoritmo *backpropagation*, que é a minimização do erro médio, um critério de parada que pode ser adotado é valor do erro médio de uma determinada iteração ser suficientemente pequeno, ou seja, o valor das saídas desejadas com as saídas calculadas serem muito próximos (BRAGA *et al.*, 2000).

Além disso, outros possíveis critérios de parada, segundo Braga *et al.* (2000), são:

a. Quantidade de iterações;
b. Percentual aceitável de saídas satisfatórias;
c. Tempo de processamento do algoritmo;
d. Todas as sugestões anteriores combinadas.

2.5 MEDIDA DE ERRO

Para analisar o desempenho dos métodos aplicados neste trabalho, será utilizada a medida de erro *Root Mean Square Error* (RMSE), que analisa a raiz do erro quadrático médio, conforme a equação (18).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i - O_i)^2} \qquad (18)$$

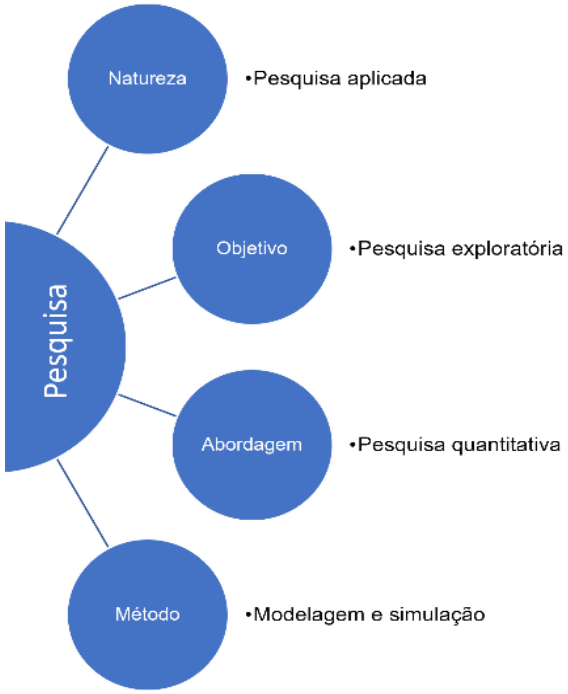
Onde, E_i é o valor estimado, O_i é o valor medido e n é o número de observações. O valor do RMSE é sempre positivo, e, quando mais próximo de zero,

melhor é o desempenho das previsões. A única desvantagem do método é que, se houverem alguns valores destoantes, mesmo sendo poucos, o valor do erro já aumenta (STONE, 1993).

3 METODOLOGIA

Quanto a classificação dessa pesquisa, o presente trabalho pode ser classificado de acordo com várias dimensões metodológicas. Em termos de sua natureza, trata-se de uma pesquisa aplicada, pois busca-se utilizar o conhecimento desenvolvido de maneira prática em contextos reais. Quanto aos objetivos, caracteriza-se como uma pesquisa exploratória, já que envolve a coleta de dados reais de vendas de bebidas para a realização do estudo. No que diz respeito à abordagem do problema, o trabalho é quantitativo, com resultados apresentados e analisados a partir de métricas numéricas. Por fim, considerando os métodos utilizados, a pesquisa se enquadra na categoria de modelagem e simulação, pois busca-se experimentar um contexto real por meio de métodos matemáticos (GIL, 2008). A Figura 8 apresenta graficamente essa classificação.

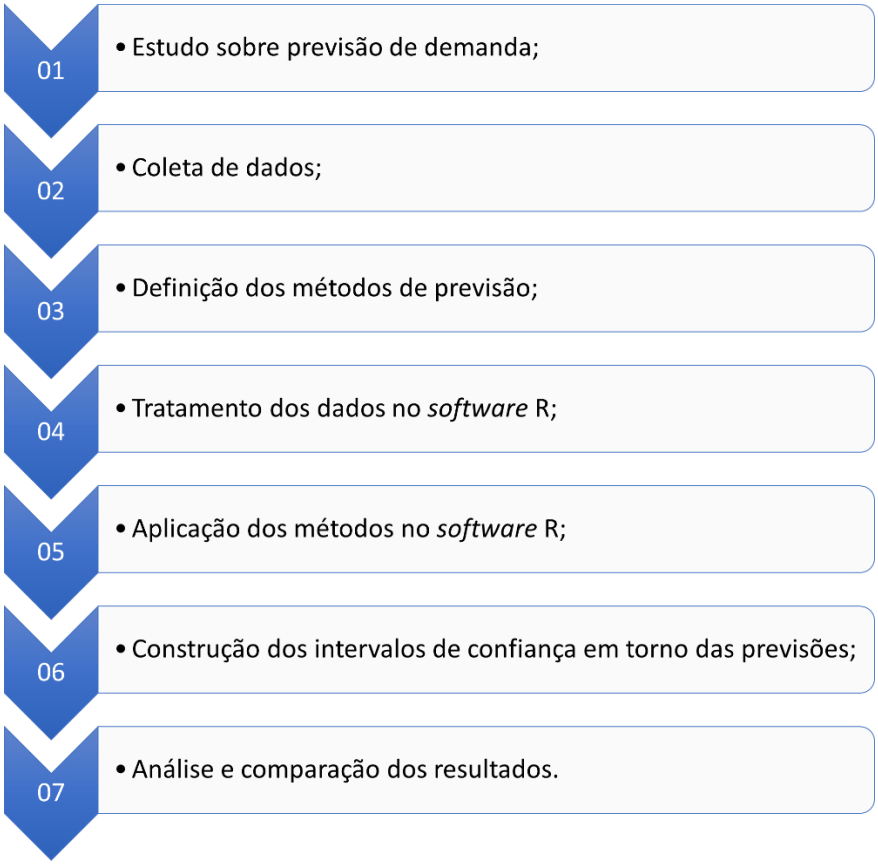
FIGURA 8 - REPRESENTAÇÃO DA CLASSIFICAÇÃO DO TRABALHO



FONTE: A autora (2024)

Para melhor desenvolvimento deste trabalho, as atividades foram divididas em etapas e apresentadas em um fluxograma (FIGURA 9).

FIGURA 9 - AS ETAPAS DA METODOLOGIA



FONTE: A autora (2024)

A primeira etapa foi voltada para o estudo do conceito de previsão de vendas e, mais especificamente, no setor de bebidas, principalmente para contextualização do tema e melhor entendimento do cenário que a base está inserida.

A segunda etapa abrange a coleta de dados e, devido à natureza confidencial das informações, foi realizada uma alteração para que pudessem ser utilizados. Para este estudo, e para maior precisão na previsão, foram feitos alguns recortes de marca e de cidade, portanto, as previsões serão especificamente da marca X em Curitiba - PR. Tem-se de histórico dos dados de volume de vendas, em hectolitros, desde o ano de 2019 até 2023, disponibilizados diariamente (exceto no domingo, que não se tem a informação), totalizando 1541 observações.

Em paralelo com a definição dos dados, a escolha dos dois métodos de previsão de demanda foi discutida na etapa 3. Apesar da diversidade de métodos de previsão de demanda, neste estudo, serão abordados dois métodos distintos: um de natureza mais simples e outro de maior robustez, especificamente, as Médias Móveis

Simples e as Redes Neurais Artificiais, em específico a rede MLP (*Perceptron* Multicamadas). Isso permitirá uma comparação entre os métodos, visando analisar se a complexidade do método impacta diretamente os resultados.

O método de Médias Móveis Simples foi adotado por conta da sua baixa complexidade (RAGSDALE, 2021). Já o método de Redes Neurais Artificiais foi utilizado para fomentar a utilização de modelos de *machine learning* na área de previsão de demanda. O modelo utilizado foi o *Multilayer Perceptron*, visto que, pela quantidade de camadas, o modelo tem um aprendizado mais assertivo, e possui um algoritmo de aprendizado supervisionado muito conhecido: o *backpropagation*, possibilitando acesso a vários recursos para o aprendizado do modelo.

Para os dois métodos aplicados, a previsão será realizada para um horizonte de 12 dias, o que abrange aproximadamente metade do mês. Essa abordagem facilita a obtenção de uma visão panorâmica do mês e permite a elaboração de planos de ação, caso necessário. Com um conjunto de dados de 1541 observações de vendas de volume em uma frequência diária, e espera-se prever 12 dias, o conjunto de teste foi definido como as últimas 12 observações, que auxiliará no cálculo do erro, enquanto o conjunto de treino será composto pelo restante das observações para a construção e treinamento dos modelos preditivos.

Antes da execução das próximas etapas, foi escolhido um *software* para auxiliar no tratamento e aplicação dos dados. O *software* selecionado foi o R, principalmente porque é gratuito, possui uma ampla variedade de pacotes pré-configurados para diversas finalidades e é extremamente potente para análises estatísticas (HYNDMAN e ATHANASOPOULOS, 2021).

A quarta etapa envolve o tratamento dos dados coletados na segunda etapa. Esse tratamento foi realizado para converter os tipos de dados. Os dados importados da planilha estavam no formato de texto (*char*), sendo necessário valores para decimal (*double*).

Após o tratamento dos dados, foi possível prosseguir com a aplicação dos métodos de previsão (quinta etapa). Para a previsão utilizando Médias Móveis Simples, a aplicação começa analisando se a série temporal é estacionária. Caso a série não seja estacionária, o método de diferenciação será aplicado até que a série se torne estacionária, permitindo a aplicação do método de Médias Móveis Simples. Em seguida, para buscar o melhor desempenho do método, foram realizados testes alterando o principal parâmetro: o intervalo do período (parâmetro k do método de

Médias Móveis Simples). Intervalos de 2 a 30 dias anteriores foram aplicados na variável para identificar qual intervalo proporciona o melhor resultado. Também foram calculados os intervalos de confiança em torno das previsões (etapa 6).

O método de Médias Móveis Simples e a construção dos intervalos de previsão foram implementados no *software* R. Para os testes de estacionariedade de uma série, foi utilizada a biblioteca *tseries* e as funções *adf.test()* e *kpss.test()* e para diferenciar uma série foi utilizada a função *diff()*.

Ainda na quinta etapa, o modelo MLP foi aplicado no *software* R para calcular as previsões. Para isso, foi necessário importar a biblioteca *forecast*, que contém funções específicas para previsão utilizando redes neurais. Entre essas funções, duas foram utilizadas: *nnetar()*, que realiza o treinamento da Rede Neural Artificial, onde são fornecidos os dados do conjunto de treino, e *forecast()*, que recebe vários parâmetros de entrada, incluindo o conjunto de teste, a rede neural treinada, a quantidade de dias a serem previstos e as informações sobre os níveis de confiança a serem considerados nos intervalos de previsão (etapa 6).

Por fim, a etapa 7 teve seu foco para a análise de resultados e comparações, com o objetivo de definir qual método teve maior assertividade, a partir do resultado do RMSE.

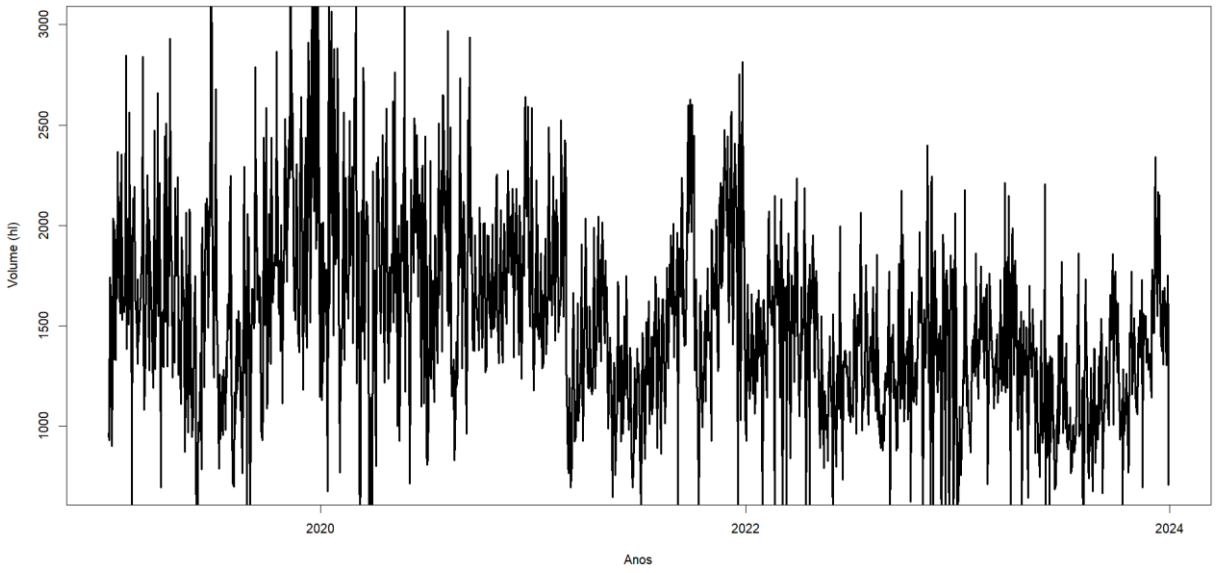
4 RESULTADO E DISCUSSÕES

Nesse capítulo, serão apresentados os resultados e discussões obtidos a partir da implementação da metodologia do capítulo anterior. Para cada método, será explicado os tratamentos feitos na base de dados, as premissas utilizadas, os resultados, e, por fim, a conclusão da aplicação do método. Como citado anteriormente, a base utilizada traz dados de volume vendido diariamente, de uma marca de bebida específica na cidade de Curitiba-PR.

No primeiro momento, foi utilizada a função *read_table*() para ler o arquivo com os dados e importar para dentro do R, possibilitando uma visualização dos dados dentro do *software* e sendo possível fazer as transformações necessárias.

A função *plot*() foi utilizada para criar o gráfico com os dados reais, para melhor análise do cenário (FIGURA 10).

FIGURA 10 - GRÁFICO COM OS DADOS REAIS



FONTE: A autora (2024)

Após fazer a análise e a plotagem dos dados em um gráfico, teve-se uma melhor compreensão sobre a série histórica, e assim foi possível começar a aplicação dos métodos propostos. Para complementar os resultados das previsões, também foram calculados os intervalos de confiança dos resultados, para que fosse considerada a incerteza relacionada a aplicação dos métodos.

4.1 APLICAÇÃO DO MÉTODO DAS MÉDIAS MÓVEIS SIMPLES

O primeiro método aplicado foi o de Médias Móveis Simples, e o ponto de partida foi a descoberta da estacionariedade da série ou não. Para isso, foram calculados os testes de ADF e KPSS (FIGURA 11), encontrados na biblioteca *tseries.* Como resultado, o teste de ADF apontou estacionariedade, porém o de KPSS não, para um nível de significância de 5%, então é indicado realizar a diferenciação, assim os dois testes concluíram a estacionariedade da série após uma diferenciação.

FIGURA 11 - APLICAÇÃO DO TESTE DE ESTACIONARIEDADE E O MÉTODO DE DIFERENCIAÇÃO

```
> adf.test(df$volume)

        Augmented Dickey-Fuller Test

data:  df$volume
Dickey-Fuller = -7.2974, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(df$volume) : p-value smaller than printed p-value
> kpss.test(df$volume)

        KPSS Test for Level Stationarity

data:  df$volume
KPSS Level = 5.8173, Truncation lag parameter = 7, p-value = 0.01

Warning message:
In kpss.test(df$volume) : p-value smaller than printed p-value
> df_diff<-diff(df$volume)
> adf.test(df_diff)

        Augmented Dickey-Fuller Test

data:  df_diff
Dickey-Fuller = -17.451, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(df_diff) : p-value smaller than printed p-value
> kpss.test(df_diff)

        KPSS Test for Level Stationarity

data:  df_diff
KPSS Level = 0.0098217, Truncation lag parameter = 7, p-value = 0.1

Warning message:
In kpss.test(df_diff) : p-value greater than printed p-value
```

FONTE: A autora (2024)

Após a realização da diferenciação, a previsão para os dados diferenciados foi calculada utilizando o método das Médias Móveis Simples. O parâmetro fundamental para a aplicação desse método foi o intervalo de dados selecionado para a previsão. Neste estudo, o horizonte de previsão foi definido para representar a metade de um mês, portanto, a previsão abrange 12 dias à frente do conjunto de

dados, caracterizando um horizonte de curto/médio prazo. Para a escolha do valor de k que representa a quantidade de dados passados utilizados no cálculo da Média Móvel Simples, foram testados valores de 2 a 30. O valor que resultou no menor RMSE para o conjunto de teste foi $k = 6$. Dessa forma, as previsões de 12 dias à frente utilizam dados dos 6 dias passados, portanto o conjunto de treinamento para o método das Médias Móveis Simples não foi usado em sua totalidade, pois foram utilizados apenas os 6 últimos dados do conjunto de treinamento para prever os próximos 12 dados, e os próprios resultados das previsões eram incorporados ao conjunto de 6 dados para prever o próximo valor.

Após realizar a previsão para os dados diferenciados, o processo inverso da diferenciação foi realizado, para que as previsões retornassem para dados de volumes de vendas.

Para o cálculo dos intervalos de confiança, primeiro foi necessário verificar se os resíduos (diferença entre os valores verdadeiros e os valores previstos) são normais. O teste utilizado para verificar a normalidade foi o de Shapiro-Wilk, que foi desenvolvido por Shapiro e Wilk em 1965 e se tornou um dos testes mais utilizados para verificar a normalidade (MENDES e PALA, 2003). Para descobrir se os resíduos seguem a distribuição normal, foi utilizada a função Shapiro.*test()* no R, que retornou um *p-valor* de 0,6477 (FIGURA 12). Como o *p-valor* apresenta um valor maior que 0,05 (nível de 5% de significância considerado), aceita-se a hipótese nula e conclui-se que os dados possuem distribuição normal.

FIGURA 12 - APLICAÇÃO DO TESTE DE NORMALIDADE SHAPIRO-WILK

```
> residuos <- numeric(length(dados_reais))
> for (i in 1:length(dados_reais)) {
+   residuos[i] = (dados_reais[i]-dados_previstos[i])
+ }
> shapiro.test(residuos)

        Shapiro-Wilk normality test

data:  residuos
W = 0.95073, p-value = 0.6477
```

FONTE: A autora (2024)

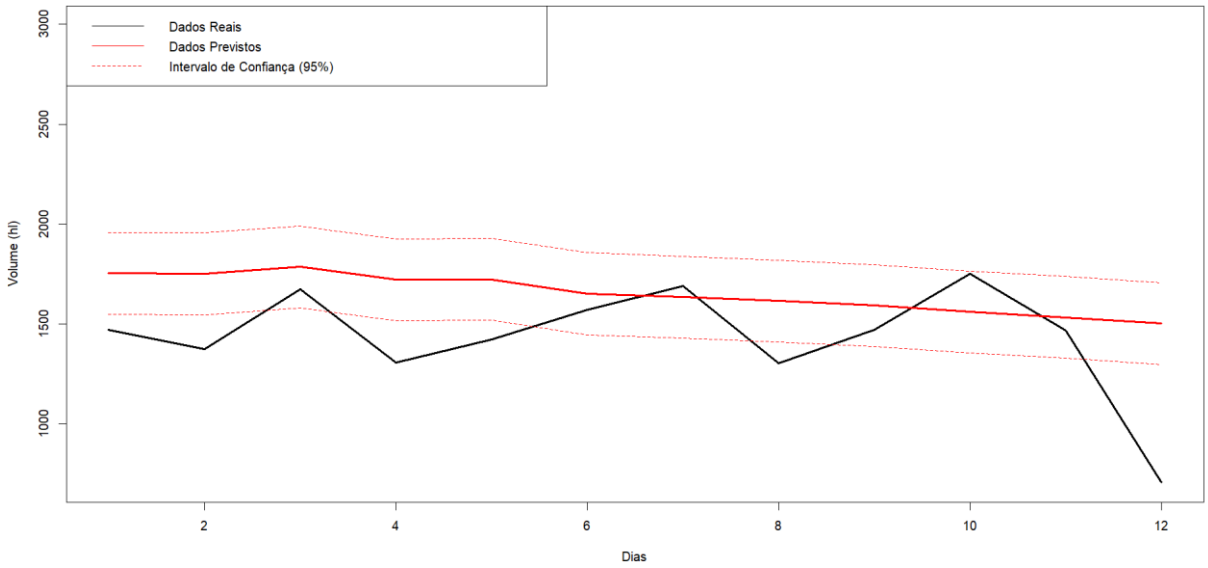
Uma vez que os resíduos possuem distribuição normal, é possível calcular intervalos de confiança para as previsões. Isso é feito a partir das definições da média μ de uma população que tenha distribuição Normal e com variância conhecida, e do coeficiente de confiança γ, como é apresentado na equação 19 (MAGALHÃES, 2008).

$$IC_{(\mu,\gamma)} = \left[\bar{X} - z_{y/2}\frac{\sigma}{\sqrt{n}};\ \bar{X} + z_{y/2}\frac{\sigma}{\sqrt{n}}\right] \quad (19)$$

Onde, $\bar{X}$ é a média dos valores previstos, $z_{y/2}$ é o índice do coeficiente de confiança, σ é o desvio padrão dos resíduos e $\sqrt{n}$ é a raiz quadrada do número de observações. A partir disso, foi estabelecido o $\gamma = 95\%$ e, obtendo o valor do índice na tabela Normal o coeficiente de confiança se iguala ao valor de 1,96.

Por fim, para melhor visualização das previsões, foi plotado em um gráfico os valores reais, valores previstos e os intervalos de confiança para o conjunto de teste, composto por 12 dados, como mostra a Figura 13.

FIGURA 13 - RESULTADO DA PREVISÃO POR MÉDIAS MÓVEIS SIMPLES



FONTE: A autora (2024)

É importante destacar que, em geral, as previsões permaneceram dentro do intervalo de confiança estabelecido. No entanto, houve alguns momentos em que os valores reais ficaram fora deste intervalo, isso pode ser causado por fatores externos que afetaram os dados reais, porém não foi possível considerá-los para as previsões.

4.2 APLICAÇÃO DO MÉTODO DE REDES NEURAIS ARTIFICIAIS

O outro método aplicado neste trabalho, foi a Rede Neural Artificial do tipo MLP e, para isso, foi utilizada a biblioteca *forecast* do R.

Para a estruturação da rede neural, foi utilizada a função *nnetar()* construída pelo Rob J Hyndman and Gabriel Caceres, que ajusta modelos de Redes Neurais *feed-foward* a séries temporais, contemplando os padrões lineares e não lineares dos dados da série temporal (R DOCUMENTATION, 2024).

Para o modelo em questão e utilizando os dados do conjunto de treino, a própria função *nnetar()* ajustou adequadamente os valores de neurônios na camada de entrada e na camada oculta, resultando em uma rede treinada que possui 28 neurônios na camada de entrada e 14 neurônios na camada oculta, o que significa que o modelo utiliza 28 dados passados para prever um valor à frente. A Figura 14 mostra o resultado da aplicação da função, enquanto a Figura 15 plota a rede, para melhor visualização.

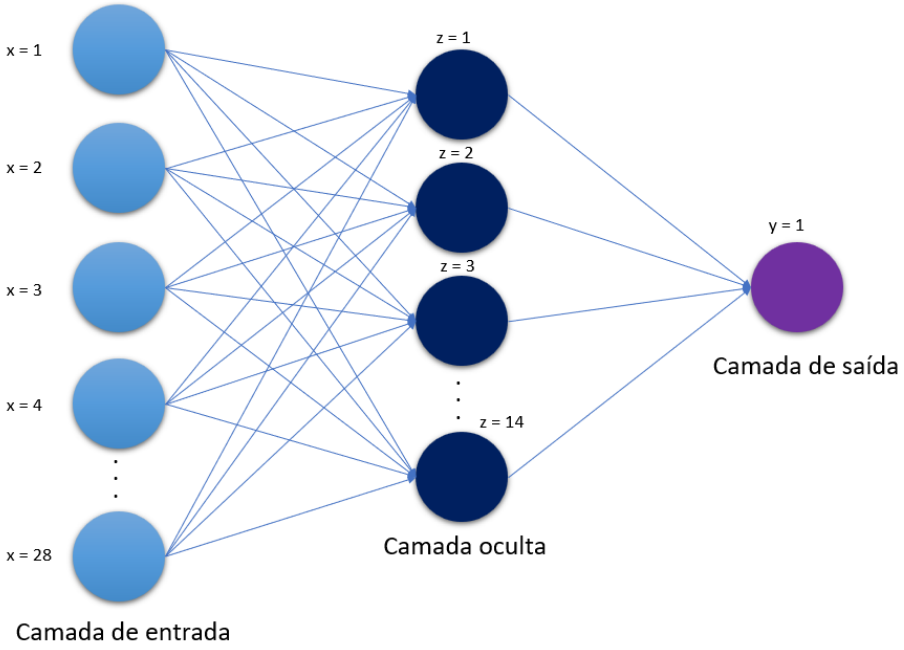
FIGURA 14 - RESULTADO DO TREINAMENTO DA REDE NEURAL

```
> fit
Series: df$volume[1:1529]
Model:  NNAR(28,14)
Call:   nnetar(y = df$volume[1:1529])

Average of 20 networks, each of which is
a 28-14-1 network with 421 weights
options were - linear output units
```

FONTE: A autora (2024)

FIGURA 15 - ARQUITETURA DA REDE NEURAL TREINADA



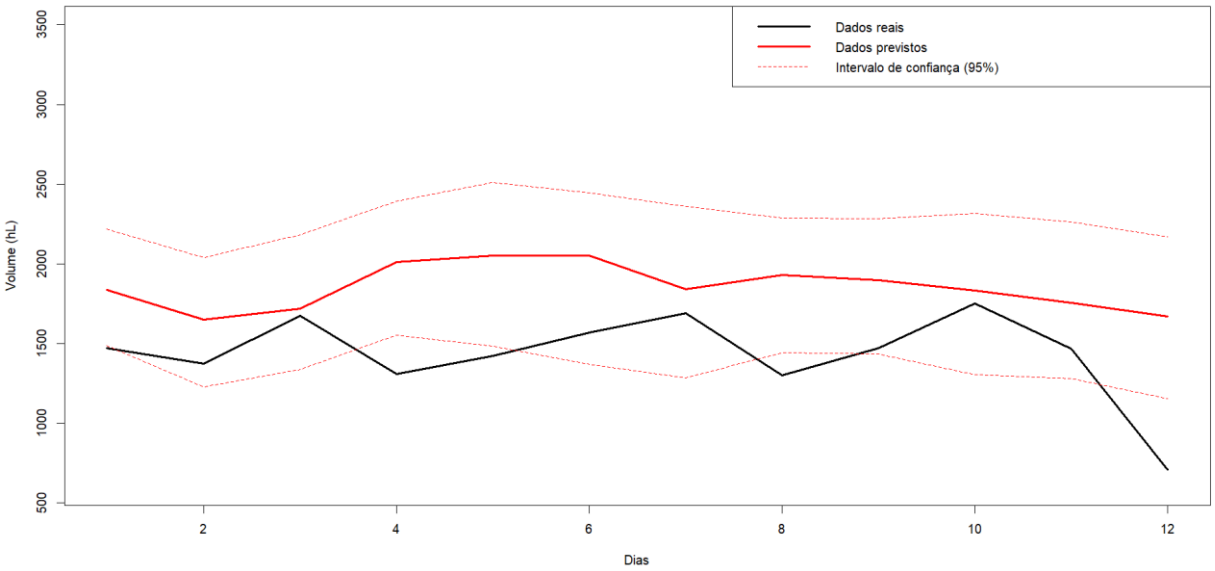
FONTE: A autora (2024)

Com a rede neural treinada, a previsão do volume de vendas foi feita pela função *forecast()*, que utiliza como parâmetros a rede neural treinada, a quantidade de dados a serem previstos à frente, se deve calcular os intervalos de previsão, quantidade de trajetórias a serem utilizadas para calcular o intervalo de previsão, e, por fim, o nível de confiança para o intervalo de previsão (RDocumentation, 2024). Os nomes dos respectivos parâmetros e os valores utilizados são: *fit*: rede treinada; $h = 12$; $PI = TRUE$; $npaths = 1000$; e $level = 95$.

É importante destacar que, para o cálculo dos intervalos de previsões com o parâmetro $PI = TRUE$, a função faz várias simulações ajustando o valor do erro da série, seja de uma forma aleatória ou por distribuição normal. A partir disso, com valores diferentes de erros sendo ajustados, é possível simular várias previsões e analisar a distribuição dos caminhos que foram criados. É com essa funcionalidade que a função *forecast()* calcula os intervalos de previsão.(HYNDMAN, 2017)

A partir da definição desses parâmetros, 12 dados à frente foram previstos e plotados em um gráfico, junto com os dados reais e os intervalos de previsão com 95% de confiança, como é apresentado na Figura 16.

FIGURA 16 - RESULTADO DA PREVISÃO POR RNA



FONTE: A autora (2024)

Para a previsão por MLP, o cenário ficou parecido com o método das Médias Móveis Simples, visto que os valores previstos ficaram dentro do intervalo de confiança, mas alguns valores reais, em alguns momentos, ficaram fora do intervalo, sendo ocasionado por fatores externos não contemplados no modelo.

4.3 COMPARAÇÃO DOS RESULTADOS E DISCUSSÃO

A comparação dos métodos aplicados foi feita a partir do cálculo da medida de erro RMSE. Para o método das Médias Móveis Simples o RMSE obteve o resultado de 245,1516, e foi calculado a partir da comparação dos valores previstos com os valores reais do conjunto de teste (12 valores). Ou seja, o erro calculado indica que as previsões estão desviando, em média, 245 hectolitros em relação ao valor real.

A avaliação do modelo de previsão pelo método de Redes Neurais também foi feita pelo cálculo do RMSE, e foi utilizada a função *accuracy()*, que tem como parâmetro os dados reais e os dados previstos, e, assim, traz o cálculo que diversas medidas de erro. A função retornou um valor de 495,2795, ou seja, há uma variação de, aproximadamente 495 hectolitros nas medidas previstas, em comparação com os dados observados.

O Quadro 2 apresenta os resultados obtidos a partir dos dois modelos aplicados, foi possível identificar que, com os dados disponíveis e as premissas aplicadas, o método de Médias Móveis Simples possui um resultado mais satisfatório, em relação ao método da Rede Neural Artificial MLP.

QUADRO 2 - COMPARAÇÃO DE RESULTADOS

	MÉDIAS MÓVEIS SIMPLES	**REDES NEURAIS ARTIFICIAIS**
DIAS PREVISTOS	12	12
RMSE	245,1516	495,2796

Fonte: A autora (2024)

5 CONSIDERAÇÕES FINAIS

O presente trabalho investigou o desempenho de dois métodos de previsão de demanda, Médias Móveis Simples e Redes Neurais Artificiais, utilizando o modelo MLP. Os resultados indicaram que, para este conjunto de dados de vendas de bebidas, o modelo de Médias Móveis Simples apresentou um desempenho superior em relação às Redes Neurais.

Essa comparação foi realizada a partir do cálculo do RMSE, que analisa a raiz do erro quadrático médio. Para as Médias Móveis Simples e as Redes Neurais, os valores de RMSE foram de 245,1516 hL e 495,2796 hL, respectivamente. Isso indica que, embora o método de Redes Neurais seja mais robusto, ele não obteve o melhor resultado neste caso específico.

Este estudo também destaca a importância da aplicação e eficácia de modelos preditivos, sejam eles mais simples ou mais complexos, principalmente para obter-se melhor planejamento quanto aos produtos e evitar desperdícios. A utilização do *software* R também trouxe uma gama de funções e flexibilidade para análises estatísticas e previsões.

Entre as limitações do estudo, destaca-se a restrição da análise a um conjunto específico de dados de vendas de bebidas, o que pode limitar a generalização dos resultados. Além disso, a configuração dos parâmetros dos modelos foi baseada em testes empíricos para este conjunto de dados específico, o que pode não representar a configuração ideal para todos os tipos de séries temporais. Portanto, vale ressaltar que não há métodos melhores ou piores, cada método possui sua particularidade, assim como o conjunto de dados utilizado.

Para futuras pesquisas, várias possibilidades podem ser exploradas:

a) Na aplicação de Redes Neurais, a inclusão de variáveis exógenas, como informações externas que possam impactar os dados;
b) A avaliação e aplicação de outros parâmetros para os modelos utilizados, permitindo obter resultados diversos e identificar com quais parâmetros os modelos se comportam melhor;
c) A utilização de outros modelos preditivos, com foco especial em modelos de aprendizado de máquina, pode enriquecer significativamente o estudo.

Por fim, o estudo responde à pergunta feita inicialmente nesse trabalho, ressaltando que, para esse caso em específico, quem obteve a previsão mais assertiva foi o método das Médias Móveis Simples, porém é necessário sempre analisar melhor o contexto que os dados estão inseridos e, assim, escolher o melhor método adequado. Dessa forma, apenas a terceira hipótese foi validada pela conclusão do estudo.

REFERÊNCIAS

ABRABE - Associação Brasileira de Bebidas. **Um brinde à vida – A história das bebidas**. 2014. Disponível em: https://www.abrabe.org.br/wp-content/uploads/2016/08/DBA-Abrabe-vFINAL.pdf . Acesso em: 2 jun. 2024.

ACKERMANN, A. E. F.; SELLITTO, M. A. **Métodos de previsão de demanda: uma revisão da literatura**. **Innovar**, Bogotá, 2022.

ARMSTRONG, J. **Strategic Planning and Forecasting Fundamentals**, In: ALBERT, K The Strategic Management Handbook. New York: MacGraw Hill, 1983.

ARVAN, M. *et al*. **Integrating human judgement into quantitative forecasting methods: A review**. *Omega*, 2019, p. 237-252.

AZEVEDO, L. P. de. **Aplicação de redes neurais artificiais no processo de classificação de orquídeas do gêner**o. 2016. 49p. Dissertação (Bacharelado em Sistemas de Informação) - Instituto Federal de Minas Gerais, Campus Sabará, 2016.

BRAGA, A. de. *et al.* **Redes Neurais Artificiais teoria e aplicações**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 2000.

BUENO, R. D. L. D. S. **Econometria de Séries Temporais**. São Paulo: Cengage Learning Edições Ltda, 2008.

CASTRO L. N**. Fundamentals of natural computing: basic concepts, algorithms, and applications**. Nova Iorque: Chapman & Hall/CRC, 2006.

COSTA Jr. *et al.* **Minimização do erro no algoritmo backpropagation aplicado ao problema de manutenção de motores**. Pesquisa Operacional. vol. 18, n. 1, p. 21 - 36, 1998.

GERBER, J. Z. *et al.* **Organização de Referenciais Teóricos sobre Diagnóstico para a Previsão de Demanda**. Revista Eletrônica de Gestão Organizacional, Recife, v. 11, n. 1, p. 160-185, jan./abr. 2013.

GIL, A. C. (2008). **Métodos e técnicas de pesquisa social** (6ª ed.). São Paulo: Atlas.

GRZEIDAK, E. **Identification of Nonlinear Systems based on Extreme Learning Machine and Multilayer Neural Networks**. 2016. 152p Dissertação (Mestrado em Sistemas Mecatrônicos) - Departamento de Engenharia Mecânica, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.

GUAJARATI, D. N.; PORTER, D. C. **Econometria básica**. Grupo A, 2011. E-book. ISBN 9788580550511. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788580550511/. Acesso em: 29 nov. 2023.

HAYKIN, S. **Redes neurais princípios e prática**. Grupo A, 2001. *E-book.* ISBN 9788577800865. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788577800865/. Acesso em: 31 out. 2023.

HYNDMAN, R. J., ATHANASOPOULOS, G. (2021) **Forecasting: principles and practice**, 3ª edição, OTexts: Melbourne, Australia. OTexts.com/fpp3. Acesso em: 27 maio 2024.

HYNDMAN, R. J. (2017). **Predictions intervals for NNETAR** models. Disponível em: https://robjhyndman.com/hyndsight/nnetar-prediction-intervals/. Acesso em 04 jun. 2024.

KANTAR. **Mercado de bebidas cresce nas altas temperaturas**. São Paulo: Kantar, 2024. Disponível em: <https://www.kantar.com/brazil/inspiration/consumo/2024-wp-mercado-de-bebidas-bra>. Acesso em: 1 jun. 2024.

LUDWIG, O.; COSTA, E. M. M**. Redes Neurais: Fundamentos e Aplicações com Programas em C**. Rio de Janeiro: Editora Ciência Moderna, 2007.

McCULLOCH, W.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics**, v.5, p.115-133, 1943.

MENDES, M.; PALA, A. (2003). **Type I Error Rate and Power of Three Normality Tests**. Pakistan Journal of Information and Technology 2(2), pp. 135-139.

MINSKY, M. L.; PAPERT, S. A. **Perceptrons**, MIT Press, Cambridge, MA. 1969.

MIRANDA, R.G. *et al.* **Método estruturado para o processo de planejamento da demanda nas organizações**. Congresso Internacional de Administração. ADM2011, Ponta Grossa – Brasil, 2011.

MORETTIN, P. A; TOLOI, C. M. C. **Análise de Séries Temporais**. Editora Blucher, 2018. *E-book.* ISBN 9788521213529. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521213529/. Acesso em: 31 out. 2023.

MUNAKATA, T. **Fundamentals of the new artificial intelligence: neural, evolution, fuzzy and more**. 2 ed. Cleveland: Springer, 2008.

RAGSDALE, C. T. **Modelagem de planilha e análise de decisão: uma introdução prática a business analytics**. [Digite o Local da Editora]: Cengage Learning Brasil, 2021. *E-book.* ISBN 9788522128303. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788522128303/. Acesso em: 31 out. 2023.

R DOCUMENTATION. 2024. Disponível em: <https://www.rdocumentation.org/>. Acesso em: 30 maio 2024.

REIMBOLD, M. M. P. *et al*. **APLICAÇÃO DE TESTE DE RAIZ UNITÁRIA ÀS VARIÁVEIS DE PROPULSORES ELETROMECÂNICOS**. Vivências: Revista Eletrônica de Extensão da URI, Erechim, 2017, p. 47-49

ROSENBLATT, F. **The perceptron: A probabilistic model for information storage** and organization in the brain. Psychological Review, v.65, n.6, p. 386-408, 1958.

SILVA I. N. da. *et al*. **Redes Neurais Artificiais para Engenharia e Ciências Aplicadas curso prático**. Artliber, 2010.

SIQUEIRA, P.H. "**Metaheurísticas e Aplicações**". Disponível em: <https://paulohscwb.github.io/metaheuristicas/>, Acessado em: 20 jan. 2021.

STOCK, J. H.; WATSON, M. W. **Econometria**. São Paulo: Pearson, 2004.

STONE, R. J. **Improved statistical procedure for the evaluation of solar radiation estimation models**. Solar Energy, v. 51, n. 4, p. 289-291, 1993.

SULTAN, K., ALI, H.; ZHAN, Z. **Call Detail Records Driven Anomaly Detection and Traffic Prediction in Mobile Cellular Networks**. 2018.

TUBINO, D. F. **Planejamento e Controle de Produção: Teoria e Prática**. São Paulo: Atlas S.A., 2007.

VEIGA, C. P.; DUCLÓS, L. C**. A Acurácia dos Modelos de Previsão de Demanda Como Fator Crítico para o Desempenho Financeiro na Industria de Alimentos**. Profuturo: Programa de Estudos do Futuro, São Paulo, v. 2, n. 2, jul./dez. 2010.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series and Forecasting**. 2. ed. New York: Springer, 2002. Disponível em: https://www.scielo.br/j/rbepid/a/KM9MndgpCGSnjSNDddSydCG/?format=pdf&lang=pt

MAGALHÃES, M. N. **Noções de Probabilidade e Estatística**. 6. ed. São Paulo: Edusp, 2008.

APÊNDICE 1 – CÓDIGO NO R PARA APLICAÇÃO DAS MÉDIAS MÓVEIS SIMPLES

```r
#CHAMANDO A BIBLIOTECA QUE SERÁ UTILIZADA

library(tseries)

#Definir a quantidade de dados para o conjunto de teste e o conjunto de treino
qtd_previsao <- 12
k <- 6

#Importação e visualização dos dados
df <- read.table("volume.csv", header = TRUE, sep = ";",encoding = "UTF-8")
print(df)

#Transformações nos tipos de dados
names(df)[names(df) == "Data"] <- "data"
names(df)[names(df) == "Volume"] <- "volume"
df$data <- as.Date(df$data, format = "%d/%m/%Y")
df$volume <- as.numeric(gsub(",", ".", gsub("\\.", "", df$volume)))
df$volume <- round(df$volume, 2)

#Visualizar a série temporal graficamente
plot(df, type = "l", col = "black", lwd = 3, xlab = "Datas", ylab = "Volume (hl)", main = "Observações reais do volume de vendas")

#Atibuir apenas os dados que serão utilizados a um novo dataframe
dados_final <- df$volume

#Definir, dentro do conjunto de dados, os índices dos dados que eu quero prever
ind_final <- length(dados_final)-qtd_previsao

#Definir os dados do meu conjunto de treino
dados <- dados_final[1:ind_final]

#Aplicar o teste de estacionareidade no conjunto de treino
adf.test(dados)
kpss.test(dados)

#Aplicação do método da diferenciação pela primeira vez
dados_diff <- diff(dados)

#Aplicar o teste de estacionareidade novamente
adf.test(dados_diff)
kpss.test(dados_diff)

#Cálculo da média móvel
for (i in 1:qtd_previsao) {
  ultima_posicao = length(dados_diff) #atribuindo a uma variável o tamanho do conjunto de teste
  primeira_posicao = length(dados_diff)-k+1 #artribuindo a uma variável o início dos dados do conjunto de teste
  media = mean(dados_diff[primeira_posicao:ultima_posicao]) #cálculo da média entre todos os valores do conjunto de teste
  dados_diff <- c(dados_diff, media) #criação de uma nova posição no conjunto de teste e atribuindo a média calculada ao conjunto
}

#Desfazer a diferenciação da série temporal
df <- numeric(length = length(dados_final)-1)
df[1] = dados_final[1]
for (i in 1:length(df)) {
  df[i+1] <- df[i]+dados_diff[i]
}

#Atribuir o conjutno de treino a um df
dados_reais <- dados_final[(length(dados_final)-(qtd_previsao-1)):length(dados_final)]
#Atribuir os dados previstos para um df
dados_previstos <- df[(length(df)-(qtd_previsao-1)):length(df)]
soma = 0

#Cálculo dos resíduos de previsão
residuos <- numeric(length(dados_reais))
for (i in 1:length(dados_reais)) {
  residuos[i] = (dados_reais[i]-dados_previstos[i])
}

#Teste de Normalidade de Shapiro-Wilk
shapiro.test(residuos)

#Cálculos necessários para o cálculo dos intervalos de previsão

#Aplicar o cálculo da variância
for (i in 1:length(dados_reais)) {
  soma = soma + (dados_reais[i]-dados_previstos[i])
}
```

```r
x = soma / qtd_previsao
soma2 = 0
for (i in 1:length(dados_reais)) {
  soma2 = soma2 + ((dados_reais[i]-dados_previstos[i])- x)^2
}
variancia <- soma2 / (qtd_previsao-1)

#Aplicar o cálculo do desvio padrão
desvio_padrao = sqrt(variancia)

#Criação de dois vetores para armazenar os limites superiores e inferiores
inf <- numeric(length(dados_previstos))
sup <- numeric(length(dados_previstos))

#Cálculo dos limites
for (i in 1: length(dados_previstos)){
  inf[i] <- dados_previstos[i] - (1.96*(desvio_padrao/(sqrt(k)))) #cálculo do limite inferior
  sup[i] <- dados_previstos[i] + (1.96*(desvio_padrao/(sqrt(k)))) #cálculo do limite superior
  #Colocar as informações em um mesmo dataframe: previsão, limite inferior e superior
  df_final <- data.frame(dados_reais, dados_previstos, limite_inferior = inf,limite_superior = sup)
}

#Visualização dos valores reais, de previsão e os limites
print(df_final)

#Cálculo do Erro (Raiz Quadrática do Erro Médio)
rmse <- sqrt(soma2/length(dados_reais)) #calculo o rmse
print(rmse)

#Crio um vetor para auxiliar na visualização dos gráficos, e todos os dados ficarem no mesmo intervalo
v <- numeric(length = qtd_previsao)
for (i in 0:qtd_previsao){
   v[i] <- i
}

```

```r
# Plot das informações
plot(dados_reais, type = "l", col = "black", lwd = 3, xlab = "Dias",
     ylab = "Volume (hl)", ylim = c(700, 3000)) #Plot dos dados reais
lines(v, dados_previstos, col = "red", lwd = 3) #Plot dos dados previstos
lines(v, inf, col = "red", lty = "dashed") #Plot dos limites inferiores
lines(v, sup, col = "red", lty = "dashed") #Plot dos limites superiores
#Plot legenda
legend("topleft", legend = c("Dados Reais", "Dados Previstos", "Intervalo de Confiança (95%)"),
       lty = c("solid", "solid", "dashed"), col = c("black", "red", "red"))

```

APÊNDICE 2 – CÓDIGO NO R PARA APLICAÇÃO DA RNA

```r
#CHAMANDO A BIBLIOTECA QUE SERÁ UTILIZADA

library(forecast)

qtd_previsao<- 12

#Importação e visualização dos dados
df <- read.table("volume.csv", header = TRUE, sep = ";",encoding = "UTF-8")
print(df)

#Transformações nos tipos de dados
names(df)[names(df) == "Data"] <- "data"
names(df)[names(df) == "Volume"] <- "volume"
df$data <- as.Date(df$data, format = "%d/%m/%Y")
df$volume <- as.numeric(gsub(",", ".", gsub("\\.", "", df$volume)))

#Atibuir apenas os dados que serão utilizados a um novo dataframe
dados_final <- df$volume

#Definir, dentro do conjunto de dados, os índices dos dados que eu quero prever
ind_final <- length(dados_final)-qtd_previsao

#Definir os dados do meu conjunto de treino
dados <- dados_final[1:ind_final]

#Fixação de uma semente para a aleatoriedade
set.seed(3)

#Treinamento da rede neural a partir do conjunto de treino
fit<-nnetar(dados)

#Cálculo de previsão e dos intervalos de confiança com 95% de confiança
fcast2 <- forecast(fit, h=qtd_previsao, PI=TRUE, npaths=1000,level=95)

#Atribuir o conjutno de treino a um df
dados_reais <- dados_final[(length(dados_final)-(qtd_previsao-1)):length(dados_final)]

#Crio um vetor para auxiliar na visualização dos gráficos, e todos os dados ficarem no mesmo intervalo
v <- numeric(length = qtd_previsao)
for (i in 0:qtd_previsao){
  v[i] <- i
}

#Plotagem do gráfico
plot(v,dados_reais,type="l",col="black",lwd=3,xlab= "Dias",
     ylab="Volume (hL)", ylim = c(600,3500)) #Plot do intervalo dos dados reais
lines(v,fcast2$mean,lwd=3, col = "red") #Plot dos dados previstos
lines(v,fcast2$lower,col="red",lwd=1,lty=2) #Plot dos limites inferiores
lines(v,fcast2$upper,col="red",lwd=1,lty=2) #Plor dos limites superioes
#Plot legenda
legend("topright",c("Dados reais","Dados previstos","Intervalo de confiança (95%)"),
       col=c("black","red", "red"),lwd=c(3,3,1),lty=c(1,1,2))

#Cálculo do erro
erro <- accuracy(as.numeric(dados_reais), as.numeric(fcast2$mean))
print(erro)
```