Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação

Vinícius Bedeschi Costa Cunha

Uso de Aprendizado de Máquina para Especificação do Tempo de Entrega em Vendas Via *E-commerce*

Monografia de Graduação

Ouro Preto, 2023

Vinícius Bedeschi Costa Cunha

Uso de Aprendizado de Máquina para Especificação do Tempo de Entrega em Vendas Via *E-commerce*

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheira(o) de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Jadson Castro Gertrudes

Coorientadora: Profa. Dra. Adrielle de Carvalho Santana

Ouro Preto

2023

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Vinícius Bedeschi Costa Cunha

Uso de Aprendizado de Máquina para Especificação do Tempo de Entrega em Vendas Via *E-commerce*

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia de Controle e Automação

Aprovada em 21 de Julho de 2023

Membros da banca

Prof. Dr. Jadson Castro Gertrudes - Orientador (DECOM - Universidade Federal de Ouro Preto)
Profa. Dra. Adrielle de Carvalho Santana - Coorientadora (DECAT - Universidade Federal de Ouro Preto)
Natália Fernanda de Castro Meira - Convidada (PPGCC - Universidade Federal de Ouro Preto)
Me. Lauro Ângelo Gonçalves de Moraes - Convidado (DECOM - Universidade Federal de Ouro Preto)

Jadson Castro Gertrudes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em XX/07/2023

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 12/08/2023, às 10:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0559384** e o código CRC **393FED80**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos a todos que contribuíram para a conclusão deste trabalho. Primeiramente, minha gratidão vai para minha família, especialmente aos meus pais Leila e Marcelo, a minha madrasta e meu padrasto, Cristina e José Luiz, e ao meu irmão, Vitor, que estiveram ao meu lado em todos os momentos, oferecendo apoio incondicional e incentivando meu crescimento acadêmico.

Agradeço também aos meus amigos, cuja presença e encorajamento foram fundamentais durante essa jornada. Suas palavras de estímulo e compreensão foram um verdadeiro suporte emocional. Não posso deixar de mencionar minha república em Ouro Preto, a Nostravamus, que se tornou meu lar durante essa etapa da minha vida. A convivência com meus colegas de república foi enriquecedora e divertida, tornando essa experiência acadêmica ainda mais memorável.

Agradeço especialmente à minha namorada, Mia, cujo amor, paciência e apoio constante foram essenciais para que eu enfrentasse os desafios do trabalho e me mantivesse motivado ao longo do caminho. Sua presença trouxe equilíbrio e felicidade à minha vida, e sou profundamente grato por tê-la ao meu lado.

Agradeço especialmente aos meus orientadores, Adrielle e Jadson, cuja orientação e expertise foram cruciais para moldar este trabalho da melhor forma possível. Todos vocês desempenharam um papel significativo no meu crescimento pessoal e no sucesso deste trabalho.

Resumo

O aumento nas compras por meios eletrônicos (*e-commerce*) tem crescido drasticamente, especialmente nos últimos dois anos, devido a pandemia do novo coronavírus. O aumento no número de vendas também acarretou o atraso nas entregas de produtos aos consumidores, o que resulta em prejuízos tanto para a empresa quanto para o cliente. Por isso, é muito importante a estimativa correta de entrega de um produto, afim de evitar transtornos entre fornecedores e clientes. Neste trabalho é apresentado uma análise exploratória de dados relacionados a vendas via *e-commerce* com o proposito de estabelecer a estimativa do prazo de entrega de uma compra. Para isso utilizou-se um algoritmo de rede neural artificial e um de *random forest*, para se tentar vincular os dados de uma compra com seu possível período para entrega. Ambos os modelos obtiveram resultados promissores, especialmente o de *random forest*, demonstrando uma eficácia significativamente melhor na estimativa do prazo de entrega em comparação com as técnicas atualmente utilizadas pelas empresas do banco de dados em questão.

Palavras-chaves: *E-commerce*, aprendizado de máquina, inteligência artificial, rede neural, *Random Forest*.

Abstract

The increase in electronic commerce (e-commerce) purchases has grown significantly, especially in the past two years, due to the COVID-19 pandemic. The rise in sales has also led to delays in product deliveries to consumers, resulting in losses for both companies and customers. Therefore, accurate estimation of product delivery is crucial to prevent disruptions between suppliers and clients. This research presents an exploratory analysis of data related to e-commerce sales with the aim of establishing the estimated delivery time of a purchase. To achieve this, an artificial neural network algorithm and a random forest algorithm were used to link purchase data with their potential delivery period. Both models yielded promising results, particularly the random forest model, demonstrating significantly better accuracy in estimating delivery times compared to the techniques currently employed by the companies in the database under analysis.

Key-words: E-commerce, machine learning, artificial intelligence, neural network, ramdon forest.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Justificativas e Relevância

O termo *e-commerce* é uma abreviação de *eletronic commerce* que, traduzido do inglês, significa comércio eletrônico. De acordo com Mendonça (2016) esse termo se refere a diferentes tipos de comércios, tais como bens, serviços, entre outros. Ele se caracteriza por fazer a ligação entre o mundo virtual e o real e pelas transações realizadas eletronicamente. Na década de 1970 ocorreram as primeiras EDI (*Eletronic Data Interchange*) e EFT (*Eletronic Funds Transfer*), que consistem em transferências eletrônicas de documentos e de fundos, respectivamente. Na época, essas transações eram utilizadas apenas para o setor bancário. O termo *e-commerce* surgiu em 1979, criado pelo inglês Michael Aldrich (1941-2014), que na época era funcionário da empresa Rediffusion Computers (Reino Unido).

A chegada da *internet* possibilitou que essa forma de venda se espalhasse, trazendo vantagens, como a possibilidade de pequenos comerciantes atenderem uma maior quantidade de pessoas, uma vez que conseguem servir clientes perto e longe de sua localidade. O faturamento e o volume de vendas realizados por meios digitais sofreram crescimento desde do advento da internet e, com as restrições de deslocamento causadas pela pandemia do coronavírus, essa evolução sofreu um aumento vertiginoso. Como pode ser observado na Figura 1, o crescimento do faturamento gerado pelo *e-commerce* no primeiro semestre de 2020 cresceu 47% em relação ao primeiro semestre de 2019. Ainda de acordo com a mesma Figura, esse valor já vinha sofrendo um crescimento constante desde de 2001.

A satisfação do cliente é um dos fatores que podem tem grande influência no sucesso de um negócio. Com isso, serviços relacionados a vendas *online*, como a previsão de entrega dos produtos, ganharam grande importância, podendo refletir o sucesso de empresas, uma vez que, de acordo com Giacomel, Cardoso e Espírito Santo (2019), esses serviços são um dos principais elementos responsáveis por gerar boas ou más impressões de clientes sobre os comércios.

A alta no volume de vendas *online* faz com que uma boa estruturação de logística de entrega passe a ser essencial para o setor de varejo *online*. Empresas com logísticas bem organizadas conseguem prever com maior antecedência quando uma entrega vai atrasar, o que, de acordo com Branquinho Filho (2020), as leva a gastarem menos em transporte, economizarem dinheiro e pouparem esforços. Ainda de acordo com o mesmo estudo de

[1] https://www.meioemensagem.com.br/home/marketing/2020/08/27/e-commerce-cresce-47-maior-alta-em-20-anos.html

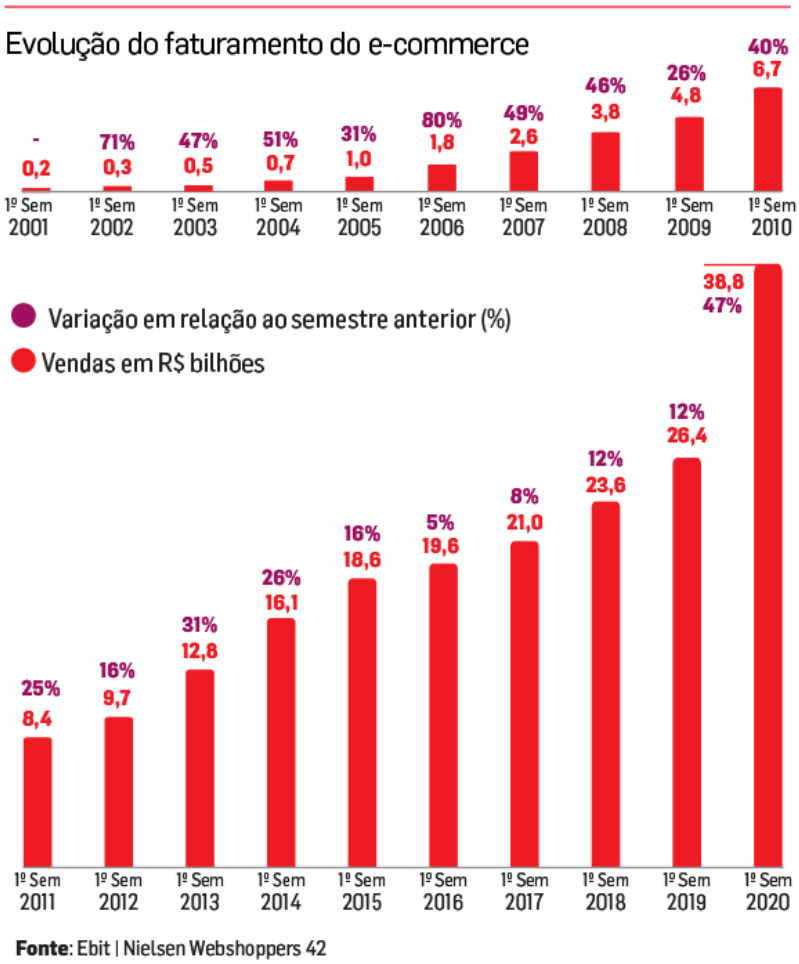


Figura 1 – Evolução no faturamento do e-commerce

Fonte: Schnaider A. *apud* Ebit Nielsen (2020)[1].

Branquinho Filho (2020), um dos elementos que mais gera insatisfação em vendas do *e-commerce* são os atrasos nas entregas. Uma boa organização do sistema de transporte também diminui as ocorrências de imprevistos nos prazos de entrega, o que gera clientes mais satisfeitos, que podem se tornarem promotores da marca.

As técnicas de aprendizado de máquinas já estão afetando o *e-commerce* de diversas maneiras. No trabalho de Branquinho Filho (2020), pode ser observado como essas técnicas obtiveram sucesso em prever quando uma entrega vai atrasar, por meio de algoritmos de regressão. No presente trabalho foram criados dois modelos de aprendizado de máquina com o objetivo de prever com eficácia o prazo de entrega de mercadorias compradas *online*. Para o desenvolvimento dos modelos será utilizada a base de dados pública *"Brazilian E-Commerce Public Dataset"*, disponível na plataforma Kaggle[2].

[2]https://www.kaggle.com/olistbr/brazilian-ecommerce

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é avaliar dois modelos de aprendizado de máquina, em particular os modelos de rede neural artificial e de *Random Forest*, a fim de prever com maior eficácia o tempo de entrega de mercadorias para vendas em lojas *online* na região sudeste do Brasil.

1.2.2 Objetivos Especificos

(1) Identificar quais atributos da base "*Brazilian E-Commerce Public Dataset*" afetam no tempo de entrega de um produto.

(2) Analisar a influência de diferentes atributos, tais como distância a ser percorrida na entrega, dimensões físicas do produto, localização geográfica do comprador e do vendedor, data da compra, entre outras, no tempo de entrega.

(3) Implementar e comparar modelos de aprendizado de máquina capazes de prever o tempo de entrega de mercadorias com base nos dados selecionados.

1.3 Metodologia

A base de dados a ser utilizada é a *Brazilian E-Commerce Public Dataset*, disponibilizada pela plataforma digital de comércio eletrônico Olist. esta base possui informações sobre mais de 100 mil entregas no Brasil, de diversas lojas, entre os anos de 2016 e 2018. A base apresenta diversas informações como *status* do pedido, preço, pagamento, dimensões do produto, previsão de entrega, data em que o produto foi entregue, geolocalização relacionada aos códigos postais do vendedor e do comprador, entre outros, distribuídos em 9 tabelas. Os dados disponibilizados pertencem a empresas reais, no entanto, foram preservadas as suas identidades, uma vez que os nomes foram substituídos por números de identificação únicos, garantindo assim a privacidade das organizações envolvidas na pesquisa.

A partir dos dados disponíveis, foram realizados cálculos e extraídas informações relevantes, as quais foram posteriormente processadas e selecionadas de acordo com a pertinência ao contexto da pesquisa em questão. Por meio dos registros de data de confirmação da compra e data de entrega do produto, foram calculados os tempos de entrega reais dos produtos. Utilizando as informações de geolocalização dos compradores e vendedores, foi possível calcular a distância percorrida pela mercadoria, que é uma das variáveis de maior relevância no treinamento do modelo, uma vez que sua distância impacta diretamente no tempo necessário para a entrega. A partir da data de confirmação do pedido,

foi extraída a informação do mês do ano, fator relevante a ser considerado, uma vez que existem datas importantes ao longo do ano, como o Natal e a *Black Friday*, que impactam substancialmente nas vendas e refletem em um aumento significativo no volume de entregas em todo o país. Isso implica em um aumento nos tempos de entrega nesse período, o que torna essa informação valiosa para a previsão de entrega de produtos. Também foi realizada a extração do dia da semana em que ocorreu a confirmação do pedido, uma vez que a disponibilidade de algumas distribuidoras pode variar nos finais de semana e há dias da semana em que o tráfego é mais intenso, fatores que podem influenciar o processo de entrega da mercadoria.

Para o treinamento dos modelos com o objetivo de prever os tempos de entrega dos produtos, as informações utilizadas foram: preço da mercadoria, valor do frete, distância entre a cidade do vendedor e do comprador, o mês do ano e o dia da semana da confirmação da compra e os dados de dimensão do produto (altura, largura, profundidade e peso), uma vez que mercadorias com maiores dimensões e peso podem gerar complicações no transporte, levando a atrasos nos tempos de entrega. O valor alvo dos modelos de aprendizado de máquina foi definido como o tempo real de entrega do produto, que consiste na variável a ser prevista. Os dados das previsões de entrega feitas pelos vendedores, que estão presentes na base de dados utilizada, são comparados com as previsões feitas pelos modelos, para avaliar a sua eficacia.

Após realizar uma análise cuidadosa e tratamento adequado da base de dados, incluindo a extração de informações relevantes que não estavam explícitas nos dados brutos e seleção criteriosa dos dados a serem utilizados, foi conduzido um estudo aprofundado acerca dos modelos de regressão e do funcionamento dos dois métodos de aprendizado de máquina escolhidos, a Rede Neural Artificial e a *Random Forest*. Com o objetivo de obter a melhor eficiência de cada método, os dados foram formatados adequadamente para estarem em conformidade com os requisitos dos modelos, de modo a possibilitar previsões precisas das datas de entrega do *e-commerce*.

Os códigos contendo a estruturação[3] da base de dados, os modelos de Rede Neural[4] e de Árvore Aleatória[5] estão disponíveis no *github*.

1.4 Organização

O presente trabalho está organizado da seguinte forma:

- Capítulo 1: Introdução apresentando justificativa e relevância do assunto, a meto-

[3] https://github.com/viniciusbedeschi/TCC/blob/main/TCC_Estruturacao_Dados_Final.ipynb

[4] https://github.com/viniciusbedeschi/TCC/blob/main/TCC_Rede_Neural_Final.ipynb

[5] https://github.com/viniciusbedeschi/TCC/blob/main/TCC_Random_Forest_Final.ipynb

dologia e os objetivos do trabalho;

- Capítulo 2: Apresentação da fundamentação teórica do conteúdo abordado;
- Capítulo 3: Desenvolvimento do trabalho;
- Capítulo 4: Apresentação dos resultados e discussão
- Capítulo 5: Conclusões e sugestões para a continuação desse trabalho

2 Fundamentação Teórica

2.1 Aprendizado de Máquina

O aprendizado de máquina, ou *machine learning* (ML), é um subcampo da inteligência artificial que se concentra no desenvolvimento de sistemas, técnicas e algoritmos que permitam que os computadores possam aprender a partir de dados (GÉRON, 2022). Ao contrário da programação convencional, em que um programador escreve um conjunto de instruções para o computador seguir para concluir uma tarefa, o aprendizado de máquina permite que o computador aprenda a reconhecer padrões subjacentes aos dados com base em exemplos e experiências anteriores. O objetivo é fazer com que a máquina seja capaz de reconhecer padrões e tomar decisões com base neles, sem a necessidade de intervenção humana. Isso torna o aprendizado de máquina uma ferramenta poderosa, principalmente para lidar com grandes quantidades de dados e problemas complexos em áreas, como previsão e análise de dados, visão computacional, processamento de linguagem natural e tomadas de decisão automatizada.

Os problemas de aprendizado de máquina são divididos em quatro categorias com base nos dados. No aprendizado supervisionado o algoritmo já inclui o conjunto de soluções desejado, assim o objetivo do modelo é encontrar um padrão ou uma relação entre as variáveis de entrada e de saída do modelo. De acordo com Quintino et al. (2020), as duas tarefas principais desses algoritmos são regressão e classificação. No aprendizado não-supervisionado os dados fornecidos não são rotulados, e o objetivo do modelo é encontrar padrões no conjunto fornecido. Ainda de acordo com Géron (2022), as principais tarefas desse algoritmos são clusterização, detecção de anomalias, redução de dimensionalidade e análise de associação. Existem também o aprendizado semissupervisionado, em que o algoritmo geralmente trata com muitos dados não rotulados e uma quantidade pequena de dados rotulados. Por último, no aprendizado por reforço, o algoritmo é treinado por meio de tentativa e erro, no qual o modelo aprende a partir de recompensas ou penalidades. No presente trabalho, o foco será a utilização de algoritmos de ML supervisionados

2.2 Regressão

Regressão é uma das principais técnicas utilizadas no campo de aprendizado de máquina para construção de modelos preditivos. De acordo com Goodfellow, Bengio e Courville (2016), no livro *Deep learning*, é um tipo de tarefa em que "o programa de computador é solicitado a prever um valor numérico com base em alguma entrada". É um método paramétrico que busca estabelecer uma função matemática que melhor descreva a relação

entre as variáveis, com base em um conjunto de dados. A Figura 2 ilustra o gráfico de uma regressão linear, no qual cada ponto representa um dado de entrada, e o erro é determinado pela distância desses pontos em relação à reta de regressão a qual modela a relação $y_i = f(x_i)$.

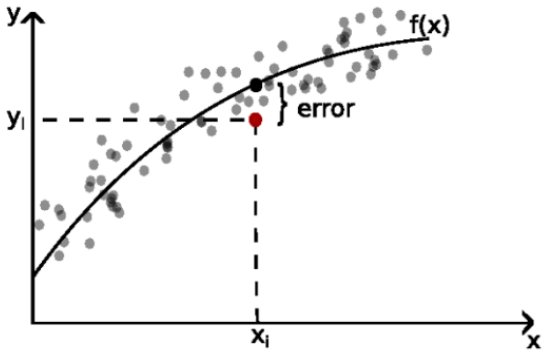


Figura 2 – Regressão Linear

Fonte: Santana (2020)

2.2.1 Métricas de Avaliação em Modelos de Regressão

As métricas de avaliação do modelo de regressão são ferramentas importantes para avaliar sua eficácia e qualidade. Essas métricas fornecem uma medida numérica de quão bem um modelo pode prever as saídas corretas para um conjunto de dados de entrada. Algumas das métricas mais comuns são: erro médio absoluto (MAE), erro quadrático médio (MSE), raiz do erro quadrático médio (RMSE), coeficiente de determinação (R^2), erro quadrático médio Logaritmo (RMSLE) e erro percentual médio absoluto (MAPE). A métrica de avaliação do modelo de regressão é usada para selecionar a melhor solução entre várias alternativas e otimizar o algorítimo selecionado, ajustando seus parâmetros de acordo com o conjunto de dados relevantes. Em suma, as métricas de avaliação do modelo de regressão permitem que diferentes modelos sejam avaliados, comparados e refinados para melhorar sua eficácia e qualidade na previsão de resultados em novos conjuntos de dados. No presente trabalho são utilizadas as métricas de erro médio absoluto, erro quadrático médio e erro percentual absoluto afim de analisar os modelos aprendizado de máquina criados.

- **MAE - Erro Médio Absoluto**

  O erro médio absoluto é uma métrica amplamente utilizada na avaliação de modelos de regressão, conforme discutido no livro *Forecasting: Principles and Practice*, de Hyndman e Athanasopoulos (2018). Essa métrica calcula a média das diferenças

entre as previsões geradas pelo modelo e os dados reais, tendo o seu valor calculado a partir da soma das diferenças absolutas entre cada previsão do modelo e a saída correta e, em seguida, dividindo-se pelo número de amostras no conjunto de dados teste, como na formula a seguir:

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

Onde:

- n é o número de amostras;
- y_i é o valor real do i-ésimo exemplo;
- $\hat{y}_i$ é a predição do modelo para o i-ésimo exemplo.

Essa métrica trata-se de uma medida direta da eficácia do modelo, em que quanto menor o valor da saída, maior a eficácia. É uma métrica extremamente fácil de interpretar, uma vez que está na mesma unidade da variável de saída. O MAE é menos sensível a valores discrepantes (*outliers*) do que outras formas de avaliação, como o erro médio quadrático.

- **MSE - Erro Médio Quadrático**

  O erro médio quadrático é outra métrica mais utilizada para avaliar a qualidade de modelos de regressão, de acordo com o livro de *An introduction to statistical learning*, de James et al. (2013). O MSE é calculado a partir do quadrado das somas das diferenças entre as previsões do modelo e os valores reais observados, como na fórmula a seguir:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

  Onde:

  - n é o número de amostras;
  - y_i é o valor real do i-ésimo exemplo;
  - $\hat{y}_i$ é a predição do modelo para o i-ésimo exemplo.

  Uma vez que os erros são elevados ao quadrado, os valores extremos, como *outliers*, afetam mais o resultado final. A saída da métrica é uma medida numérica do quanto o modelo está errado em média, em que, como no MAE, quanto menor o valor, mais preciso é o modelo.

- **MAPE - Erro Médio Percentual Absoluto**

  O erro médio percentual absoluto é uma técnica utilizada para avaliar modelos de regressão em termos de porcentagem de erro médio absoluto. Essa métrica calcula a diferença absoluta entre cada valor previsto e o valor correspondente observado. Em seguida, essas diferenças absolutas são divididas pelos valores observados e multiplicadas por 100 para obter a porcentagem de erro absoluto para cada observação (HYNDMAN; ATHANASOPOULOS, 2018).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|Y_i - \hat{Y}_i\right|}{Y_i}\right) \times 100 \tag{2.2}$$

  Onde:

  - n é o número de amostras;
  - y_i é o valor real do i-ésimo exemplo;
  - $\hat{y_i}$ é a predição do modelo para o i-ésimo exemplo.

  Assim o valor final é obtido calculando a média das porcentagens de erro absoluto para todas as observações como na fórmula a seguir. Quanto menor o valor do MAPE, melhor a eficácia do modelo de regressão, pois indica que as previsões estão mais próximas dos valores reais.

2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são um conjunto de modelos computacionais que consistem em unidades de processamento interconectadas, chamadas de neurônios artificiais, que são capazes de aprender com um conjunto de dados e ajustar seus pesos sinápticos para melhorar a eficácia do modelo. Essas redes são constituídas por uma arquitetura formada por camadas de neurônios. A camada de entrada é responsável por receber os dados que serão processados. As camadas intermediárias são denominadas ocultas, sendo responsáveis pelo processamento dos dados de forma a gerar uma saída. A última camada da rede é a de saída, responsável por gerar a resposta final do modelo. De acordo com Goodfellow, Bengio e Courville (2016), cada camada oculta da rede neural é composta por vetores de valores, sendo que a dimensão dessas camadas que determina a largura do modelo, e cada elemento desses vetores, pode ser entendido como tendo uma função semelhante à de um neurônio. A organização das camadas é ilustrada na Figura 3.

No trabalho acadêmico "Avaliação automática da utilidade de *reviews* usando Redes Neurais Artificiais no *corpus* do *Steam*", realizado por Andrade Silva e Feitosa (2021),

[1] https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0

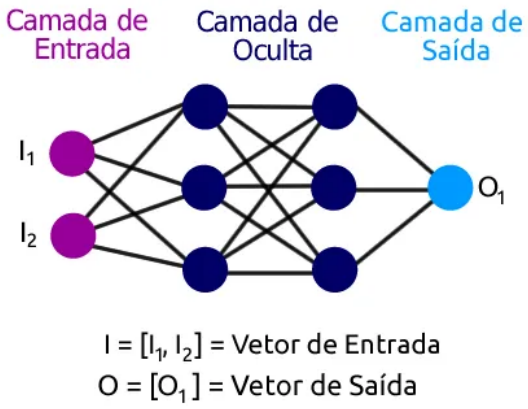


Figura 3 – Rede Neural Artificial

Fonte: Medium[1]

foram obtidos resultados promissores por meio da aplicação de uma rede neural artificial de multicamadas em dados de *reviews* em sites de *e-commerce*. O objetivo foi identificar as características que mais impactam a percepção dos usuários sobre os comentários nas páginas.

2.3.1 Rede Neural de Propagação Direta

As Redes Neurais de Propagação Direta (*Feedforward Neural Networks*) são uma etapa essencial de modelo de aprendizado profundo (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse modelo é chamado de propagação direta pois ele consiste em múltiplas camadas de neurônios que processam informações em um sentido direto, sem conexões de *feedback*. As redes de propagação direta são de extrema importância para as aplicações de aprendizado de máquina.

2.3.2 Treinamento da Rede Neural

O treinamento da rede neural é uma etapa essencial para o desenvolvimento de modelos de aprendizado de máquina. Durante o processo de treinamento, a rede neural é exposta aos dados de entrada e saída disponíveis, relativos ao problema em questão, de modo que ela aprenda a reconhecer padrões e realizar previsões por meio do mapeamento das relações entre esses dados. Antes do treinamento, é necessário configurar os hiperparâmetros da rede. Os hiperparâmetros são elementos externos que influenciam o comportamento e desempenho do modelo de aprendizado de máquina, eles são usados para regular a complexidade do modelo e ajustar sua capacidade de generalização. De acordo com Reis (2021), os hiperparâmetros podem ser entendidos como as configurações de um modelo, e o ajuste adequado deles é crucial para otimizar o seu desempenho. A seguir, são descritos

os principais hiperparâmetros utilizados na construção de uma rede neural de propagação direta.

2.3.2.1 Batch Size

O *batch size*, ou tamanho do lote, define a quantidade de amostras de treinamento que serão usadas em cada iteração do algoritmo. Quanto maior o lote, mais rápido é o treinamento, pois é processado uma maior quantidade de dados ao mesmo tempo. Valores de *Batch sizes* menores podem tornar o treinamento mais preciso nos dados de treinamento, pois permite que o modelo se adapte melhor aos dados de treinamento, evitando com que o modelo se ajuste demais, e acabe não sendo capaz de generalizar bem para novos dados, problema conhecido como *overfiting*.

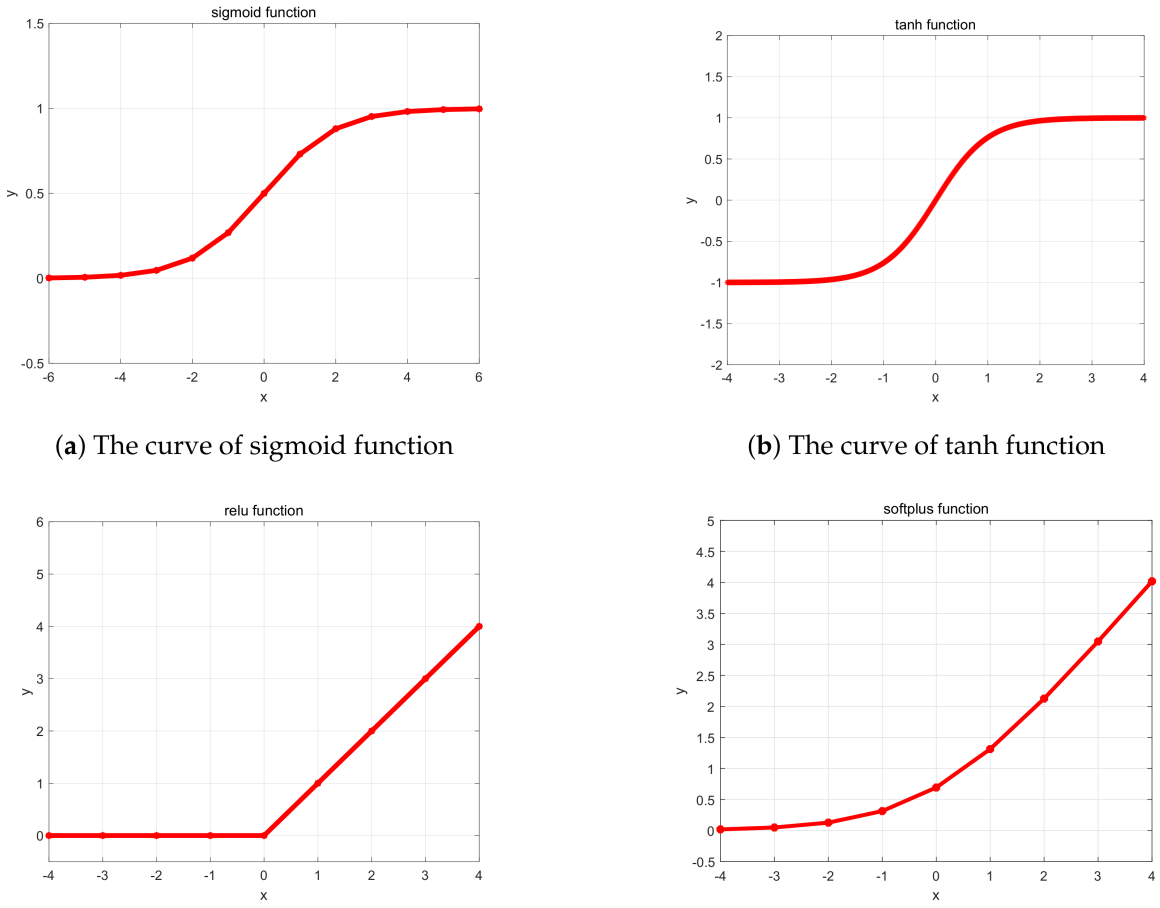
2.3.2.2 Taxa de aprendizado

A taxa de aprendizado, também conhecida como *learning rate*, é um dos hiperparâmetros mais relevantes no treinamento de redes neurais. Seu valor define o tamanho do passo que o algoritmo de otimização deve dar em cada atualização dos pesos da rede. É crucial encontrar um equilíbrio adequado na escolha desse parâmetro, pois valores muito pequenos tornam o processo de treinamento muito lento, enquanto valores muito grandes podem levar à instabilidade do algoritmo, ocasionando divergência ou oscilações em torno de uma solução ótima local. Para contornar esses problemas, é comum ajustar a taxa de aprendizado ao longo do processo de treinamento.

2.3.2.3 Função de Ativação

A função de ativação é uma parte essencial da rede neural, responsável por produzir as saídas de um neurônio, que serão passadas para o próximo neurônio, de acordo com as entradas. Ela é de extrema importância pois permite com que a rede aprenda relações não-lineares entre os dados. Sem ela a rede ficaria limitada a realizar apenas operações lineares. Existem diversas funções de ativação que podem ser utilizadas, mas de acordo com Goodfellow, Bengio e Courville (2016) atualmente, em redes neurais, a recomendação padrão é utilizar a *Rerectified linear unit* (ReLu), que produz uma saída de 0 para entradas negativas e a própria entrada para valores positivos. Ela geralmente é uma escolha popular devido à sua simplicidade e capacidade de treinar redes neurais profundas. Outra função, comumente utilizada, é a *sigmoid*, que é amplamente aplicada à problemas de classificação. Ela mapeia qualquer valor real para um intervalo entre 0 e 1, sendo que entradas positivas se aproximam de 1, e entradas negativas se aproximam de 0. A função de tangente hiperbólica(tanh) é centrada em 0, e varia sua saída entre -1 e 1, e é amplamente utilizada em problemas de regressão e classificação binária. Por último vale a pena mencionar a função de ativação *softplus*, que é uma versão suavizada da ReLU, sendo di-

ferenciável em todos os pontos, é uma função crescente que retorna valores positivos para todas as entradas. Na Figura 4 é possível observar o gráfico das quatro funções citadas.



(**a**) The curve of sigmoid function

(**b**) The curve of tanh function

(**c**) The curve of ReLu function

(**d**) The curve of softplus function

Figura 4 – Funções de Ativação

Fonte: Wang et al. (2020).

2.3.2.4 Camadas

Nas redes neurais podem ser alteradas tanto a quantidade de camadas como o tamanho de cada uma. O número de camadas, ou *layers*, vai definir a profundidade da rede, ou seja, quantas camadas de neurônios ela vai possuir. Redes mais profundas são capazes de aprender recursos mais complexos, como capturas características mais sutis nos dados. Mas um número muito grande de camadas pode dificultar o treinamento, pois aumenta a complexabilidade do modelo, podendo levá-lo a se ajustar excessivamente aos dados de treinamento, fazendo com que ele não seja capaz de generalizar bem para novos dados. O tamanho da camada se refere ao número de neurônios que ela possui. Ele também deve ser ajustado para encontrar um equilíbrio entre a capacidade do modelo e a generalização.

2.3.2.5 Número de Épocas

O número de épocas define a quantidade de vezes que o algoritmo de treinamento processa todo o conjunto de dados de treinamento. Definir um número pequeno de épocas pode gerar um modelo não treinado o suficiente, com baixo desempenho para novos dados. Por outro lado, um número muito grande pode levar ao *overfiting*, explicado anteriormente. Para contornar esse problema é comum utilizar técnicas para interromper o treinamento quando não há mais melhora no desempenho do modelo em um conjunto de validação, como a *EarlyStopping* e a *ReduceLROnPlateau*.

2.3.2.6 Algoritmo de Otimização

O algoritmo de otimização é responsável por ajustar os parâmetros da rede de forma a minimizar a função de perda, que é a diferença entre a saída da rede neural e o valor esperado. A otimização adequada dos pesos é crucial para que a rede seja capaz de produzir resultados precisos e confiáveis. Dessa forma, a seleção correta do algoritmo de otimização é essencial para garantir uma convergência eficiente e estável do modelo. Um dos algoritmos mais utilizado é o *Gradient Descent*, que atualiza os pesos na direção oposta do gradiente da função de perda, tentando diminuir a discrepância entre a saída da rede e o valor correto. Outro algoritmo muito utilizado é o *Adaptive Moment Estimation* (Adam), que é baseado em gradientes adaptativos. Ele usa da estimativa de momento de primeira ordem (média móvel dos gradientes) e o momento de segunda ordem (média móvel ponderada dos gradientes quadrados recentes), para ajustar dinamicamente a taxa de aprendizado de cada peso da rede. De acordo com Kingma e Ba (2014), o Adam é um algoritmo robusto e que se adapta bem a uma grande variedade de problemas no campo de *Machine Learning*.

2.4 *Random Forest*

A *Random Forest* (*Random Forest*) é um algoritmo de aprendizado de máquina, baseado em um conjunto de árvores de decisão (*Decision Tree*), normalmente utilizado para realizar tarefas de regressão e classificação. As árvores de decisão são modelos preditivos baseados em regras de decisão hierárquicas. O seu processo de construção se baseia em um nó com todo o conjunto de dados de treinamento. A partir dele é escolhida a variável de decisão que melhor divide os dados para formar a primeira ramificação da árvore, e esse processo se repete até que sejam criados nós terminais, que representam as classes ou valores de saída da árvore. De acordo com Chollet (2021) a *Decision Tree* possui a vantagem de ser um modelo de fácil visualização e análise.

O funcionamento do *Random Forest* se baseia na criação de uma grande quantidade de árvores de decisão, em que cada uma é treinada com uma amostra aleatória dos dados.

Cada árvore gera um conjunto de resultados, e o valor final da predição é determinado pela média ou votação dos resultados de cada uma.

Ainda de acordo com Chollet (2021), a *Random Forest* é uma forma robusta e prática para utilizar árvore de decisão, e na maioria dos casos gera um bom resultado para qualquer tarefa de aprendizado de máquina. Maman (2017) conduziu um estudo com objetivo de classificar usuários testes em um *software* de mercado financeiro. O autor testou diferentes modelos e, após análise comparativa de desempenho, verificou que o algoritmo de *Random Forest* foi o que apresentou melhor desempenho para prever a propensão dos usuários à assinarem o *software* após testá-lo.

2.4.1 Treinamento da *Random Forest*

O treinamento de uma *random forest*, assim como na Rede Neural, é uma das etapas mais importantes para o desenvolvimento do modelo. Nesse processo o modelo aprende com os dados de treinamento, e ajusta seus parâmetros de forma a otimizar a eficácia na previsão de novos dados. A seguir são explicados os principais hiperparâmetros que podem ser ajustados na *Random Forest* com objetivo de melhorar o desempenho do modelo.

2.4.1.1 N Estimators

O hiperparâmetro *n estimators* define qual será o número de árvores a serem criadas no modelo. Como a previsão final é obtida por meio da média das previsões individuais de cada árvore, aumentar o valor do *n estimators* pode aumentar a robustez do modelo, tornando-o menos suscetível a variações aleatórias dos dados de treinamento. Porém, colocar um número muito grande de árvores no modelo pode aumentar o tempo de treinamento, uma vez que aumenta sua complexidade, também podendo levar o modelo ao sobreajuste (*overfitting*) o conjunto de treinamento.

2.4.1.2 Max Depth

O *Max Deph* define a profundidade máxima da árvore de decisão, o que pode afetar significativamente o desempenho do modelo. A profundidade do modelo define quantos nós de decisão são criados, ou seja, quantas vezes os dados serão divididos. Valores muito pequenos para o *Max Deph* podem resultar em uma árvore subajustada, que não é capaz de capturar a complexidade do relacionamento entre as variáveis de entrada e a de saída, fazendo suposições simplistas dos dados, o que pode resultar em baixa eficácia. Valores muito altos possibilitam o modelo compreender melhor a relação entre os dados, porém, definir um número muito grande para esse hiperparâmetro pode gerar o problema do *overfitting*.

2.4.1.3 Min Samples Split

O hiperparâmetro *Min Samples Split* é responsável por delimitar o número mínimo de amostras que são necessárias para dividir um nó interno em dois filhos, durante o processo de construção da árvore. Caso o número de amostras seja menor do que o definido, a divisão não ocorre, e aquele nó se torna um nó folha, ou seja, a extremidade da árvore, onde a decisão é tomada. O uso desse hiperparâmetro de forma adequada para o conjunto de dados pode evitar a superajustagem do modelo, impedindo que a árvore se torne muito complexa ou profunda.

2.4.1.4 Min Samples Leaf

O *Min Samples Leaf* define a quantidade mínima de amostras que devem ser alocadas em um último nó (nó folha) da *decision tree*. A definição apropriada desse hiperparâmetro pode ajudar a regularizar a árvore, evitando o sobreajuste, da mesma forma que o *Min Samples Split*. Valores altos de *Min Samples Leaf* podem gerar nós folha maiores, reduzindo as divisões e a profundidade da árvore, deixando ela mais simples e menos sucessível a ruídos e padrões aleatórios dos dados de treinamento, mas por outro lado essa simplicidade pode fazer com que o modelo não consiga assimilar a complexidade dos padrões nos dados de treinamento.

2.4.1.5 Max Features

O hiperparâmetro *Max Features* regula o número máximo de recursos que são considerados em cada divisão de nó durante a construção do modelo. A determinação do valor adequado para o *Max Features* depende de características dos dados e do problema em questão. Como nos hiperparâmetros anteriores, esse valor influencia principalmente na capacidade de ajuste e generalização do modelo, valores muito altos podem levar ao sobreajuste, deixando-o com pouca capacidade de generalizar para novos dados e valores muito pequenos podem levar a uma árvore subajustada, que não consegue capturar as relações relevantes entre as variáveis independentes.

2.5 Validação Cruzada

A validação cruzada é uma estratégia usada principalmente quando o conjunto de dados disponíveis para treinar um modelo não é muito grande. De acordo com Goodfellow, Bengio e Courville (2016) uma pequena amostra pode implicar em imprecisões estatísticas em torno do erro médio de teste estimado. Como forma de contornar essa questão essa técnica se baseia na ideia de repetir o cálculo de treinamento e teste em diferentes subconjuntos ou divisões aleatórias do conjunto de dados original. A validação desse tipo mais utilizada é a *k-fold*, em que o conjunto de dados é dividido e k subconjuntos exclusivos, cada um

com o mesmo número de amostras. Em seguida, o modelo é treinado em $k-1$ partições e testado na partição restante. Esse processo é repetido k vezes, onde cada partição é usada exatamente uma vez como conjunto de teste, conforme representado na Figura 5.

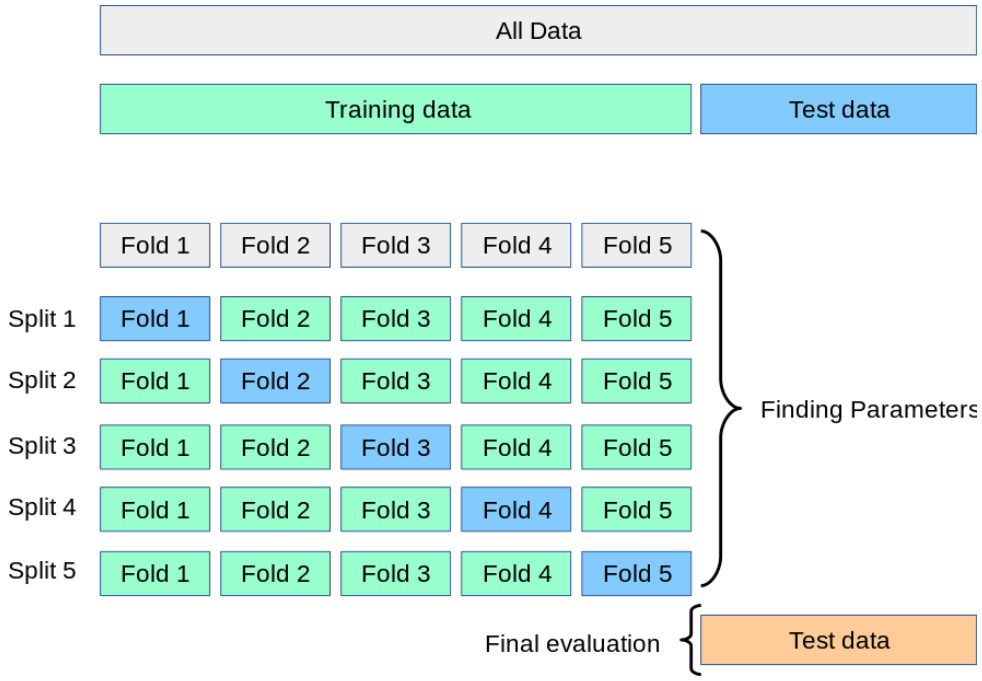


Figura 5 – Validação Cruzada Utilizando K-fold

Fonte: scikitlearn[2]

A medida de desempenho relatada pela validação cruzada *k-fold* é, então, a média dos valores calculados no *loop*. Esse método é mais custoso do ponto de vista computacional, mas evita com que sejam perdidos dados de treinamento para o conjunto de teste, representando uma vantagem significativa para conjuntos de dados pequenos.

2.6 Aplicação de Técnicas de Aprendizado de Máquina em Logísticas de Entrega

O emprego de técnicas de aprendizado de máquina tem sido amplamente utilizado em diversas áreas, sobretudo por meio da regressão, para realização de previsões. Branquinho Filho (2020) realizou uma análise sobre a importância de prever atrasos em entregas, com o objetivo de economizar recursos e dinheiro para as empresas. Para isso, conduziu um estudo que buscou prever atrasos em mercadorias, identificando os critérios mais relevantes nessa determinação. O autor utilizou o AutoML (*Auto Machine Learning*), uma função capaz de encontrar o melhor modelo de *machine learning* para o conjunto de dados, sem necessidade de conhecimentos prévios sobre o conjunto de dados. Para versão

[2]https://scikitlearn.org/stable/modules/cross_validation.html

usada no trabalho, o AutoML comparou um padrão para *random forest*, um *grid* para o *Gradient Boosting Machines* (GBMs), um *grid* aleatório para *Deep Neural Nets*, uma *grid* de GLMs, e dois modelos de *Stacked Ensemble*. A técnica xGBoost apresentou o melhor desempenho na classificação binária para prever se uma entrega irá ou não atrasar. "A eficácia dos modelos, DRF, xGBoost e GBM demonstram que os algoritmos com regressão entregam resultados mais robustos"(BRANQUINHO FILHO, 2020). No trabalho de Araujo e Etemad (2021), os autores utilizam redes neurais baseadas em convolução para prever tempo de entrega de encomendas, especificamente para a fase final do processo de entrega. Foram utilizados 3 categorias de redes neurais artificiais: arquiteturas VGG, redes neurais residuais (ResNet) e operadores de *squeeze and excitation* (SE). Os modelos superaram várias linhas de base frequentemente utilizadas, como as soluções de origem destino, que é um método utilizado para prever ou o tempo de viagem entre um ponto de origem e um ponto de destino. O resultado promissor demostra o potencial do aprendizado de máquina para melhorar a eficácia na previsão do tempo de entrega no *e-commerce*.

3 Desenvolvimento

3.1 Estruturação dos Dados

A base de dados utilizada no presente trabalho está disponível no site da *Kaggle*, uma plataforma amplamente conhecida por disponibilizar conjuntos de dados para fins de análise e pesquisa. Os dados em questão são referentes a vendas de produtos de empresas cadastradas na *Olist Store*, uma estrutura que atua como intermediário entre compradores e vendedores, sendo um dos maiores *marketplaces* brasileiros. As informações estão disponibilizadas em nove arquivos, referentes as diferentes dimensões relacionadas ao contexto das vendas realizadas na plataforma.

A transformação e processamento dos dados foram realizados no ambiente de desenvolvimento baseado em nuvem *Google Colaboratory*, que permite aos usuários escrever, executar e compartilhar códigos na linguagem de programação Python. Para utilizar os dados disponibilizados no site da *Kaggle* foi necessário baixar os nove arquivos em formato CSV, e salvá-los no *Google Drive*. Com os arquivos salvos no *Drive* foi possível criar tabelas no código em formato de *dataframes*, utilizando a biblioteca Pandas do Python. Por meio das colunas em comum foi possível unir todas as informações necessárias para as análises e treinamento dos modelos de aprendizado de máquina em apenas um *dataframe* final.

Dado que o objetivo deste estudo é prever o tempo de entrega das mercadorias, criou-se uma nova coluna no conjunto de dados com o cálculo dos dias transcorridos entre a compra e a entrega do produto ao cliente, visando obter o tempo real de entrega efetuado pelo vendedor. A partir da coluna criada, foi realizado um estudo do tempo médio de entrega por estado, o qual revelou discrepâncias significativas, como pode ser observado na Figura 6. Com o objetivo de melhorar a performance do modelo de aprendizado de máquina, optou-se por selecionar apenas uma região do Brasil, visando reduzir as diferenças nos tempos de entrega entre os dados. A região Sudeste foi escolhida para este trabalho, por não possuir grandes discrepâncias dos tempos médios de entrega entre os seus estados, e devido à sua maior volumetria de dados na base, como evidenciado na Figura 7. Logo foram utilizadas apenas os dados referentes às compras feitas por clientes dos estados de São Paulo, Rio de Janeiro, Minas Gerais e Espírito Santo.

A base de dados utilizada no presente estudo contém informações de localização dos clientes e dos vendedores para cada venda realizada, entretanto, não inclui a distância entre essas localizações. Dado que a distância percorrida pela mercadoria é um fator de grande potencial para afetar o tempo de entrega, tornou-se necessário o cálculo desse valor

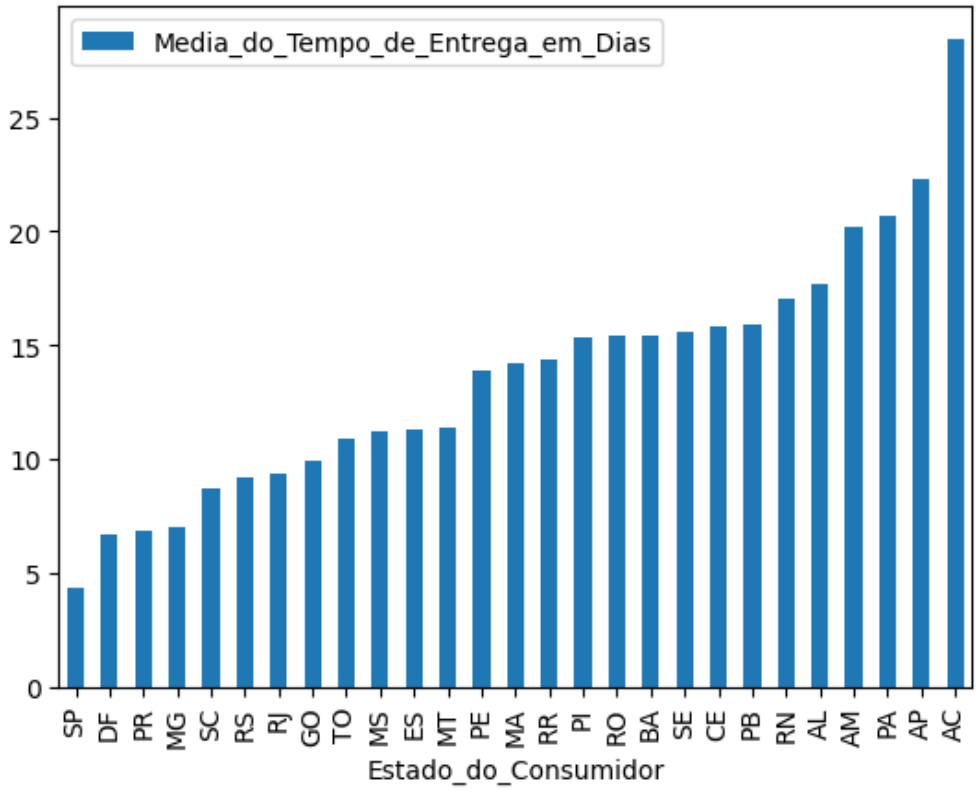


Figura 6 – Tempo médio de entrega por estado do cliente

Fonte: Elaborado pelo autor.

a fim de utilizá-lo no treinamento dos modelos de aprendizado de máquina. Para tal, foi utilizado um dos nove arquivos presentes na base de dados que contém informações de latitude e longitude de cada cidade. Com base nesses dados e fazendo uso da biblioteca *geopy* do Python, foi possível criar uma nova coluna contendo a distância em quilômetros entre a cidade do vendedor e a do comprador, enriquecendo assim a base de dados para análises subsequentes.

Observou-se que a data em que as compras foram realizadas é um fator de influência no tempo total de entrega, conforme evidenciado nas Figuras 8 e 9, que representam respectivamente, o valor médio do tempo de entrega por mês e por semana, é possível observar variações consideráveis em relação aos meses e dias da semana em que as compras ocorreram, destacando a importância desses dados como entrada para o modelo de inteligência artificial utilizado no presente estudo. Contudo, devido à natureza do modelo, que aceita apenas elementos numéricos como entrada, foi necessário transformar os dados de dias da semana e meses do ano em números. No entanto, a simples atribuição de valores progressivos poderia induzir o modelo a interpretar erroneamente que valores numericamente mais altos têm maior impacto em relação aos valores mais baixos, o que poderia levar a conclusões equivocadas. Por exemplo, o modelo poderia erroneamente inferir que o mês de dezembro (representado pelo número inteiro doze) é mais relevante do que o mês

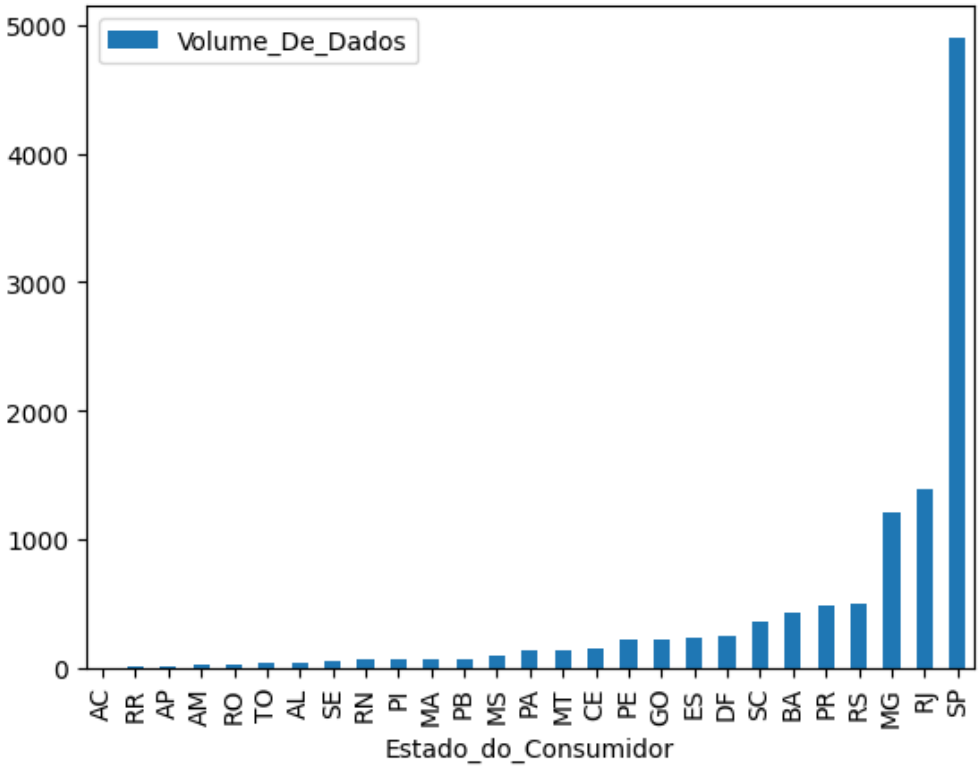


Figura 7 – Volume de dados por estado do cliente

Fonte: Elaborado pelo autor.

de janeiro (representado pelo número inteiro um). Para evitar tal interpretação incorreta, foi utilizada a técnica de codificação conhecida como *one-hot encoding*. Nessa abordagem, cada dia da semana e mês do ano é representado por uma coluna distinta, em que o valor é atribuído como 1 se o pedido ocorreu naquele dia da semana ou mês específico, e 0 caso contrário, possibilitando uma representação adequada dos dados para o modelo de aprendizado de máquina.

Considerando que os dados empregados são provenientes de informações de vendas reais, constatou-se a presença de observações com *outliers*, isto é, valores discrepantes que se diferenciam significativamente dos demais tempos de entrega. Essas situações ocorreram em decorrência de adversidades que resultaram em tempos de entrega atípicos para rotas de entrega semelhantes. Dado que tais valores não seguem um padrão específico e não representam o comportamento usual dos tempos de entrega, eles podem interferir na eficácia do modelo de previsão. Com o propósito de aprimorar a performance do modelo, adotou-se o algoritmo de identificação de *outliers* LOF (*Local Outlier Factor*) para identificação de *outliers*. Este algoritmo foi previamente utilizado no trabalho Guimarães (2022) para a mesma base de dados empregada neste estudo, e obteve resultados promissores, sendo esta a motivação para sua escolha. O algoritmo opera comparando a densidade de cada ponto de dado com seus vizinhos, classificando, assim, cada linha de dado como

um valor atípico ou não. Em seguida, todos os valores identificados como *outliers* foram eliminados da base de dados.

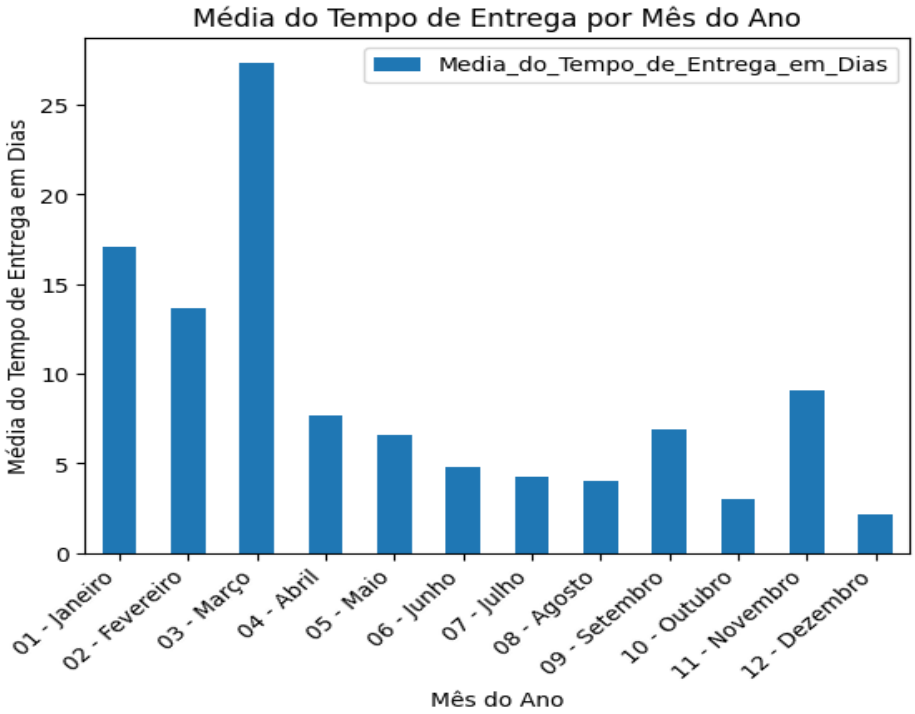


Figura 8 – Média do tempo de entrega por mês do ano

Fonte: Elaborado pelo autor.

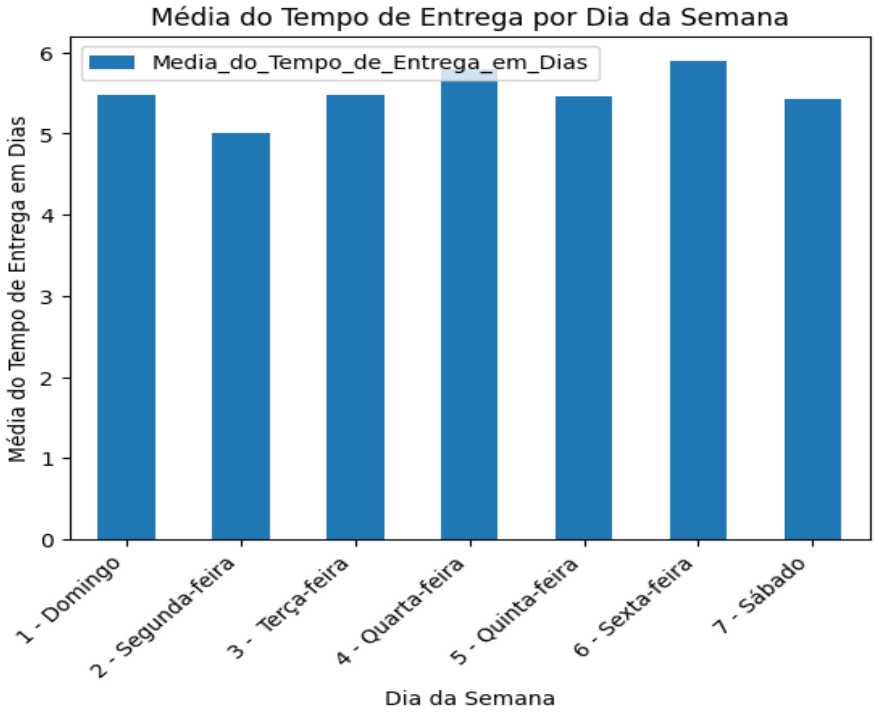


Figura 9 – Média do tempo de entrega por dia da semana

Fonte: Elaborado pelo autor.

3.2 Treinamento dos Modelos de Aprendizado de Máquina

O presente estudo se propôs a realizar a tarefa de regressão para predizer o tempo de entrega de compras *online*, utilizando dois modelos distintos de aprendizado de máquina, as Redes Neurais Artificiais e o *Random Forest*, ambos conhecidos por apresentarem resultados promissores em problemas de regressão em trabalhos anteriores. Essa escolha foi motivada tanto pela relevância desses modelos quanto pela sua utilização como objeto de estudo da disciplina de Inteligência Artificial, inserida no curso de Engenharia de Controle e Automação na Universidade Federal de Ouro Preto.

Como etapa prévia ao treinamento dos modelos foi necessário definir o atributo alvo do conjunto de dados, isto é, o valor que o modelo buscará prever. Também como os atributos que seriam utilizadas no treinamento. Nesse sentido, os dados referentes ao tempo de entrega real foram selecionados como o atributo alvo do modelo, e para o treinamento, foram escolhidos os dados de preço do produto, valor do frete, peso, largura, altura e comprimento do produto, distância percorrida na entrega, o mês e o dia da semana da data de confirmação da compra.

Devido às diferentes escalas de variação dos dados selecionados, foi utilizado o método de normalização de dados *MinMaxScaler*, que mapeia os dados de forma que o menor valor seja zero e o maior valor seja um.

3.2.1 Treinamento do Modelo de Rede Neural Artificial

O modelo de Rede Neural Artificial foi implementado utilizando a biblioteca Keras, na versão 2.12.0. Para construir o modelo, foi utilizada a função "keras.Sequential", sendo que cada camada da rede neural densa foi definida utilizando o comando "keras.layers.Dense".

Foram utilizadas 6 camadas, para cada uma foi definido o número de neurônios, e todas utilizaram a função de ativação *softplus*. A quantidade de neurônio por camada pode ser observado na Figura 10. Para compilar o modelo foi necessário escolher uma função de perda, que será minimizada durante o treinamento. Nesse trabalho foi utilizado o erro médio quadrático, comumente utilizada em modelos de regressão.

O treinamento do modelo foi realizado utilizando a técnica de validação cruzada *k-fold*. Os valores utilizados para a quantidade de épocas de treinamento, o *batch size* e o número de épocas podem ser observados na Figura 11. Além disso, a taxa de aprendizado foi definida dinamicamente, de forma que após as primeiras 50 épocas de treinamento, ela seja reduzida a metade de seu valor a cada época.

Todos os hiperparâmetros foram cuidadosamente ajustados com base em estudos realizados e em diversos testes, visando obter a combinação entre os mesmos para gerar um modelo de Rede Neural Artificial com o melhor desempenho.

```
model = keras.Sequential([
    keras.layers.Dense(32, activation='softplus', input_shape=[len(df_treino.keys())]),
    keras.layers.Dense(64, activation='softplus'),
    keras.layers.Dense(128, activation='softplus'),
    keras.layers.Dense(64, activation='softplus'),
    keras.layers.Dense(32, activation='softplus'),
    keras.layers.Dense(1)

])
```

Figura 10 – Definição das camadas da rede neural artificial

Fonte: Elaborado pelo autor.

```
history = model.fit(
      X_train,
      y_train,
      epochs = 1000,
      batch_size=8,
      verbose = 0,
      validation_data=(X_test, y_test),
      callbacks=[early_stop, PrintDot(), lr_scheduler]
)
```

Figura 11 – Hiperparâmetros para o trinamento do modelo de redes neurais artificiais

Fonte: Elaborado pelo autor.

3.2.2 Treinamento do Modelo de *Random Forest*

O modelo de *random forest* foi implementado utilizando a biblioteca *sklearn*. Para construir o modelo foi utilizado a função "RandomForestRegressor". Os valores dos hiperparâmetros, assim como no modelo de rede neural artificial, foram definidos com base no estudo de como cada um interfere no modelo, e comparando diversos testes realizados.

A função "GridSearchCV", disponível na biblioteca *sklearn* do Python, foi utilizada para auxiliar no processo de descoberta dos melhores hiperparâmetros. Essa função executa testes com diferentes valores cogitados para cada hiperparâmetro, explorando todas as combinações possíveis desses valores para determinar qual combinação proporciona o melhor resultado. Assim o hiperparâmetro *N Estimators* foi definido como 500, o *Max Depth* como 50, o *Min Samples Leaf* como 1, o *Min Samples Split* como 50 e o *Max Features* foi definido para utilizar a raiz quadrada do número total de recursos.

O treinamento desse modelo também foi feito utilizando a técnica de validação cruzada *k-fold*, dividindo o conjunto de dados em 10 *folds*.

4 Experimentos e Resultados

No presente capítulo, serão expostos os resultados obtidos por meio da metodologia descrita no Capítulo 3.

4.1 Análise dos Resultados Obtidos com o Modelo de Redes Neurais Artificiais

Após o treinamento do modelo de rede neural, foram conduzidos testes para avaliar seu desempenho no contexto do presente estudo. Para essa avaliação, utilizaram-se as métricas previamente descritas de erro médio absoluto (MAE) e erro quadrático médio (MSE). Os resultados foram o MAE de 2,55 dias e o MSE de 19,39. Os resultados podem ser observados na Tabela 1.

Tabela 1 – Performance do modelo de Rede Neural

	Rede Neural
MSE:	19.39
MAE:	2,55

Fonte: Elaborado pelo autor.

4.2 Análise dos Resultados Obtidos com o Modelo de *Random Forest*

Assim como na rede neural, após o treinamento da *random forest*, foram realizados os testes para avaliar o seu desempenho. Para esse caso também foram utilizadas as métricas de erro médio absoluto e erro quadrático médio, e além dessas, também foi utilizado o erro percentual médio absoluto (MAPE), explicado anteriormente na subseção 2.2.1. Os resultados obtidos apresentaram valores próximos aos do modelo de rede neural artificial, com um MAE de 2,58 dias, MSE de 18,78 e MAPE de 0,71. A Tabela 2 ilustra esses resultados com mais casas decimais.

A partir da aplicação do modelo de *random forest*, foi possível analisar as características mais relevantes para prever o tempo de entrega de produtos, como evidenciado na Figura 12. Essa informação é de grande relevância para as empresas que trabalham com comércio eletrônico, pois permite identificar os elementos que devem ser considerados com maior ênfase no cálculo do prazo de entrega, independentemente do método utilizado para realizar essa previsão. Os dados revelaram que os atributos que mais influenciam o

Tabela 2 – Performance do modelo de Árvore Aleatória

	Modelo de Floresta Aleatória
MSE:	18,78
MAE:	2,58
MAPE:	0,71

Fonte: Elaborado pelo autor.

tempo que a mercadoria demora pra chegar no cliente são: a distância (*Distance*), o valor do frete (*freight_value*), o peso do produto (*product_weight_g*) e o seu comprimento (*product_lenght_cm*).

Notavelmente, o mês de março (*March*) apresentou um valor de influência significativo no gráfico da Figura 12. Entretanto, uma análise aprofundada foi conduzida para investigar essa questão, pois março não representa um período de aumento nas atividades comerciais ou de eventos que possam interferir no tempo de entrega. Nesse estudo, constatou-se que a grande interferência de março ocorreu devido à escassez de amostras de dados para esse mês na base de dados utilizada, com valores de tempo de entrega real significativamente acima da média geral, como pode ser observado na Figura 8 e no gráfico da Figura 13, que mostra o volume de dados, presente na base estudada, por mês do ano. Com uma base de dados mais ampla e melhor distribuída, essa questão pode ser facilmente contornada.

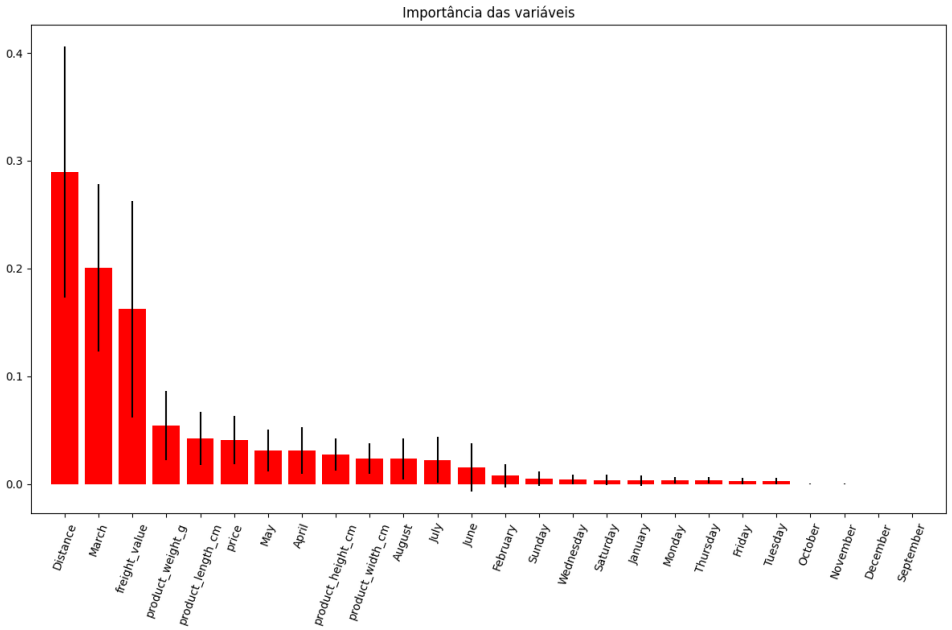


Figura 12 – Importância das Variáveis no Modelo de Árvore Aleatória

Fonte: Elaborado pelo autor.

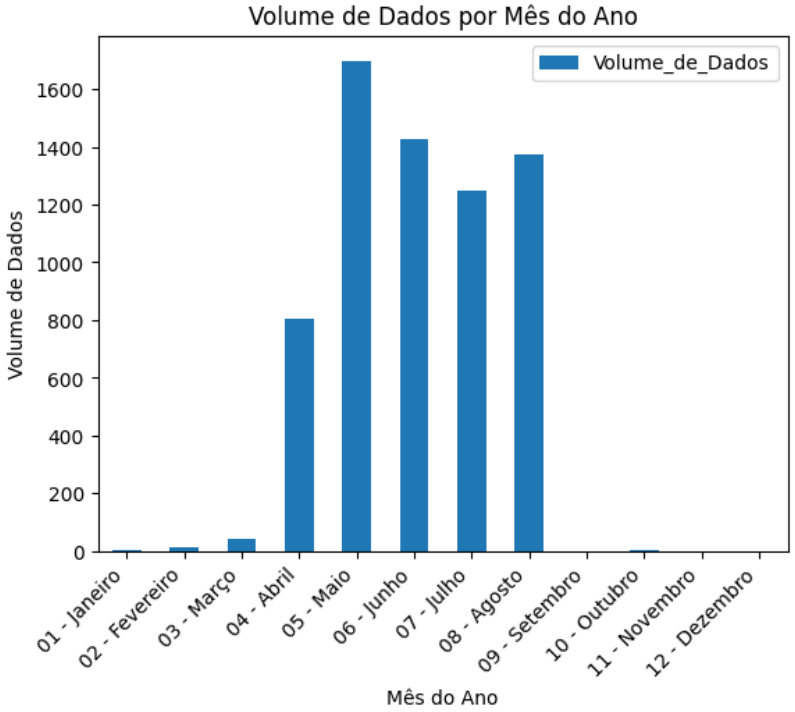


Figura 13 – Volume de dados por mês do ano

Fonte: Elaborado pelo autor.

4.3 Comparação dos Resultados dos Modelos com a previsão Utilizada pelas Empresas

A partir da comparação dos resultados obtidos nos dois modelos de *machine learning* criados, com os valores dos tempos de entrega previstos pelas empresas no momento da compra, registrados no banco de dados utilizado, observa-se uma melhoria substancial. Os modelos foram capazes de prever de forma consideravelmente mais exata o tempo de entrega. As Tabelas 3 e 4 ilustram a comparação entre os modelos e a previsão que foi dada pelas lojas na hora da confirmação da compra.

Tabela 3 – Comparação dos resultados da Rede Neural com a previsão utilizada pelas empresas

	Previsão Site	Modelo de Rede Neural	Diminuição do erro (%)
MSE	210,8839	19,3916	-90,80%
MAE	12,0011	2,5553	-78,71%

Fonte: Elaborado pelo autor.

Tabela 4 – Comparação dos Resultados da *Random Forest* com a Previsão Utilizada Pelas Empresas

	Previsão Site	Modelo de Floresta Aleatória	Diminuição do erro (%)
MSE	210,8839	18,7811	-91,09%
MAE	12,0011	2,5824	-78,48%
MAPE	3,8604	0,7132	-81,53%

Fonte: Elaborado pelo autor.

5 Conclusão

A proposta desse trabalho de conclusão de curso consistiu no estudo de modelos de inteligência artificial e na criação de dois modelos, um de *Random Forest* e outro de Rede Neural, com o objetivo de prever o tempo de entrega de uma mercadoria vendida em uma loja *online*.

Ao fim do trabalho foi possível concluir que os modelos conseguiram alcançar uma melhora significativa em relação ao método atualmente utilizado. No entanto, algumas limitações foram identificadas. O baixo volume de dados disponíveis para o treinamento dos modelos, bem como a concentração desses dados em determinados meses do ano, impactaram negativamente a precisão final dos modelos, resultando em valores de erro superiores ao esperado. Além disso, o fato da base de dados abranger informações de entregas de diferentes empresas também afetou os resultados, uma vez que diferentes métodos e logísticas de entrega podem ser utilizados por cada uma.

Ao analisar o tempo médio de entrega previsto pelos sites, é possível supor que as empresas tendem a estimar prazos de entrega maiores do que os reais como forma de garantir a pontualidade da entrega. Isso é evidenciado pelo fato de a média das previsões ser de 16,63 dias, enquanto o tempo médio real de entrega é de 5,50 dias. No entanto, essa abordagem pode levar os clientes a desistirem da compra devido ao longo período de entrega.

O resultado encontrado pelos modelos criados neste estudo possibilita o cálculo do tempo de entrega com uma precisão consideravelmente melhor do que a atualmente utilizada. Além disso, por meio do modelo de *Random Forest*, foram identificados os atributos de distância percorrida pela mercadoria, valor do frete, peso e comprimento do produto como sendo os mais relevantes na influência do tempo de entrega. Essas informações podem ser utilizadas independentemente da forma como o prazo de entrega é calculado. Dessa forma, os modelos se tornam ferramentas importantes para as empresas, contribuindo para o aumento das vendas ao fornecer aos clientes uma previsão mais próxima do tempo real de entrega das mercadorias.

5.1 Sugestão Para Trabalhos Futuros

Como sugestão para trabalhos futuros, recomenda-se enriquecer a base de dados utilizada, realizando uma coleta de dados mais abrangente e uniforme ao longo do ano. Além disso, seria interessante adicionar informações sobre o clima nas rotas utilizadas durante as entregas. Esses aprimoramentos podem resultar em uma melhora significativa na precisão

final dos modelos. Ao considerar esses fatores adicionais, seria possível obter previsões mais precisas e confiáveis para o tempo de entrega das mercadorias.

Referências

ANDRADE SILVA, Larissa de; FEITOSA, Eduardo L. Avaliando Modelos de Graph Neural Networks para Detecção de Usuários Fraudulentos em e-Commerce. In: SBC. ANAIS Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 2021. P. 280–287. Citado 1 vez na página 18.

ARAUJO, Arthur Cruz de; ETEMAD, Ali. End-to-end prediction of parcel delivery time with deep learning for smart-city applications. *IEEE Internet of Things Journal*, IEEE, v. 8, n. 23, p. 17043–17056, 2021. Citado 1 vez na página 26.

BRANQUINHO FILHO, MSc Delermando. PREDIÇÃO DE FALHAS NA LOGÍSTICA DE ENTREGAS EM E-COMMERCE. *Revista de Ubiquidade*, v. 3, n. 2, p. 6–19, 2020. Citado 5 vezes nas páginas 10, 11, 25, 26.

CHOLLET, Francois. *Deep learning with Python*. Simon e Schuster, 2021. Citado 2 vezes nas páginas 22, 23.

GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc.", 2022. Citado 2 vezes na página 15.

GIACOMEL, Cristina; CARDOSO, Janaína Gularte; ESPÍRITO SANTO, Carlos Alberto do. Proposta de um instrumento para mensurar a satisfação de clientes de e-commerce. *Navus-Revista de Gestão e Tecnologia*, v. 9, n. 2, p. 105–120, 2019. Citado 1 vez na página 10.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep learning*. MIT press, 2016. Citado 5 vezes nas páginas 15, 18–20, 24.

GUIMARÃES, Gabriel Bueno. Uma análise exploratória da influência da detecção de outliers na precificação de produtos em e-commerce., 2022. Citado 1 vez na página 29.

HYNDMAN, Rob J; ATHANASOPOULOS, George. *Forecasting: principles and practice*. OTexts, 2018. Citado 2 vezes nas páginas 16, 18.

JAMES, Gareth et al. *An introduction to statistical learning*. Springer, 2013. v. 112. Citado 1 vez na página 17.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado 1 vez na página 22.

MAMAN, Natan Hermano de. Técnicas de data mining na classificação de usuários em teste de um software do mercado financeiro, 2017. Citado 1 vez na página 23.

MENDONÇA, Herbert Garcia de. E-commerce. *Revista Inovação, Projetos e Tecnologias*, v. 4, n. 2, p. 240–251, 2016. Citado 1 vez na página 10.

QUINTINO, Joyce et al. Avaliaçao de Desempenho de Algoritmos de Machine Learning para Otimizaçao de Simulaçoes de Redes de Computadores. In: SBC. ANAIS do XXV Workshop de Gerência e Operação de Redes e Serviços. 2020. P. 167–180. Citado 1 vez na página 15.

REIS, Carlos Henrique. Otimização de Hiperparâmetros em Redes Neurais Profundas. *Minas Gerais*, 2021. Citado 1 vez na página 19.

SANTANA, Adrielle De Carvalho. *Behavioral and Neurophysiological Representations of Speech Phonemic Units*. 2020. Tese (Doutorado) – Université Grenoble Alpes [2020-....]; Universidade federal de Minas Gerais. Citado 0 vez na página 16.

WANG, Yingying et al. The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, MDPI, v. 10, n. 5, p. 1897, 2020. Citado 0 vez na página 21.