UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS TRINDADE
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
ENGENHARIA DE PRODUÇÃO ELÉTRICA

Gustavo Girardi

Título: Modelos de previsão de preços baseados em machine learning para veículos usados

Florianópolis
2024

Gustavo Girardi

Modelos de previsão de preço baseados em machine learning para veículos usados

Trabalho de Conclusão de Curso submetida ao Departamento de Engenharia de Produção e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de Grau de Engenheiro Eletricista com habilitação em Engenharia de Produção.
Orientador: Prof. Mauricio Uriona Maldonado, Dr.

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Girardi, Gustavo
Modelos de previsão de preço baseados em machine learning para veículos usados / Gustavo Girardi ; orientador, Mauricio Uriona Maldonado, 2024.
88 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Produção Elétrica, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Produção Elétrica. 2. Precificação de produtos automotivos. 3. Ciência de Dados. 4. Aprendizado de Máquina. I. Uriona Maldonado, Mauricio. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Produção Elétrica. III. Título.

Gustavo Girardi

Modelos de previsão de preço baseados em machine learning para veículos usados

O presente trabalho em nível de bacharelado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Carlos Ernani Fries, Dr.
Universidade Federal de Santa Catarina

Prof. Guilherme Ernani Vieira, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Grau de Engenheiro Eletricista com habilitação em Engenharia de Produção.

ICP-Edu
Documento assinado digitalmente
Monica Maria Mendes Luna
Data: 26/12/2024 17:40:33-0300
CPF: ***.104.993-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Monica Mendes Luna, Dra.
Coordenador do Programa

ICP-Edu
Documento assinado digitalmente
Mauricio Uriona Maldonado
Data: 26/12/2024 14:56:19-0300
CPF: ***.972.939-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Mauricio Uriona Maldonado, Dr.
Orientador

Florianópolis, 2024.

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Agradeço primeiramente à minha família, que sempre me deu amparo e apoio incondicional. Sem vocês, nada disso seria possível. O amor, a paciência e as orientações de cada um foram fundamentais para que eu chegasse até aqui.

À minha namorada Larissa, que sempre me apoiou, me incentivou e esteve ao meu lado, oferecendo não apenas compreensão, mas também a força necessária para superar os desafios que surgiram ao longo dessa trajetória.

Aos amigos que fiz durante a graduação, que tornaram essa jornada ainda mais especial. As trocas de experiências, o aprendizado mútuo e a convivência diária foram essenciais para o meu crescimento acadêmico e pessoal. Ao meu amigo Leandro, que esteve comigo nos altos e baixos dessa jornada acadêmica. Sua amizade, ideias e companhia enriqueceram essa experiência.

Também sou muito grato aos meus colegas de trabalho, que contribuíram com seu conhecimento e experiência para o meu aprendizado diário. As habilidades que desenvolvi no ambiente profissional facilitaram enormemente o desenvolvimento deste trabalho e enriqueceram minha trajetória acadêmica.

Por fim, agradeço ao professor orientador Maurício Maldonado, à Universidade Federal de Santa Catarina e aos projetos de extensão que participei durante a graduação, por proporcionarem um ambiente acadêmico de excelência, onde pude não apenas aprender, mas também crescer e me preparar para os desafios da vida profissional.

RESUMO

O mercado brasileiro de veículos seminovos enfrenta desafios na precificação devido à diversidade de fatores que influenciam o valor, como a rápida movimentação do mercado e as variações regionais, o que limita a eficácia de métodos tradicionais, como tabelas de preços, que não capturam a complexidade desse mercado. O objetivo deste trabalho foi desenvolver um modelo de previsão de preços para veículos automotores no Brasil utilizando técnicas de Machine Learning. A partir de dados coletados via web scraping da plataforma Webmotors, foram obtidas informações detalhadas sobre os anúncios e as características físicas dos veículos, como marca, modelo, quilometragem, ano de fabricação, entre outros. O processo metodológico incluiu etapas de limpeza de dados, remoção de outliers e feature engineering. Foram avaliados três modelos preditivos: Lasso Regression, Random Forest e XGBoost. Após a otimização dos hiperparâmetros, o modelo XGBoost (Otimizado) se destacou, apresentando um desempenho superior com um R^2 de 0,95 e um MAPE de 7%. Esse modelo foi capaz de prever com alta precisão os preços dos veículos com base nos atributos coletados. Também trouxemos a aplicação para três diferentes veículos e comparamos o preço previsto pelo modelo com os anúncios de carros semelhantes. A conclusão do estudo ressaltou a eficácia do uso de Machine Learning para a precificação de veículos seminovos, oferecendo maior transparência e precisão em comparação com métodos tradicionais.

Palavras-chave: Precificação de produtos automotivos. Ciência de Dados. Aprendizado de Máquina.

ABSTRACT

The Brazilian market for used vehicles faces pricing challenges due to the diversity of factors that influence value, such as the rapid movement of the market and regional variations, which limits the effectiveness of traditional methods like price tables that do not capture the complexity of this market. The aim of this study was to develop a price prediction model for motor vehicles in Brazil using Machine Learning techniques. Data collected through web scraping from the Webmotors platform provided detailed information about the ads and the physical characteristics of the vehicles, such as make, model, mileage, year of manufacture, among others. The methodological process included data cleaning, outlier removal, and feature engineering. Three predictive models were evaluated: Lasso Regression, Random Forest, and XGBoost. After hyperparameter optimization, the XGBoost (Optimized) model stood out, achieving superior performance with an R^2 of 0.95 and a MAPE of 7%. This model was able to predict vehicle prices with high accuracy based on the collected attributes. We also applied the model to three different vehicles and compared the predicted prices with ads for similar cars. The conclusion of the study highlighted the effectiveness of using Machine Learning for the pricing of used vehicles, offering greater transparency and accuracy compared to traditional methods.

Keywords: Pricing of automotive products. Data Science. Machine Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

No início do século XX, automóveis importados dos Estados Unidos e Europa já circulavam no Brasil e rapidamente se tornaram objeto de desejo popular, logo após a casa própria. O papel dos agentes na comercialização desses veículos foi crucial para concretizar o sonho de consumo dos brasileiros mais abastados, e os lucros elevados favoreceram o surgimento de concessionárias autorizadas para venda e assistência técnica. A expansão da indústria automobilística global, impulsionada pela produção em massa após a Primeira Guerra Mundial, beneficiou também os agentes no Brasil, permitindo-lhes expandir sua atuação e aumentar os lucros nas capitais onde atuavam (CASOTTI; GOLDENSTEIN, 2008).

O mercado de consumo possui uma ampla variedade de produtos, cada qual com combinações de atributos que o tornam único. Apesar de abranger uma grande diversidade de produtos, os negociantes enfrentam dificuldades de estabelecer o real valor da mercadoria, isso se dá pelas diversas variáveis que são demandadas pelo consumidor, que vão de acordo com suas necessidades individuais. No setor automotivo esse cenário não se difere, as montadoras estão constantemente buscando conquistar uma maior participação de mercado, lançando produtos cada vez mais diversificados e buscando incessantemente satisfazer as expectativas dos clientes. No ano de 2023, através de um estudo conduzido pelo Serasa em colaboração com o Instituto Opinion Box, foi identificado que 52% planejavam vender o carro nos próximos meses, contudo, apenas 4 em cada 10 pretendiam trocar por outro. Para 40% das pessoas é encontrada certa dificuldade em calcular as despesas geradas pelo automóvel, além disso, 30% relatam que os custos extras pesam na decisão de manter o veículo. Gastos não planejados são quase uma regra: 90% dos brasileiros já sofreram com algum custo inesperado relacionado ao uso do automóvel, como troca de pneus (52%), consertos mecânicos (50%) e multas (38%). Apesar dos gastos, a maior parte dos brasileiros continua adepta do veículo próprio: 59% acreditam que automóvel é um patrimônio para a família e 58% dos entrevistados entendem que ter um carro ainda vale a pena hoje em dia (SERASA, 2023)

Dados do Instituto Brasileiro de Geografia e Estatística (IBGE) mostram a importância do mercado brasileiro de automóveis. Em 2022 o Brasil possuía mais de 115 milhões de automóveis em circulação e vem aumentando gradativamente ao longo dos anos (IBGE, 2023). A indústria automotiva brasileira se destaca como uma peça fundamental na economia do país, apresentando uma sólida presença com 26 fabricantes e 59 unidades industriais distribuídas em 9 estados. Esse setor dinâmico e diversificado contribui significativamente para o desenvolvimento econômico, gerando empregos, impulsionando a inovação e fomentando a produção local (ANFAVEA2023, 2023). As 59 unidades industriais representam uma rede complexa que abrange dife-

rentes regiões do Brasil. Essa distribuição estratégica não apenas facilita a produção eficiente, mas também promove o desenvolvimento regional, levando oportunidades econômicas para diversas áreas do país. Os estados que abrigam essas unidades industriais desempenham papéis cruciais no cenário automotivo, proporcionando mão de obra qualificada, infraestrutura logística adequada e um ambiente propício para investimentos no setor. Essa diversidade geográfica contribui para a resiliência da indústria automotiva brasileira diante de desafios e mudanças no panorama econômico global, Com uma infraestrutura robusta, a indústria automotiva brasileira demonstra sua resiliência e capacidade de adaptação às demandas do mercado. Os 26 fabricantes englobam desde montadoras de renome internacional até empresas locais, criando uma paisagem industrial rica e variada. Além de sua importância econômica, a indústria automotiva desempenha um papel fundamental na inovação tecnológica e no avanço sustentável (ANFAVEA2023, 2023).

O mercado de veículos seminovos e usados alcançou sua segunda maior marca histórica em vendas, com um aumento de 9,1% no volume diário. No último mês do ano de 2023, foram vendidos 1.383.870 veículos, totalizando 14.448.434 unidades comercializadas no mesmo ano. Em março de 2024 foram comercializadas 1.181.376 de unidades, um crescimento de 3,4% em relação a fevereiro. O total acumulado no ano já alcançou a marca de 3.525.531 de veículos comercializados, um volume 4,9% maior do que no mesmo período de 2023. A comparação entre março de 2024 e março de 2023, considerando os resultados em dias úteis, revela um crescimento de 3,7%. No acumulado de janeiro a março, o aumento foi ainda mais significativo, alcançando 11,8%. É importante destacar que a média diária de vendas se manteve em torno de 60 mil unidades, a melhor registrada para o mês de março nos últimos (FENAUTO2024, 2024).

Avaliar o preço de venda de um veículo seminovo envolve diversas abordagens que combinam dados objetivos e análises detalhadas. A determinação do preço é um processo multifacetado que considera uma série de fatores essenciais, incluindo a marca, o modelo, o ano de fabricação, a quilometragem e o estado de conservação. Portanto, ao integrar dados detalhados e recomendações especializadas, é possível alcançar uma avaliação mais precisa e confiável, garantindo tanto para compradores quanto para vendedores um entendimento claro e justo do valor de um veículo seminovo.

Ao analisar as diversas abordagens para avaliar características de veículos usados, notamos que tabelas e guias, fundamentados em dados confiáveis, são facilmente acessíveis por meio de diversas fontes. Ao dispor dessas informações, se torna viável identificar grupos de carros que compartilham características semelhantes. A capacidade de categorizar veículos com base em suas características se revela especialmente útil ao explorar anúncios de carros na internet.

1.1 PROBLEMA DE PESQUISA

Há um consenso geral de que o comportamento do consumidor refere-se, antes de tudo, ao ato de comprar um produto ou serviço específico. No entanto, esse não é o único comportamento de interesse para os psicólogos do consumidor (AJZEN, 2018).

Os estudos sobre comportamento do consumidor podem ser feitos em diferentes níveis de análise. As questões teóricas mais importantes geralmente são formuladas de maneira ampla, abordando decisões gerais, como a de comprar ou não um produto, ou os fatores que influenciam comportamentos como adquirir um seguro de vida, investir em planos de aposentadoria, usar cartões de crédito, entre outros (AJZEN, 2018).

Quando a questão envolve escolher entre duas ou mais opções, geralmente se analisa com mais detalhe, como, por exemplo, por que os consumidores preferem uma marca de carro em vez de outra, um tipo de tratamento médico específico ou uma companhia aérea. Para isso, é essencial definir claramente os elementos de ação, alvo, contexto e tempo das opções, já que essas decisões podem variar conforme o destino, o tipo de produto ou serviço e a situação (AJZEN, 2018).

Em uma situação na qual uma família de classe média deseja comprar um carro, ela tem duas opções: adquirir um veículo novo ou usado. Se o carro for utilizado apenas como meio de transporte diário, a diferença entre as duas opções pode ser mínima. Além disso, como carros novos geralmente são mais caros e se desvalorizam rapidamente após a compra, optar por um veículo usado pode ser uma escolha mais econômica e confiável neste contexto (LI, 2024).

Essa decisão, entretanto, vai além do custo imediato. Para a maioria das pessoas, comprar um carro é a segunda decisão mais importante e dispendiosa, ficando atrás apenas da compra de uma casa. Para os fabricantes de automóveis, compradores de carros pela primeira vez oferecem a oportunidade de criar uma imagem positiva da marca, o que pode resultar em futuras compras repetidas (SHENDE, 2014).

No entanto, no mercado de carros usados, questões de confiança frequentemente complicam as negociações. Proprietários tendem a acreditar que os preços oferecidos pelas concessionárias são baixos, enquanto estas alegam estar sendo justas. Para resolver essa discrepância, é fundamental a atuação de um terceiro, como um programa de previsão de preços. Esse tipo de serviço utiliza dados objetivos para fornecer uma avaliação imparcial, que pode aumentar a transparência e melhorar o processo de negociação, beneficiando tanto proprietários quanto concessionárias. O modelo é alimentado por um conjunto de dados obtido através de scraping de sites especializados, como o carsome.id, para gerar previsões de preços mais precisas e confiáveis.(BUDIONO *et al.*, 2024).

A precificação tradicional de veículos é realizada, predominantemente, por meio de dois métodos principais: a comparação de mercado e as tabelas de preços.

O método de comparação de mercado, também conhecido como método dos

comparáveis, avalia o valor de um veículo com base em anúncios recentes de veículos semelhantes. Para isso, são analisadas diversas características, como marca, modelo, ano de fabricação, quilometragem, condição (se o carro sofreu acidentes ou passou por reparos) e itens adicionais, como características de conforto ou segurança. Esse método permite uma estimativa mais precisa do valor de mercado, pois reflete as condições reais do mercado e as variações de preço que ocorrem em diferentes contextos. Sua principal vantagem é estar diretamente relacionado ao comportamento de compra e venda, proporcionando uma avaliação mais ajustada à realidade atual do mercado.

As tabelas de preços, como a Tabela FIPE no Brasil, fornecem valores médios de mercado com base em características como marca, modelo, ano de fabricação, versão e configuração do veículo. Elas são compostas por um banco de dados de transações, o que permite uma certa padronização nos valores. No entanto, um dos principais pontos fracos desse método é a falta de consideração da condição específica de cada veículo, como quilometragem, além da ausência de ajustes para diferentes regiões. Isso é especialmente problemático no Brasil, um país de dimensões continentais, onde a demanda por tipos de veículos pode variar consideravelmente entre as regiões, levando a uma avaliação menos precisa.

Além disso, outro ponto relevante é a dependência da Tabela FIPE de dados históricos, o que a limita em acompanhar mudanças rápidas no mercado, como o lançamento de novos modelos ou flutuações econômicas que influenciam a procura por certos veículos. Esse aspecto pode tornar a tabela obsoleta em determinados momentos, especialmente durante períodos de incerteza econômica ou mudanças nas preferências dos consumidores, já que ela não consegue refletir com precisão o valor de mercado atual de um veículo.

Essas limitações tornam-se ainda mais evidentes quando se considera o impacto de fatores econômicos e políticos no mercado de veículos. Como mencionou o gerente de Planejamento e Inteligência de Mercado na B3, em entrevista publicada em B3 (2023), o crescimento de 10% em 2023 reflete a recuperação do mercado de financiamento de veículos após a queda observada em 2022. Ele ressaltou que, além da medida provisória do governo para incentivar o setor automotivo, a expansão da oferta de crédito, com os índices de inadimplência controlados, e a redução das taxas de juros foram fatores cruciais para essa recuperação.

Todas essas mudanças no mercado impactam diretamente o preço dos veículos. Uma análise criteriosa, com o cruzamento de dados atualizados, pode ajudar a determinar o preço ideal, identificando tendências de mercado, padrões de consumo e fatores econômicos que afetam a oferta e a demanda. Este trabalho visa enriquecer as pesquisas sobre precificação de veículos usados, aplicando modelos de aprendizado de máquina a dados coletados de anúncios em e-commerce, analisando o desempenho dos modelos e buscando disponibilizar os resultados de forma acessível para

auxiliar na tomada de decisão de agentes do setor.

1.2 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos.

1.2.1 Objetivo Geral

Estimar o preço de veículos usados com base em atributos publicamente disponíveis em um sites de vendas

1.2.2 Objetivos Específicos

- Construir um banco de dados com informações de veículos usados a partir da web-scrapping de uma plataforma de vendas online.
- Identificar padrões e insights nos dados utilizando análise exploratória de dados.
- Estimar o preço de venda de veículos usados a partir de técnicas de machine learning.

1.3 JUSTIFICATIVA

A estratégia de precificação é a política adotada por uma empresa para determinar o valor que cobrará por seus produtos e serviços. As abordagens estratégicas se dividem amplamente em três categorias: precificação baseada em custos, precificação baseada na concorrência e precificação baseada no valor (SAMMUT-BONNICI; CHANNON, 2015). Um fator comum entre as estratégias de precificação é que, no final, a receita total gerada pelo preço estabelecido multiplicado pelas unidades vendidas deve cobrir os custos operacionais e permitir uma margem de lucro suficiente, o que garante um retorno aceitável sobre o investimento. O processo de definir essa estratégia varia de acordo com a indústria e as condições do mercado, a vantagem competitiva disponível e, em alguns casos, as restrições regulatórias. A estratégia de precificação é uma variável chave na modelagem financeira, que determina as receitas alcançadas, os lucros obtidos e os valores reinvestidos no crescimento da empresa para sua sobrevivência a longo prazo (SAMMUT-BONNICI; CHANNON, 2015).

Técnicas de Machine Learning (ML) têm se mostrado ferramentas poderosas para a previsão de preços, pois permitem analisar grandes volumes de dados e identificar padrões complexos, resultando em estimativas mais precisas. Essa abordagem pode ser aplicada na precificação em diversas áreas, como finanças, imóveis e veículos. Nagel (2021) utiliza modelos de Machine Learning para precificar ações, onde investidores precisam prever os fluxos de caixa futuros das empresas. Investidores

que buscam estratégias de negociação que superem o mercado procuram sinais que prevejam os retornos dos ativos. Pesquisadores que testam modelos de precificação de ativos buscam variáveis preditoras que possam prever diferenças nos retornos entre os ativos ou que capturem a variação previsível nos retornos ao longo do tempo. Modelos de risco de crédito exigem preditores de inadimplência. Modelos de hedge e gestão de risco necessitam de previsões sobre a comovimentação dos retornos dos ativos.

O artigo publicado por Adetunji *et al.* (2022) destacou a eficácia da técnica de machine learning Random Forest para a previsão de preços de casas, utilizando as variáveis disponíveis no conjunto de dados de imóveis de Boston. A comparação entre os preços previstos e reais indicou uma diferença pequena, evidenciando que o modelo é capaz de prever com precisão os preços de imóveis.

Ter cuidado ao comprar um carro é fundamental, pois trata-se de um investimento significativo que pode resultar em despesas financeiras imprevistas, como altos custos de manutenção, consumo de combustível ineficiente e depreciação rápida. Informações essenciais como a quilometragem do veículo, tamanho do motor e ano de fabricação são cruciais para avaliar o desgaste e a durabilidade do veículo.

É importante destacar que muitas pessoas, especialmente aquelas menos familiarizadas com o mercado automotivo, enfrentam uma significativa falta de conhecimento e expertise na estimativa precisa dos preços de veículos. Essa lacuna pode resultar em avaliações subjetivas ou imprecisas, prejudicando as transações e a satisfação do cliente. A implementação de modelos de machine learning (ML) representa uma solução promissora para preencher essa lacuna, proporcionando ferramentas e insights objetivos que permitem avaliações mais precisas e fundamentadas.

Além disso, o uso da ciência de dados na compra de veículos é altamente benéfico por diversas razões. A aplicação de técnicas avançadas permite uma análise detalhada na precificação, comum em vários setores. No contexto automobilístico, isso significa que os compradores podem contar com análises abrangentes e precisas para determinar o valor justo de um carro, considerando fatores como marca, modelo, ano, quilometragem, condição e tendências de mercado. Essa abordagem elimina incertezas na negociação de preços e capacita os consumidores a fazer escolhas informadas, garantindo que paguem um preço justo pelo veículo desejado.

Para automóveis, artigo publicado por Asghar *et al.* (2021) revela que a maioria dos veículos usados é vendida na faixa intermediária de preços, com o valor e a quilometragem sendo os principais fatores considerados pelos compradores. O estudo também destaca que, embora o interesse por veículos de luxo seja baixo, isso não impede a compra de carros de luxo usados. Com uma precisão de 90% nas previsões, obtida por meio de modelos de machine learning, como random forest e regressão linear, os resultados indicam que a abordagem é eficaz e benéfica para ambas as

partes no mercado.

Outro estudo de precificação de veículos realizado por Shaprapawad, Borugadda e Koshika (2023) obteve um valor de R^2 de 95,27% ao aplicar o modelo Support Vector Regressor utilizando 90% do conjunto de dados para treinamento e 10% para validação.

Esses dados são fundamentais para fornecer uma estimativa precisa do valor de mercado de um veículo. Utilizar dados fornecidos pelas montadoras e anunciantes permite uma avaliação técnica mais robusta, minimizando a subjetividade inerente às avaliações empíricas. As montadoras fornecem especificações detalhadas e histórico de manutenção, enquanto os anunciantes disponibilizam informações sobre o desempenho de mercado e a demanda por modelos específicos. Essa abordagem técnica facilita a identificação do valor ideal de um veículo seminovo, pois se baseia em critérios quantitativos e qualitativos verificáveis, ao invés de impressões pessoais.

1.4 ESTRUTURA DO TRABALHO

O primeiro capítulo visa introduzir o tema de estudo, destacando os desafios existentes e a motivação que deu origem ao trabalho. Nele, também são delineados os objetivos e as razões que justificam a pesquisa.

No segundo capítulo, dedicado à fundamentação teórica, são discutidos os conceitos essenciais para o desenvolvimento do tema, incluindo uma introdução à teoria relacionada ao Mercado Imobiliário, Machine Learning e Análise de Dados.

No capítulo 3, são detalhadas as metodologias adotadas no projeto, abordando os recursos disponíveis, os métodos de aprendizado de máquina aplicados, e as técnicas de avaliação e otimização utilizadas. O capítulo também descreve em detalhes cada etapa do processo de pesquisa.

Finalmente, o último capítulo apresenta uma análise final do trabalho, avaliando o cumprimento dos objetivos propostos e utilizando tabelas e gráficos para ilustrar as variáveis mais relevantes. Além disso, são oferecidas reflexões finais sobre o trabalho, discutindo suas limitações e sugerindo possíveis direções para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

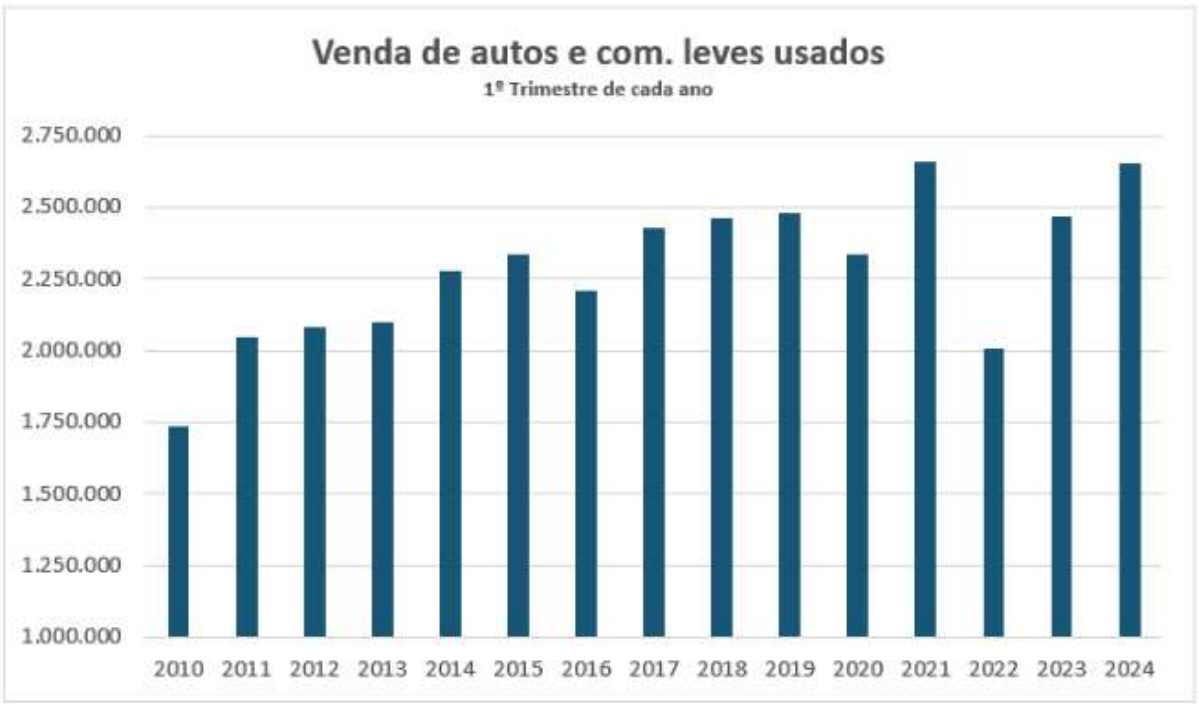
2.1 MERCADO AUTOMOTIVO E SUAS CARACTERISTICAS

No mercado automotivo, observamos uma leve mudança no comportamento dos consumidores. No primeiro trimestre do ano passado, oito marcas que representavam 10% do mercado venderam 46 mil carros; este ano, as mesmas marcas venderam 89 mil veículos, um aumento de 96%, elevando sua participação para 19%. Essas marcas, que ocupam posições intermediárias ou inferiores no ranking de vendas, incluem Nissan (+31%), Honda (+20%), BYD (+2.058%), Caoa-Chery (+170%), Ford (+47%), Ram (+206%), GWM (+4.025%) e Mitsubishi (+30%). Por outro lado, as marcas que lideram o mercado (Fiat, VW, GM, Toyota, Hyundai, Renault e Jeep), que detinham 83% do mercado, viram sua participação cair para 76%. Apesar de terem vendido 364 mil veículos no primeiro trimestre do ano passado, o aumento foi modesto, de apenas 0,7%, totalizando 367 mil unidades neste ano (INFOMONEY, 2024).

Como mostra a figura abaixo, também houve um crescimento nas vendas de veículos usados. Neste ano, foram vendidos 2,65 milhões de automóveis usados, em comparação com 2,47 milhões no primeiro trimestre de 2023, representando um aumento de 7,6%. Em resumo, as vendas de carros usados neste primeiro trimestre estão praticamente iguais ao resultado de 2021, quando atingimos nosso recorde histórico. As vendas de carros usados são importantes porque são a principal moeda de troca para a compra de um carro novo – um mercado de usados aquecido reflete positivamente no mercado de carros novos (INFOMONEY, 2024).

O mercado de venda de veículos usados enfrenta vários desafios que afetam tanto consumidores quanto concessionárias. Um dos principais desafios é a depreciação dos veículos. Diferentemente dos carros novos, que tendem a manter um valor mais estável, os veículos usados perdem valor mais rapidamente. A depreciação é influenciada por vários fatores, incluindo a marca e o modelo do carro. Como SGAToyota (2024) demonstra, veículos de marcas reconhecidas por sua confiabilidade e durabilidade geralmente depreciam mais lentamente, tornando a escolha de marcas com boa reputação um fator essencial. Além disso, a idade do carro desempenha um papel crítico, com a depreciação sendo mais pronunciada nos primeiros anos de propriedade, onde um carro novo pode perder até 20% do seu valor no primeiro ano. A quilometragem também é uma consideração importante, pois carros com mais quilômetros rodados tendem a ter um valor de revenda mais baixo. A condição e a manutenção do veículo são fundamentais; um carro bem conservado e com um histórico de manutenção completo geralmente mantém seu valor melhor. Além disso, as tendências do mercado, como a crescente popularidade de carros elétricos e híbridos, podem influenciar a depreciação, refletindo uma menor perda de valor devido às mudanças nas preferências dos consumidores e no cenário automotivo. Esses fatores podem

Figura 1 – Venda de Veículos Usados.



Fonte: (INFOMONEY, 2024)

complicar a negociação para os vendedores, que precisam ajustar seus preços para refletir a desvalorização.

Outro desafio é a concorrência acirrada e a variedade de opções disponíveis no mercado. Com o aumento da oferta de veículos usados e o crescimento das plataformas de venda online,que como mostra R7 (2024) o número de vendas online de veículos, motos e peças apresentou um crescimento significativo em 2024, aumentando 22,4% no segundo trimestre, segundo a Sondagem de Comércio do FGV IBRE, os consumidores têm acesso a uma gama muito ampla de opções, o que pode tornar o processo de compra mais complexo e demorado. As concessionárias precisam investir em estratégias eficazes de marketing e atendimento ao cliente para se destacar nesse ambiente competitivo. Além disso, a necessidade de implementar tecnologia de ponta para gerenciar inventários e processos de venda também representa um desafio financeiro e operacional. O mercado de veículos usados requer, portanto, um equilíbrio cuidadoso entre oferecer transparência e garantir a competitividade, ao mesmo tempo em que se lida com a constante evolução das preferências dos consumidores e das condições econômicas.

2.1.1 Precificação de Automóveis

O marketing consiste em quatro elementos-chave: o produto, sua promoção, sua colocação ou distribuição e seu preço. Os três primeiros elementos — produto, promoção e colocação — compõem o esforço de uma empresa para criar valor no mercado. O último elemento — o preço — diferencia-se essencialmente dos outros três: ele representa a tentativa da empresa de capturar parte do valor no lucro que ela gera (NAGLE, 2020).

Se o desenvolvimento de produto, a promoção e a colocação eficazes plantam as sementes do sucesso nos negócios, o preço eficaz é a colheita. Embora uma precificação eficaz nunca possa compensar uma execução ruim dos três primeiros elementos, uma precificação ineficaz certamente pode impedir que esses esforços resultem em sucesso financeiro. Infelizmente, isso é uma ocorrência comum (NAGLE, 2020).

Para complicar ainda mais as coisas, a capacidade de colher lucros potenciais está em um estado contínuo de mudança, à medida que tecnologia, regulamentação, informações de mercado, preferências dos consumidores ou custos relativos mudam. Consequentemente, as empresas que esperam crescer de forma lucrativa em mercados em mudança muitas vezes precisam quebrar regras antigas, incluindo aquelas que regem como elas definirão os preços para gerar receita (NAGLE, 2020).

O mercado de carros usados mostra grande potencial. Uma avaliação precisa do preço de carros usados é essencial para o desenvolvimento saudável desse mercado. Para os clientes, saber o preço razoável do carro pode ajudá-los a comprar ou vender um carro usado sem preocupações; para as empresas de aluguel de carros, prever o valor residual é útil para a definição do preço de seus serviços de aluguel; para os bancos e outras instituições financeiras, avaliar o preço do carro de um mutuário pode ajudá-los a controlar a cota do seu empréstimo (CHEN, C.; HAO; XU, 2017).

Outra abordagem considera o impacto das políticas de subsídio e programas de troca, que podem influenciar o preço de carros usados ao oferecer incentivos para a compra de veículos novos ou mais ecológicos. A precificação dinâmica, que leva em conta o comportamento estratégico dos consumidores e a resposta às políticas governamentais, também é relevante. Estudos recentes exploram como as empresas ajustam suas estratégias de precificação em resposta a mudanças nas políticas e no comportamento dos consumidores, utilizando métodos como análise de dados e estratégias de reembolso de troca para otimizar os preços e maximizar as vendas (ZHANG, X. *et al.*, 2024).

A Tabela Fipe é uma referência de preços médios de veículos no mercado nacional, amplamente utilizada para seguros e financiamentos. Ela apresenta valores para pagamento à vista, destinados ao consumidor final (pessoa física) e considerando a revenda de veículos no Brasil (FUNDAÇÃO INSTITUTO DE PESQUISAS ECONÔMICAS,

2024).

Para veículos zero quilômetro, a Tabela Fipe estima o valor médio considerando versões básicas, intermediárias e completas de cada modelo. Já para veículos seminovos, a tabela calcula a média dos preços anunciados, excluindo os valores extremos (muito altos e muito baixos) para obter um valor mais representativo. Veículos para revenda, uso governamental, frotistas, e aqueles com alterações como blindagem são desconsiderados, assim como importados independentes ou de marcas não consolidadas (FUNDAÇÃO INSTITUTO DE PESQUISAS ECONÔMICAS, 2024).

Vale ressaltar que os preços da Tabela Fipe são apenas referências. Fatores como região, estado de conservação, quilometragem, cor, acessórios e outros elementos influenciam o valor final, que pode ser ajustado conforme a negociação entre comprador e vendedor (FUNDAÇÃO INSTITUTO DE PESQUISAS ECONÔMICAS, 2024).

2.1.2 Caracteristicas dos Automóveis

Os automóveis são caracterizados por uma série de aspectos técnicos e de desempenho que variam conforme o modelo, marca e tipo de veículo. O motor, por exemplo, é definido pela sua cilindrada, potência e torque, que determinam a força e a capacidade de aceleração do carro. A transmissão, que pode ser manual, automática ou CVT, e o tipo de tração, seja dianteira, traseira ou integral, também influenciam no comportamento do veículo na estrada. Além disso, o chassi e a carroceria, que englobam o tipo de carroceria (como sedan ou SUV) e as dimensões do veículo, são fatores cruciais para a estabilidade, conforto e segurança durante a condução.

As características de um carro muitas vezes refletem as escolhas e preferências pessoais do motorista. Por exemplo, para aqueles que valorizam a performance, a escolha pode recair sobre um carro esportivo com um motor potente e uma suspensão ajustada para alta velocidade, oferecendo uma experiência de condução dinâmica e envolvente. Por outro lado, quem prioriza a eficiência pode optar por um veículo híbrido ou elétrico, que combina baixo consumo de combustível com uma pegada ambiental reduzida. O tipo de transmissão, seja manual ou automática, também pode ser uma questão de preferência pessoal, dependendo do conforto e do controle desejados.

2.2 CIÊNCIA DE DADOS

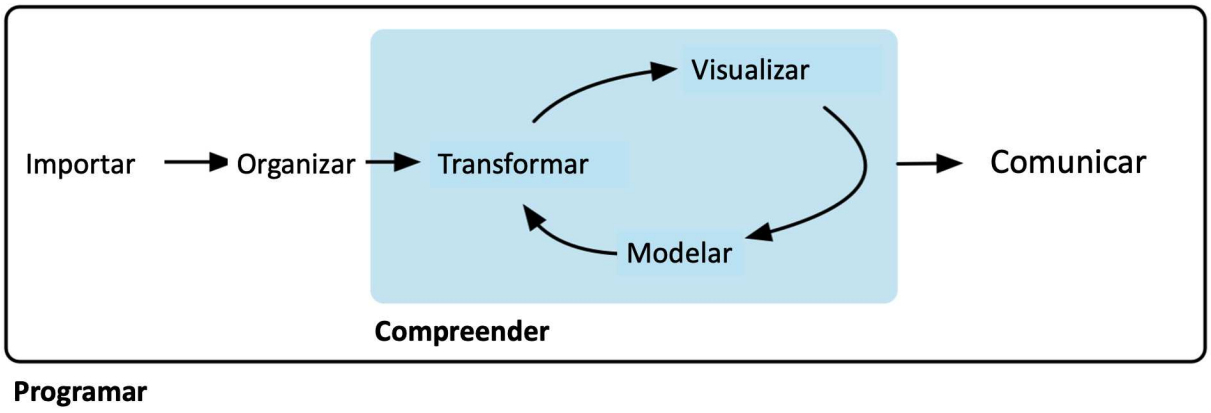
O conceito de "Ciência de Dados"está gradualmente se tornando mais comum. O termo “Ciência” implica conhecimento adquirido por estudo sistemático. De acordo com uma definição, é um empreendimento sistemático que constrói e organiza conhecimento na forma de explicações e previsões testáveis sobre o universo. A Ciência de Dados pode, portanto, implicar um foco em torno dos dados e, por extensão, da

Estatística, que é um estudo sistemático sobre a organização, propriedades e analise de dados e seu papel na inferência, incluindo nossa confiança em tal inferência. Se faz o uso do termo “Ciência de Dados “por diferentes aspectos, primeiro que a matéria-prima, a parte “dados” da Ciência de Dados, é cada vez mais heterogênea e desestruturada – texto, imagens e vídeo, muitas vezes provenientes de redes com relações complexas entre as suas entidades. Segundo que a proliferação de linguagens de marcação, etiquetas, etc. são concebidas para permitir que os computadores interpretem os dados automaticamente, tornando-os agentes ativos no processo de construção de sentido. Em contraste com as primeiras linguagens de marcação, como o HTML, que tratavam da exibição de informações para consumo humano, a maioria dos dados agora gerados por computadores é para consumo por outros computadores (DHAR, 2013).

Com a enorme quantidade de dados agora disponível, empresas de quase todos os setores estão concentradas em explorar esses dados para obter vantagem competitiva. O volume e a diversidade dos dados superaram em muito a capacidade de análise manual e, em alguns casos, ultrapassaram também a capacidade dos bancos de dados convencionais. Ao mesmo tempo, os computadores tornaram-se muito mais poderosos, as redes estão por toda parte, e foram desenvolvidos algoritmos capazes de conectar conjuntos de dados para realizar análises mais amplas e profundas do que antes era possível. A convergência desses fatores levou ao uso cada vez mais disseminado da ciência de dados nas aplicações empresariais (PROVOST; FAWCETT, 2013).

Ferramentas da ciência de dados são formadas por: importar , organizar , transformar e visualizar dados , conforme mostrado na Figura 1 (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

Figura 2 – Modelo das fases de um projeto típico de ciência de dados.



Fonte: (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023)

2.2.1 Importar

Importar dados de diferentes fontes é essencial para a análise e manipulação de informações. Para arquivos CSV, que são amplamente utilizados, existem métodos que facilitam a detecção automática de nomes de colunas e tipos de dados. Quando se trata de arquivos com delimitadores diferentes, como tabulações ou ponto e vírgula, também é possível especificar o delimitador desejado (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

Para dados armazenados em arquivos Excel, há métodos que permitem selecionar folhas e intervalos específicos. Arquivos de dados salvos em formatos compactos podem ser lidos mantendo a eficiência dos dados. Além disso, é possível importar dados de softwares estatísticos como SPSS, Stata e SAS, preservando etiquetas e formatos (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

Dados de bancos de dados SQL podem ser importados diretamente para análise, trazendo tabelas do banco de dados. Arquivos disponíveis online também podem ser lidos diretamente de URLs, facilitando o acesso a dados da web. Essas ferramentas abrangem uma ampla gama de formatos e fontes, oferecendo flexibilidade para diferentes necessidades de importação de dados (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

Muitas vezes, para que os dados importados sejam utilizados na geração de insights, é necessário construir uma estrutura de engenharia de dados que possibilite a coleta e integração de informações provenientes de várias fontes (bancos de dados de ERP, APIs, etc.).

A engenharia de dados envolve o design, desenvolvimento e gerenciamento de sistemas para a ingestão, processamento, armazenamento (CHINTHAPATLA, 2024). Consolidando-as em uma única fonte de verdade chamado Data Warehouse. Esse processo ajuda a evitar duplicidades e inconsistências, fornecendo uma base sólida para as análises.

Um data warehouse é um sistema corporativo utilizado para análise e relatórios de dados estruturados e semi-estruturados provenientes de múltiplas fontes, como transações de ponto de venda, automação de marketing, gestão de relacionamento com clientes, entre outras. Ele é adequado tanto para análises ad hoc quanto para relatórios personalizados. Um data warehouse pode armazenar dados atuais e históricos em um único local e é projetado para oferecer uma visão de longo prazo dos dados ao longo do tempo, tornando-se um componente essencial da inteligência de negócios (CLOUD, 2024).

Os data warehouses tradicionais são hospedados localmente, com os dados provenientes de bancos de dados relacionais, sistemas transacionais, aplicativos empresariais e outros sistemas de origem. No entanto, eles são tipicamente projetados para capturar um subconjunto de dados em lotes e armazená-los com base em esque-

mas rígidos, o que os torna inadequados para consultas espontâneas ou análise em tempo real. Além disso, as empresas precisam adquirir seu próprio hardware e software para um data warehouse local, o que torna caro escalá-lo e mantê-lo. Em um data warehouse tradicional, o armazenamento geralmente é limitado em comparação com o poder de processamento, de modo que os dados são transformados rapidamente e, em seguida, descartados para liberar espaço de armazenamento (CLOUD, 2024).

As atividades de análise de dados de hoje se tornaram centrais para todas as atividades principais dos negócios, incluindo geração de receita, contenção de custos, melhoria de operações e aprimoramento da experiência do cliente. À medida que os dados evoluem e se diversificam, as organizações precisam de soluções de data warehouse mais robustas e ferramentas analíticas avançadas para armazenar, gerenciar e analisar grandes volumes de dados em toda a organização (CLOUD, 2024).

No contexto da nuvem, essa disciplina aproveita os serviços de computação em nuvem para construir pipelines de dados escaláveis e flexíveis. Plataformas de nuvem, como Amazon Web Services (AWS), Microsoft Azure e Google Cloud Platform (GCP), oferecem uma ampla gama de serviços e ferramentas específicas para as tarefas de engenharia de dados. Essa abordagem tem transformado a maneira como as organizações gerenciam, processam e extraem insights de grandes volumes de dados, tornando-se essencial para lidar eficientemente com informações em ambientes de nuvem (CHINTHAPATLA, 2024).

2.2.2 Web Scraping

A web scraping como habilidade traz a oportunidade de desenvolver trabalho na indústria, governo e academia, como por exemplo muitas agencias nacionais de estatística começaram a confiar o web scraping como forma de recolher dados. Uma maneira generalizada de essas agencias usarem web scarpin é automatizando a coleta de preços de produtos de consumo específicos (por exemplo, eletrônicos, habitação e medicamentos) para calcular alguma forma de índice de preços ao consumido (DOGUCU; ÇETINKAYA-RUNDEL, 2021).

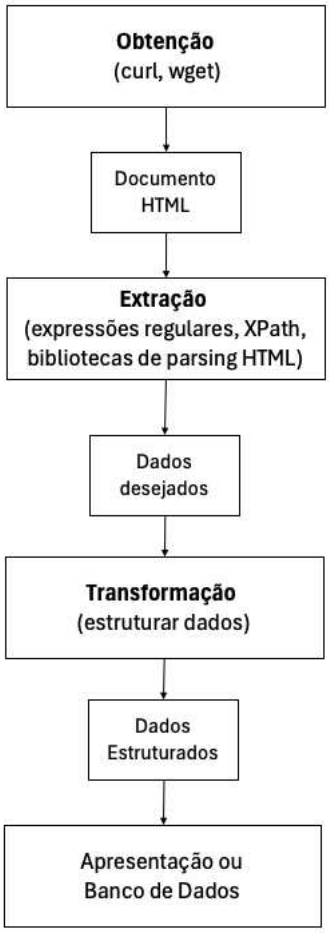
De maneira geral, o web scraping pode ser definido como o processo de extrair e combinar conteúdos de interesse da web de forma sistemática. Nesse processo, um agente de software, também conhecido como robô da web, imita a interação de navegação entre os servidores web e o usuário humano em uma navegação convencional. Passo a passo, o robô acessa os sites necessários, analisa seu conteúdo para encontrar e extrair os dados de interesse, e organiza essas informações conforme desejado (GLEZ-PEÑA *et al.*, 2014).

Existem dois tipos de web scraping. A primeira é a captura de tela, onde você extrai dados do código-fonte de um site, com um analisador HTML ou correspondência de expressão regular. A segunda é usar interfaces de programação de aplicativos,

comumente chamadas de APIs. É aqui que um site oferece um conjunto de solicitações HTTP estruturadas que retornam arquivos JSON ou XML (DOGUCU; ÇETINKAYA-RUNDEL, 2021).

O processo de web scraping é dividido em 3 etapas, conforme mostrado na Figura 6, que são:

Figura 3 – Processo de Web Scraping.



Fonte: (PERSSON, 2019)

- Etapa de Busca (Fetching stage): O site desejado com as informações relevantes deve ser acessado primeiro, na fase conhecida como etapa de busca. Isso é realizado por meio do protocolo HTTP, que é um protocolo da Internet para enviar e receber solicitações de servidores da Web. Os navegadores da Web utilizam métodos semelhantes para obter material nas páginas da Web. Nesta etapa, bibliotecas como `curl` e `wget` podem ser usadas para enviar uma solicitação HTTP GET para o endereço de destino (URL) e obter a página HTML como resposta (PERSSON, 2019).

- Etapa de Extração (Extraction stage): Após a recuperação da página HTML, os dados importantes devem ser extraídos. Expressões regulares, bibliotecas de análise de HTML e consultas XPath são utilizadas nesta etapa, que é chamada de etapa de extração. A Linguagem de Caminho XML (XPath) é uma ferramenta para localizar informações em documentos. Esta é a segunda fase do projeto (PERSSON, 2019).
- Etapa de Transformação (Transformation stage): Agora que apenas os dados relevantes permanecem, eles podem ser convertidos em um formato estruturado para apresentação ou armazenamento. Usando os dados armazenados, é possível reunir informações que podem ajudar a inteligência de negócios a tomar decisões melhores e muito mais (PERSSON, 2019).

2.2.3 Limpar

A limpeza de dados é uma etapa crucial no processo de análise de dados, responsável por garantir a qualidade e a integridade das informações. Ela envolve a identificação e a correção de erros, como valores ausentes, duplicados ou inconsistentes, que podem comprometer os resultados das análises. Além disso, a limpeza pode incluir a padronização de formatos, a remoção de outliers que não são relevantes para o estudo e a transformação de dados para atender aos requisitos específicos de um modelo ou metodologia. Um conjunto de dados bem limpo proporciona uma base sólida para insights confiáveis e decisões informadas, minimizando o risco de interpretações errôneas ou vieses.

A limpeza de dados, também chamada de depuração ou higienização de dados, lida com a detecção e remoção de erros e inconsistências nos dados para melhorar a qualidade dos mesmos. Problemas de qualidade de dados estão presentes em coleções de dados individuais, como arquivos e bancos de dados, por exemplo, devido a erros de digitação durante a entrada de dados, informações ausentes ou outros dados inválidos. Quando múltiplas fontes de dados precisam ser integradas, como em data warehouses, sistemas de bancos de dados federados ou sistemas de informação globais baseados na web, a necessidade de limpeza de dados aumenta significativamente. Isso ocorre porque as fontes frequentemente contêm dados redundantes em diferentes representações. Para fornecer acesso a dados precisos e consistentes, torna-se necessária a consolidação das diferentes representações de dados e a eliminação de informações duplicadas (RAHM; DO *et al.*, 2000).

Além disso, os outliers são casos desviantes que exercem um impacto indevido nos resultados da análise. Eles podem tanto aumentar quanto diminuir as médias e, ao fazer isso, criar significância artificial ou encobrir significância real. Quase sempre, eles aumentam a dispersão, o que, por sua vez, aumenta os erros e distorce correlações. A

inclusão de outliers em um conjunto de dados torna o resultado da análise imprevisível e não generalizável, exceto para uma população que acontece a incluir o mesmo tipo de outlier (BEHRENS; YU, 2003).

Tukey (1977) introduziu o conceito de "cercas"(fences) em boxplots para identificar outliers. Ele definiu as cercas internas como $Q1 - (1{,}5 \times IQR)$ e $Q3 + (1{,}5 \times IQR)$, enquanto as cercas externas são definidas por $Q1 - (3 \times IQR)$ e $Q3 + (3 \times IQR)$. As observações situadas entre uma cerca interna e sua cerca externa mais próxima foram denominadas “fora”, enquanto as que estão além das cercas externas foram classificadas como “muito fora”.

High (2000) renomeou essas categorias, referindo-se às observações “fora” como potenciais outliers e às “muito fora” como outliers problemáticos. As observações “fora” e “muito fora” podem, portanto, ser chamadas de possíveis outliers e prováveis outliers, respectivamente. Essa abordagem se mostrou eficaz, especialmente ao lidar com grandes conjuntos de dados contínuos que não apresentam alta assimetria.

Multiplicar o IQR por 1,5 cria uma “margem” em torno dos quartis que ajuda a incluir a maioria dos dados legítimos, excluindo apenas os pontos mais distantes, que tendem a se afastar da tendência central dos dados. Esse critério é amplamente utilizado porque funciona bem em distribuições simétricas ou levemente assimétricas, capturando a maioria dos dados em distribuições normais.

O uso de 1,5 vezes o IQR também oferece um bom equilíbrio entre precisão e sensibilidade. Ele evita a exclusão de valores próximos aos quartis que podem conter informações valiosas, enquanto filtra dados que se distanciam significativamente e provavelmente não representam a tendência.

Em distribuições altamente assimétricas, é possível ajustar o multiplicador, como usar 3 vezes o IQR, para um critério de outliers mais flexível e que melhor se ajuste ao perfil dos dados.

2.2.4 Transformar

Sobre a transformação de dados, é essencial entender as diversas técnicas e ferramentas usadas para a manipulação e análise de informações. A transformação de dados começa com o uso de vetores lógicos, que, apesar de serem os tipos mais simples de vetores, são fundamentais para executar operações condicionais e filtrar dados. A habilidade de criar e manipular vetores lógicos permite comparações numéricas e a combinação desses vetores usando álgebra booleana, crucial para resumos e transformações condicionais de dados (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

Outro aspecto crucial é a conversão entre diferentes tipos de variáveis, especialmente entre strings e formatos numéricos. Essa capacidade de transitar entre tipos de dados é vital para integrar e analisar informações em diferentes formatos. Expres-

sões regulares são particularmente úteis para a manipulação avançada de strings, facilitando a busca, substituição e análise de padrões complexos em textos. Esse conhecimento é fundamental para a limpeza e preparação de dados textuais (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

A manipulação de datas e horários também apresenta desafios significativos devido à complexidade dos formatos. Utilizar pacotes especializados é essencial para superar esses desafios e garantir uma análise temporal precisa dos dados. Além disso, é importante lidar com a ausência de dados, distinguindo entre valores ausentes implícitos e explícitos e aplicando técnicas adequadas para seu tratamento. Finalmente, a compreensão das junções de tabelas é vital para integrar dados de diferentes fontes. O uso eficiente de chaves para unir e combinar registros é fundamental para uma análise coesa e integrada dos dados (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

2.2.5 Visualizar

Nesta etapa, utiliza-se a visualização e a transformação para explorar os dados de maneira sistemática, em um processo conhecido como Análise Exploratória de Dados (EDA, do inglês Exploratory Data Analysis). A EDA é um ciclo iterativo, onde você gera perguntas sobre seus dados, busca respostas por meio de visualizações, transformações e modelagem, e utiliza os conhecimentos adquiridos para refinar essas perguntas ou criar novas. Não é um processo formal com regras rígidas, mas sim um estado mental de curiosidade e investigação. Durante as fases iniciais, é importante se sentir livre para explorar todas as ideias que surgirem, aceitando que algumas serão frutíferas e outras não. Com o tempo, a exploração se afunila em torno de insights produtivos que podem ser documentados e compartilhados (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

A EDA é essencial em qualquer análise de dados, mesmo quando as perguntas de pesquisa são previamente definidas, pois investigar a qualidade dos dados é sempre necessário. A limpeza de dados, por exemplo, é uma aplicação da EDA, onde se faz perguntas sobre a conformidade dos dados com as expectativas. Para isso, todas as ferramentas da EDA, como visualização e transformação, são indispensáveis (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023).

2.2.6 Modelo

A modelagem é uma etapa crucial no processo de Data Science, onde os dados preparados são utilizados para treinar algoritmos que visam identificar padrões e fazer previsões. Nesta fase, escolhem-se os modelos matemáticos ou estatísticos mais adequados ao problema em questão, como regressão, classificação, clustering, entre outros. O processo envolve o treinamento do modelo, onde ele aprende a partir dos

dados de treino, e a validação, onde seu desempenho é avaliado utilizando dados de teste. Durante a modelagem, também são ajustados hiperparâmetros e realizada a otimização do modelo para melhorar sua precisão e capacidade preditiva. A eficácia do modelo é medida por meio de métricas específicas, garantindo que ele atenda aos objetivos do projeto antes de ser implementado em produção (JAMES *et al.*, 2013).

Os modelos ajudam a entender e prever dados. No contexto de aprendizado supervisionado, esses modelos são desenvolvidos para prever uma saída com base em uma ou mais entradas, sendo aplicados em áreas como negócios, medicina e astrofísica. Já no aprendizado não supervisionado, embora não haja uma saída predefinida, os modelos permitem identificar padrões e estruturas nos dados, oferecendo insights valiosos (JAMES *et al.*, 2013).

Ao longo da história, diversas técnicas de modelagem foram desenvolvidas e aprimoradas. Desde o método dos mínimos quadrados de Legendre e Gauss no século XIX, que deu origem à regressão linear, até as árvores de classificação e regressão introduzidas por Breiman e colaboradores nos anos 1980, a evolução dos modelos estatísticos tem sido marcada pela busca de métodos mais eficientes e precisos. Com o avanço da tecnologia, os métodos não lineares tornaram-se viáveis, e o campo do aprendizado estatístico se expandiu, especialmente com o advento de softwares como o R, que democratizou o acesso a essas ferramentas de modelagem.

2.2.7 Comunicar

A visualização de dados é uma técnica essencial para a análise e compreensão de grandes volumes de informações. Ao transformar dados brutos em gráficos, mapas, e outras representações visuais, ela facilita a identificação de padrões, tendências e outliers que podem não ser imediatamente perceptíveis em tabelas ou listas. A visualização permite que analistas e tomadores de decisão interpretem os dados de forma mais intuitiva e eficaz, comunicando insights complexos de maneira clara e acessível. Ferramentas e técnicas como gráficos de linhas, barras, diagramas de dispersão e mapas de calor são amplamente utilizadas para tornar os dados mais compreensíveis e úteis em diversas áreas, desde a ciência até os negócios.

Gráficos visualmente atraentes ganham força não apenas ao exibir números, mas ao oferecer interpretações profundas e significativas. Os melhores gráficos vão além do trivial; eles abordam temas essenciais, como questões de vida e morte, ou até mesmo a vastidão do universo. Gráficos realmente belos não se limitam a representar dados banais. Em ocasiões raras, a estrutura gráfica e o conteúdo dos dados se unem de maneira tão harmoniosa que resultam em gráficos extraordinários. Essas criações são dignas de admiração, mas replicá-las não segue princípios simples; criar um gráfico excepcional em meio a milhões é uma arte única (TUFTE; GRAVES-MORRIS, 1983).

2.3 MACHINE LEARNING

Desde que começou a evoluir, a humanidade tem empregado uma variedade de instrumentos para simplificar a execução de diversas tarefas. A criatividade do cérebro humano deu origem a várias máquinas distintas, os quais vieram a tornar a vida dos seres humanos mais conveniente, possibilitando que satisfizessem várias exigências da existência, tais como transporte, produção industrial e processamento de informações. O aprendizado de máquina é uma dessas inovações (MAHESH, 2020).

Machine learning é uma área abrangente que está presente em tecnologia da informação, estatística, probabilidade, inteligência artificial, psicologia, neurobiologia e muitas outras disciplinas. Com machine learning, problemas podem ser resolvidos simplesmente pela construção de um modelo que represente bem um conjunto de dados selecionado. Machine learning evoluiu como uma área avançada, desde ensinar computadores a imitar o funcionamento do cérebro humano até expandir o campo da estatística para uma disciplina ampla, que desenvolve teorias fundamentais de computação estatística sobre os processos de aprendizado (NASTESKI, 2017).

O aprendizado de máquina costuma ser categorizado como um subcampo da inteligência artificial, mas acho que a categorização pode ser enganosa à primeira vista. O estudo do aprendizado de máquina certamente surgiu de pesquisas nesse contexto, mas na aplicação de métodos de aprendizado de máquina pela ciência de dados, é mais útil pensar no aprendizado de máquina como um meio de construir modelos de dados.(VANDERPLAS, 2016)

Fundamentalmente, o aprendizado de máquina envolve a construção de modelos matemáticos para ajudar a entender os dados. A "aprendizagem"entra em jogo quando damos a esses modelos parâmetros ajustáveis que podem ser adaptados aos dados observados; desta forma, o programa pode ser considerado como "aprendendo"com os dados. Uma vez que esses modelos tenham sido ajustados a dados vistos anteriormente, eles podem ser usados para prever e entender aspectos de dados recém-observados. Vou deixar para o leitor a digressão mais filosófica sobre até que ponto esse tipo de "aprendizado"matemático baseado em modelos é semelhante ao "aprendizado"exibido pelo cérebro humano (VANDERPLAS, 2016).

O aprendizado de maquina envolve a codificação de programas que ajustam automaticamente seu desempenho de acordo com sua exposição às informações contidas nos dados. Esse aprendizado é obtido por meio de um modelo parametrizado com parâmetros ajustáveis que são ajustados automaticamente de acordo com diferentes critérios de desempenho. O aprendizado de máquina pode ser considerado um subcampo da inteligência artificial (IA) e podemos dividir aproximadamente o campo nas três classes principais a seguir (SEGUI; IGUAL, 2017).

2.3.1 Formas de aprendizagem de máquina

Atualmente, existem diversas técnicas de aprendizado de máquina, cada uma adequada para diferentes tipos de problemas e dados. Nas seções abaixo, vamos explorar as principais abordagens, como o aprendizado supervisionado, não supervisionado e por reforço, detalhando suas características, aplicações e os desafios associados a cada uma.

2.3.1.1 Aprendizagem supervisionada

Os diversos algoritmos geram uma função que mapeia entradas para saídas desejadas. Uma formulação padrão da tarefa de aprendizado supervisionado é o problema de classificação: o modelo precisa aprender (aproximar o comportamento de) uma função que mapeia um vetor para uma das várias classes, observando vários exemplos de entrada e saída dessa função (NASTESKI, 2017).

O aprendizado supervisionado envolve estabelecer uma correspondência entre variáveis de entrada X e uma variável de saída Y, que pode ser usada para prever saídas de novos dados ainda não vistos (CUNNINGHAM; CORD; DELANY, 2008).

Algoritmos que aprendem a partir de um conjunto de treinamento de exemplos rotulados (exemplos) para generalizar para o conjunto de todas as entradas possíveis. Exemplos de técnicas em aprendizagem supervisionada: regressão logística, maquinas de vetores de suporte, arvores de decisão, floresta aleatória, etc (SEGUI; IGUAL, 2017).

O aprendizado supervisionado é a técnica mais comum para o treinamento de redes neurais e árvores de decisão, ambas dependentes das informações fornecidas por uma classificação pré-determinada. Esse tipo de aprendizado é também utilizado em aplicações onde dados históricos ajudam a prever eventos futuros. Existem muitos exemplos práticos de aprendizado supervisionado, como um aplicativo que prevê a espécie de uma íris com base em medições de sua flor. Como mencionado anteriormente, as tarefas de aprendizado supervisionado se dividem em duas categorias: classificação e regressão. Na classificação, o rótulo é discreto, enquanto na regressão, o rótulo é contínuo (NASTESKI, 2017).

2.3.1.2 Aprendizagem não supervisionada

Algoritmos que aprendem com um conjunto de treinamento de exemplos não rotulados. Utilizado para explorar dados segundo algum critério estatístico, geométrico ou de similaridade. Exemplos de aprendizagem não supervisionada incluem agrupamento k-means e estimativa de densidade de kernel (SEGUI; IGUAL, 2017).

No aprendizado não supervisionado, a máquina recebe apenas as entradas $x_1, x_2, \ldots$, mas não tem acesso a saídas-alvo supervisionadas nem recompensas do

ambiente. Embora possa parecer enigmático imaginar o que a máquina poderia aprender sem feedback, é possível estabelecer uma estrutura formal para o aprendizado não supervisionado, considerando que o objetivo da máquina é criar representações das entradas que possam ser usadas para tomada de decisões, previsão de entradas futuras, comunicação eficiente com outras máquinas, entre outros. De certa forma, o aprendizado não supervisionado pode ser encarado como a busca por padrões nos dados, além do que seria considerado puro ruído não estruturado. Dois exemplos clássicos e simples de aprendizado não supervisionado são o agrupamento (clustering) e a redução de dimensionalidade (GHAHRAMANI, 2003).

2.3.1.3 Aprendizagem por reforço

Algoritmos que aprendem através do reforço a partir de críticas que fornecem informações sobre a qualidade de uma solução, mas não sobre como melhorá-la. Soluções aprimoradas são alcançadas explorando iterativamente o espaço de soluções (SEGUI; IGUAL, 2017).

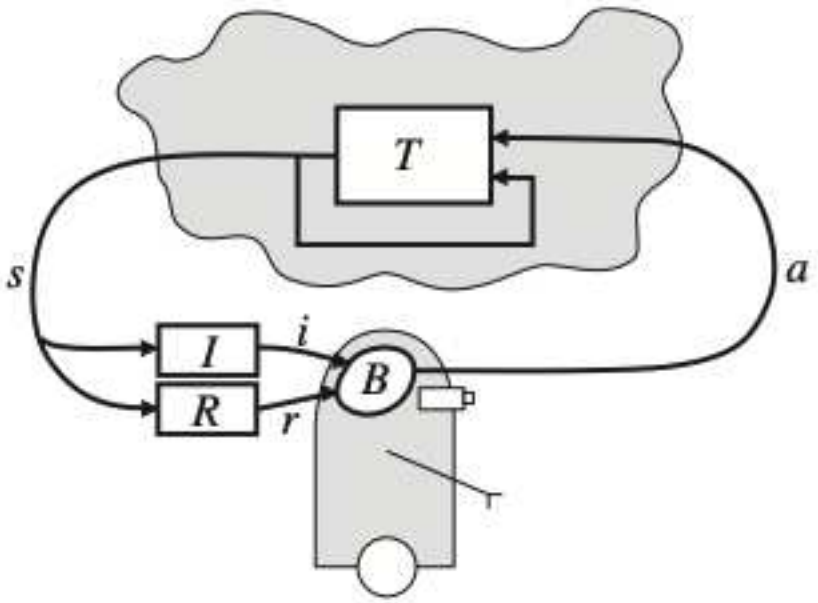
O aprendizado por reforço remonta aos primeiros dias da cibernética e ao trabalho em estatísticas, psicologia, neurociência e ciência da computação. Nos últimos cinco a dez anos, tem atraído um interesse crescente nas comunidades de aprendizado de máquina e inteligência artificial. Sua promessa é tentadora — uma forma de programar agentes por meio de recompensa e punição, sem a necessidade de especificar como a tarefa deve ser realizada. No entanto, existem obstáculos computacionais consideráveis para cumprir essa promessa (KAELBLING; LITTMAN; MOORE, 1996).

No modelo padrão de aprendizado por reforço, um agente está conectado ao seu ambiente por meio de percepção e ação, conforme ilustrado na abaixo. A cada interação, o agente recebe como entrada, i, uma indicação do estado atual, s, do ambiente; o agente então escolhe uma ação, a, para gerar como saída. A ação altera o estado do ambiente, e o valor dessa transição de estado é comunicado ao agente por meio de um sinal escalar de reforço, r. O comportamento do agente, B, deve escolher ações que tendem a aumentar a soma de longo prazo dos valores do sinal de reforço. Ele pode aprender a fazer isso ao longo do tempo por meio de tentativa e erro sistemáticas, orientado por uma ampla variedade de algoritmos, que são abordados nas seções posteriores deste trabalho (KAELBLING; LITTMAN; MOORE, 1996).

2.3.2 Viés e Variância

Métodos de aprendizado de máquina envolvem algoritmos computacionais que relacionam variáveis preditoras a um resultado. Para estimar o modelo, eles buscam o melhor ajuste, seja de forma estocástica (aleatória) ou determinística. O processo de busca varia entre os diferentes algoritmos, mas todos buscam equilibrar dois interesses conflitantes: viés e variância (GOLDSTEIN; NAVAR; CARTER, 2017).

Figura 4 – O modelo padrão de aprendizagem por reforço.



Fonte: (KAELBLING; LITTMAN; MOORE, 1996)

No contexto de aprendizado de máquina, o viés se refere ao grau em que as previsões ajustadas correspondem aos valores reais — ou seja, quão bem o modelo prevê o risco real de morte na população. A variância, por sua vez, mede a sensibilidade das previsões a variações nos dados de entrada, ou seja, como a variabilidade da amostragem afeta as previsões (GOLDSTEIN; NAVAR; CARTER, 2017).

Embora não seja possível quantificar separadamente o viés e a variância de um modelo, esses aspectos são resumidos juntos em funções de perda. Nosso objetivo é reduzir tanto o viés quanto a variância, mas muitas vezes esses dois objetivos estão em conflito: reduzir o viés pode aumentar a variância e vice-versa. Por exemplo, um algoritmo pode prever corretamente todas as mortes no conjunto de dados, mas se ajustar demais às especificidades do conjunto, modelando ‘ruído estatístico’. Isso resultará em um desempenho ruim quando aplicado a um conjunto de dados de validação, caracterizando um modelo ‘overfit’ (ajustado em excesso). Diferentes estratégias são empregadas para equilibrar viés e variância, e os parâmetros que controlam esse equilíbrio são conhecidos como parâmetros de ajuste (GOLDSTEIN; NAVAR; CARTER, 2017).

2.3.3 Validação Cruzada

As amostras são divididas aleatoriamente em k conjuntos de tamanho aproximadamente igual. Um modelo é ajustado usando todas as amostras, exceto o primeiro subconjunto (chamado de primeiro "fold"). As amostras excluídas são previstas por esse modelo e usadas para estimar medidas de desempenho. O primeiro subconjunto é retornado ao conjunto de treinamento e o procedimento é repetido com o segundo subconjunto excluído, e assim por diante. As k estimativas de desempenho reamostradas são resumidas (geralmente com a média e o erro padrão) e usadas para entender a relação entre o(s) parâmetro(s) de ajuste e a utilidade do modelo (KUHN, 2013).

A escolha de k é geralmente 5 ou 10, mas não há uma regra formal. À medida que k aumenta, a diferença de tamanho entre o conjunto de treinamento e os subconjuntos de reamostragem diminui. À medida que essa diferença diminui, o viés da técnica se torna menor (ou seja, o viés é menor para k = 10 do que para k = 5). Nesse contexto, o viés é a diferença entre os valores estimados e os verdadeiros valores de desempenho (KUHN, 2013).

2.3.4 Otimização de hiperparâmetros

Para construir um modelo de aprendizado de máquina (ML) ideal, é fundamental explorar diferentes possibilidades, com o ajuste de hiperparâmetros desempenhando um papel chave na definição da melhor arquitetura e configuração. Esse processo envolve identificar e otimizar os hiperparâmetros, que são parâmetros de configuração do modelo. Inclui a seleção dos hiperparâmetros a serem ajustados, a definição de um espaço de busca e a aplicação de métodos como grid search, tamvem conhecida como busca aleatória ou otimização bayesiana para encontrar a combinação ideal. A escolha correta desses parâmetros é crucial para melhorar o desempenho do modelo em uma tarefa específica. (SHARMA; HARSORA; OGUNLEYE, 2024).

Para superar as desvantagens da busca manual, foram propostos algoritmos de busca automática, como a grid search. O princípio da grid search é a busca exaustiva. A grid search treina um modelo de aprendizado de máquina com cada combinação possível dos valores dos hiperparâmetros no conjunto de treinamento e avalia o desempenho de acordo com uma métrica pré-definida em um conjunto de validação cruzada. Finalmente, a busca em grade produz hiperparâmetros que alcançam o melhor desempenho. Embora esse método permita a sintonização automática e possa teoricamente obter o valor ótimo global da função objetivo de otimização, ele sofre com a maldição da dimensionalidade, ou seja, a eficiência do algoritmo diminui rapidamente à medida que o número de hiperparâmetros ajustados e o intervalo de valores dos hiperparâmetros aumentam (WU *et al.*, 2019).

2.3.5 Modelos de machine learning

2.3.5.1 Análise de regressão

A regressão está relacionada a como fazer previsões sobre quantidades do mundo real, como, por exemplo, as previsões mencionadas nas questões a seguir. Como o volume de vendas muda com as mudanças no preço? Como o volume de vendas é afetado pelo clima? Como o título de um livro afeta suas vendas? Como varia a quantidade de um medicamento absorvido com o peso corporal do paciente; e essa relação depende da pressão arterial? Quantos clientes posso esperar hoje? A que horas devo ir para casa para evitar engarrafamentos? Qual a chance de chuva nas próximas duas segundas-feiras; e qual é a temperatura esperada? (SEGUI; IGUAL, 2017)

Todas estas questões têm uma estrutura comum: pedem uma resposta que pode ser expressa como uma combinação de uma ou mais variáveis (independentes) (também chamadas de covariáveis ou preditores). O papel da regressão é construir um modelo para prever a resposta das variáveis. Este processo envolve a transição dos dados para o modelo (SEGUI; IGUAL, 2017).

Como mostra Sarmento e Costa (2017), no modelo de regressão linear, a relação funcional entre a variável dependente e as variáveis independentes X_i; $i = 1, \ldots, p$ é descrita por (Maroco, 2011):

$$Y_j = \beta_0 + \beta_1 X_{j1} + \beta_2 X_{j2} + \cdots + \beta_p X_{jp} + \varepsilon_j, \quad (j = 1, \ldots, n)$$

Nesse modelo, β representam os coeficientes de regressão, e ε são os erros ou resíduos do modelo. O coeficiente β_0 é o intercepto (ou termo constante), enquanto $\beta_1, \beta_2, \ldots, \beta_p$ representam os coeficientes angulares parciais (ou seja, uma medida da influência de X_i sobre Y, ou a variação em Y por unidade de variação em X_i). O termo ε_j reflete os erros de medição e a variação natural em Y (SARMENTO; COSTA, 2017).

2.3.5.1.1 Regressão Lasso

A regressão Lasso (Least Absolute Shrinkage and Selection Operator) é uma técnica de regressão linear que aprimora a capacidade preditiva do modelo ao incorporar um termo de penalização. Diferente da regressão linear tradicional, que busca minimizar o erro quadrático, a Lasso adiciona à função de custo a soma dos valores absolutos dos coeficientes dos parâmetros do modelo, ponderada por um hiperparâmetro λ. Esse termo de penalização tem o efeito de forçar alguns coeficientes a serem exatamente zero, o que resulta na seleção automática de variáveis mais relevantes, tornando o modelo mais simples e interpretável.

Essa tecnica encolhe alguns coeficientes e define outros como zero, buscando reter as boas características tanto da seleção de subconjuntos quanto da regressão de crista (TIBSHIRANI, 1996).

Uma das grandes vantagens da Lasso é sua capacidade de evitar o overfitting, especialmente em cenários com muitas variáveis ou dados multicolineares. No entanto, essa abordagem pode introduzir viés nos coeficientes estimados, particularmente quando o valor de λ é grande. Além disso, em situações onde as variáveis preditoras são altamente correlacionadas, a Lasso pode escolher uma variável de forma arbitrária, ignorando as outras, o que pode ser uma limitação.

Talks (2020) mostra que uma regressão linear tenta ajustar uma função linear aos dados:

$$y_i = b + \underbrace{w_1 x_{i1} + \cdots + w_p x_{ip}}_{\mathbf{w}\cdot\mathbf{x}_i}$$

Segundo Talks (2020), o procedimento de ajuste envolve a função de custo como soma residual dos quadrados ou RSS. Os coeficientes w são escolhidos para minimizar essa função de custo com base nos dados de treinamento:

$$RSS_{\text{lasso}} = \sum_{i=1}^{n} [y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)]^2$$

No entanto, o overfitting pode ocorrer, o que significa que o modelo "memoriza"o ruído presente nos dados de treinamento, resultando em um alto erro de generalização na base de teste. Esse fenômeno está relacionado à variância do modelo. Para reduzir o erro, uma estratégia é aumentar o viés, ou seja, tornar o modelo mais simples e menos suscetível ao ajuste excessivo aos dados de treinamento (TALKS, 2020).

Para isso, regularizamos os coeficientes w, ou seja, limitamos o seu tamanho. Isso é alcançado ao adicionar um termo na função de custo, de modo que, ao minimizar a função de custo, os coeficientes também sejam automaticamente reduzidos (TALKS, 2020).

$$RSS_{\text{lasso}} = \sum_{i=1}^{n} [y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)]^2 + \alpha \sum_{j=1}^{p} |w_j|$$

Além de reduzir a variância do modelo, a regularização também tem uma aplicação importante em machine learning. Quando há múltiplas features altamente correlacionadas (ou seja, que se comportam de maneira semelhante), a regularização Lasso seleciona apenas uma dessas features e zera os coeficientes das demais, minimizando assim a penalização L1. Isso faz com que o modelo realize automaticamente a seleção de features, atribuindo peso zero a várias delas, ou seja, ignorando-as. Essa

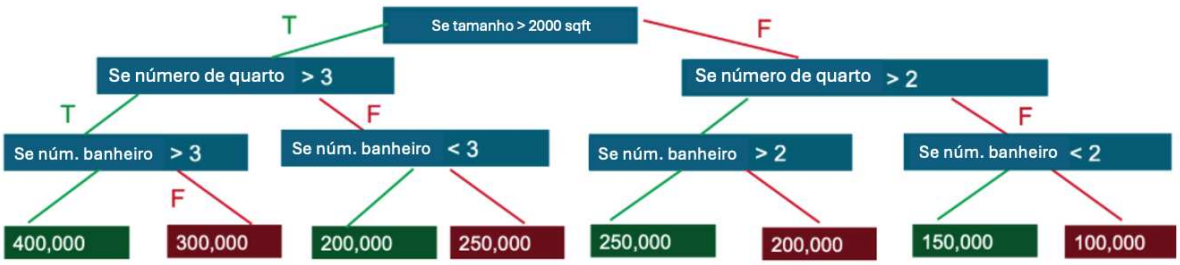
abordagem facilita a interpretação do modelo, o que representa uma grande vantagem. (TALKS, 2020).

2.3.5.2 Random forests

Random Forest são um exemplo de um conjuntométodo, o que significa que ele se baseia na agregação dos resultados de um conjunto de estimadores mais simples. O resultado um tanto surpreendente com tais métodos de conjunto é que a soma pode ser maior que as partes: ou seja, uma votação majoritária entre vários estimadores pode acabar sendo melhor do que qualquer um dos estimadores individuais fazendo a votação (VANDERPLAS, 2016).

Para entender melhor como o modelo Random Forest funciona, é fundamental primeiro compreender as Árvores de Decisão. Árvores de Decisão criam um modelo que prevê o rótulo avaliando uma série de perguntas sobre as características, no formato de decisões if-then-else (verdadeiro/falso), estimando o número mínimo de perguntas necessárias para chegar a uma decisão correta. Elas podem ser usadas tanto para classificação, para prever uma categoria, quanto para regressão, para prever um valor numérico contínuo. No exemplo simples abaixo, uma árvore de decisão é usada para estimar o preço de uma casa (o rótulo) com base no tamanho e no número de quartos (as características) (NVIDIA, 2024).

Figura 5 – Exemplo de Árvores de Decisão.



Fonte: (NVIDIA, 2024)

Como o nome sugere, uma Floresta Aleatória é um conjunto baseado em árvores, no qual cada árvore depende de um conjunto de variáveis aleatórias. De forma mais formal, para um vetor aleatório X de p dimensões, representado por $X = (X_1, ..., X_p)^T$, que representa as variáveis de entrada ou preditoras com valores reais, e uma variável aleatória Y que representa a resposta com valores reais, supomos uma distribuição conjunta desconhecida $P_{XY}(X, Y)$. O objetivo é encontrar uma função de previsão $f(X)$ para prever Y. A função de previsão é determinada por

uma função de perda $L(Y, f(X))$ e é definida para minimizar o valor esperado da perda (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012).

$$E_{XY}(L(Y, f(X)))$$

Onde os subscritos indicam a expectativa em relação à distribuição conjunta de X e Y.

De forma intuitiva, $L(Y, f(X))$ é uma medida de quão próxima $f(X)$ está de Y; penaliza valores de $f(X)$ que estão longe de Y. Escolhas típicas para L são a perda de erro quadrado, $L(Y, f(X)) = (Y - f(X))^2$, para regressão, e a perda zero-um, para classificação (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012):

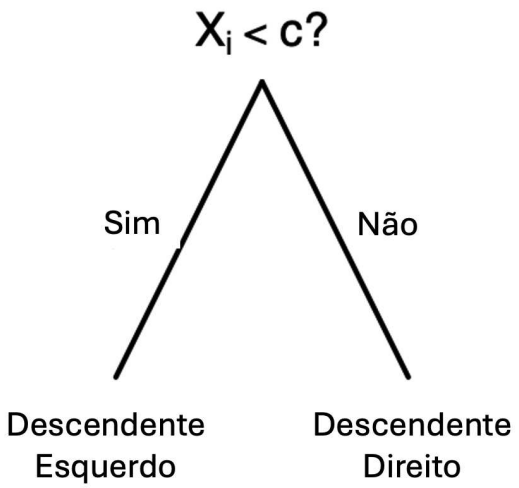
$$L(Y, f(X)) = I(Y \neq f(X)) = \begin{cases} 0, & \text{se } Y = f(x) \\ 1, & \text{caso contrário.} \end{cases}$$

As árvores resultantes são combinadas por meio de votação não ponderada se a resposta for categórica (classificação) ou por meio de média não ponderada se a resposta for contínua (regressão) (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012).

As árvores utilizadas nas Florestas Aleatórias são baseadas nas árvores de particionamento binário recursivo. Essas árvores dividem o espaço dos preditores por meio de uma sequência de partições binárias ("splits") em variáveis individuais. O nó "raiz"da árvore abrange todo o espaço dos preditores. Os nós que não são divididos são chamados de "nós terminais"e formam a partição final do espaço dos preditores. Cada nó não terminal se divide em dois nós descendentes, um à esquerda e outro à direita, de acordo com o valor de uma das variáveis preditoras. Para uma variável preditora contínua, uma divisão é determinada por um ponto de divisão; os pontos para os quais o preditor é menor do que o ponto de divisão vão para a esquerda, os demais vão para a direita (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012).

Conjuntos de dados desequilibrados, nos quais algumas classes são muito menores do que outras, representam um desafio para muitos classificadores. Um classificador ingênuo trabalhará para acertar nas classes grandes, permitindo uma alta taxa de erro nas classes pequenas. Florestas Aleatórias têm um método eficaz para ponderar as classes a fim de fornecer resultados equilibrados em dados desequilibrados. Uma razão para fazer isso é que as variáveis preditoras importantes podem ser diferentes quando o método é forçado a prestar mais atenção a uma classe pequena. Mesmo no caso equilibrado, os pesos podem ser ajustados para obter taxas de erro mais baixas para decisões que têm um alto custo de classificação incorreta. Por exemplo, muitas vezes é mais sério concluir incorretamente que alguém está saudável do que concluir incorretamente que alguém está doente (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012).

Figura 6 – Realizar uma divisão em uma variável preditora contínua X_j, usando um ponto de divisão c.



Fonte: (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012)

Medidas da importância das variáveis preditoras são úteis para a seleção de variáveis e para interpretar a floresta ajustada. Embora seja padrão em muitas aplicações realizar uma análise de componentes principais (PCA) para reduzir a dimensionalidade antes de ajustar um classificador ou preditor de regressão, é possível que os componentes principais não capturem as informações importantes para o problema de previsão. Nesse caso, pode ser preferível obter a importância das variáveis diretamente do algoritmo e, em seguida, refazer o ajuste usando apenas as variáveis preditoras mais importantes (CUTLER, A.; CUTLER, D. R.; STEVENS, 2012).

2.3.5.3 XGBoost

O algoritmo de extreme gradient boosting (XGBoost) destaca-se como uma poderosa técnica de aprendizado em ensemble, oferecendo vantagens como alta flexibilidade, forte poder preditivo, capacidade de generalização, escalabilidade, eficiência no treinamento de modelos e robustez (ZHANG, P.; JIA; SHANG, 2022).

Além de suas características técnicas, o XGBoost consolidou-se como um sistema de aprendizado de máquina escalável para tree boosting, amplamente utilizado e disponível como um pacote de código aberto. Seu impacto é amplamente reconhecido em uma variedade de desafios relacionados ao aprendizado de máquina e à mineração de dados (CHEN, T.; GUESTRIN, 2016).

O fator mais importante por trás do sucesso do XGBoost é sua escalabilidade em todos os cenários. O sistema é mais de dez vezes mais rápido do que soluções

populares existentes em uma única máquina e escala para bilhões de exemplos em configurações distribuídas ou com limitação de memória (CHEN, T.; GUESTRIN, 2016).

A escalabilidade do XGBoost se deve a várias otimizações importantes tanto no sistema quanto no algoritmo. Essas inovações incluem: um algoritmo de aprendizado de árvores inovador para lidar com dados esparsos; um procedimento de esboço de quantil ponderado teoricamente justificado, que permite lidar com pesos de instâncias no aprendizado aproximado de árvores. O uso de computação paralela e distribuída torna o aprendizado mais rápido, o que possibilita uma exploração mais ágil do modelo (CHEN, T.; GUESTRIN, 2016).

Mais importante ainda, o XGBoost explora computação fora do núcleo e permite que cientistas de dados processem centenas de milhões de exemplos em um desktop (CHEN, T.; GUESTRIN, 2016).

A maioria dos algoritmos de aprendizado de árvores existentes é otimizada apenas para dados densos ou requer procedimentos específicos para lidar com casos limitados, como a codificação de variáveis categóricas. O XGBoost, por outro lado, oferece uma abordagem mais eficiente ao tratar todos os padrões de esparsidade de forma unificada. Além disso, o método aproveita a esparsidade dos dados para reduzir a complexidade computacional, tornando-a linear em relação ao número de entradas não ausentes (CHEN, T.; GUESTRIN, 2016).

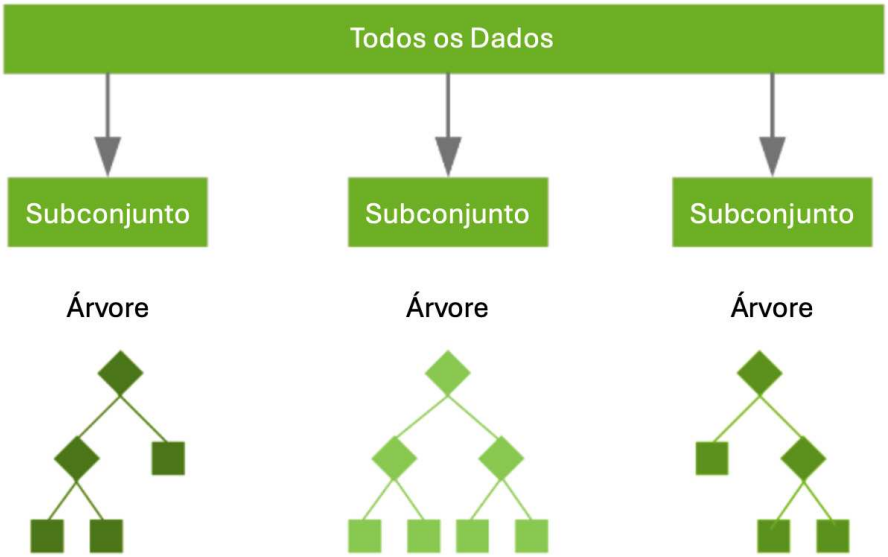
Essa eficiência é particularmente relevante no contexto do Gradient Boosting Decision Tree, um algoritmo de aprendizado de máquina baseado em conjuntos de árvores de decisão, semelhante ao Random Forest. O GBDT é amplamente utilizado tanto para classificação quanto para regressão, aproveitando a combinação de múltiplos modelos para criar resultados finais mais robustos e precisos (NVIDIA, 2024).

Durante muitos anos, o MART foi o método preferido para tree boosting. Mais recentemente, o XGBoost, uma nova abordagem de tree boosting, ganhou popularidade ao vencer diversas competições de aprendizado de máquina (NIELSEN, 2016).

Tanto o Random Forest quanto o GBDT constroem um modelo composto por múltiplas árvores de decisão. A principal diferença entre eles está na forma como as árvores são construídas e combinadas. Enquanto o Random Forest utiliza uma abordagem de votação para combinar os resultados das árvores, o GBDT constrói as árvores de forma sequencial, corrigindo erros das árvores anteriores, com base na minimização de um erro residual (NVIDIA, 2024).

O Random Forest utiliza uma técnica chamada bagging, onde várias árvores de decisão são construídas em paralelo a partir de amostras aleatórias do conjunto de dados (bootstrap). A previsão final é obtida pela média das previsões de todas as árvores, o que ajuda a reduzir a variância e evitar o overfitting. Em contraste, o Gradient Boosting (GBDT) melhora o desempenho de um modelo fraco, combinando-o com vários outros modelos fracos, gerando um modelo final forte. O GBDT é uma

Figura 7 – Múltiplas árvores de decisão.



Fonte: (NVIDIA, 2024)

extensão do boosting, onde a construção sequencial das árvores é guiada pela descida de gradiente, com o objetivo de minimizar os erros do modelo anterior, ajustando as previsões com base nos resíduos de erro (NVIDIA, 2024).

O XGBoost é uma implementação escalável e eficiente do Gradient Boosting, projetada para melhorar a velocidade computacional e a precisão do modelo. Diferente do GBDT, que constrói as árvores de forma sequencial, o XGBoost constrói as árvores em paralelo, otimizando o desempenho e a eficiência. Ele utiliza uma abordagem de nível, avaliando a qualidade das divisões em cada árvore com base nos gradientes, o que torna o algoritmo altamente eficaz e adequado para grandes volumes de dados, mantendo o equilíbrio entre precisão e velocidade (NVIDIA, 2024).

2.4 MÉTRICAS DE AVALIAÇÃO

2.4.1 RMSE

RMSE, ou "Root Mean Square Error"(Erro Quadrático Médio), é uma métrica usada para medir a precisão de um modelo de previsão. Ele calcula a raiz quadrada da média dos quadrados dos erros entre as previsões do modelo e os valores reais. A fórmula é:

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

onde:

- y_i é o valor real,
- $\hat{y}_i$ é o valor previsto,
- n é o número total de observações.

O RMSE fornece uma medida da magnitude dos erros de previsão e tem a vantagem de penalizar grandes erros mais severamente do que pequenos erros. Quanto menor o RMSE, melhor a precisão do modelo.

Para problemas de regressão, onde tentamos prever um valor numérico, os resíduos são fontes importantes de informação. Resíduos são calculados como o valor observado menos o valor previsto (ou seja, $y - \hat{y}$). Ao prever valores numéricos, o erro quadrático médio da raiz (RMSE) é comumente usado para avaliar os modelos. Descrito com mais detalhes no Cap. 7, o RMSE é interpretado como a distância média dos resíduos em relação a zero (KUHN, 2013).

2.4.2 Coeficiente de Correlação e (r) R^2

O coeficiente de correlação (denotado como r) mede a força e a direção da relação linear entre duas variáveis. Ele varia de -1 a 1. Um valor de $r = 1$ indica uma correlação positiva perfeita, onde ambas as variáveis aumentam ou diminuem juntas de forma linear. Um valor de $r = –1$ indica uma correlação negativa perfeita, onde uma variável aumenta enquanto a outra diminui de forma linear. Um valor de $r = 0$ sugere que não há correlação linear entre as variáveis. A fórmula para calcular o coeficiente de correlação é:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Onde x_i e y_i são os valores das variáveis, e $\bar{x}$ e $\bar{y}$ são as médias dessas variáveis.

O coeficiente de determinação (R^2) uma medida estatística que quantifica a proporção da variabilidade na variável dependente que é explicada pelo modelo de regressão. Ele é utilizado para avaliar o ajuste do modelo aos dados e ajuda a entender o quão bem o modelo explica a variação observada na variável dependente. Ele varia de 0 a 1, com $R^2 = 1$ indicando que o modelo explica toda a variabilidade dos dados e $R^2 = 0$ indicando que o modelo não explica nenhuma variabilidade. A fórmula para calcular R^2 é:

$$R^2 = \frac{\text{Variância explicada pelo modelo}}{\text{Variância total}} = 1 - \frac{\text{Soma dos Quadrados dos Resíduos}}{\text{Soma dos Quadrados Totais}}$$

O coeficiente de determinação é frequentemente utilizado para avaliar a adequação de um modelo de regressão. Quando X e Y são variáveis aleatórias distribuídas conjuntamente, o coeficiente de determinação, denotado por R^2, corresponde ao quadrado do coeficiente de correlação entre X e Y. No caso do modelo de regressão, onde temos R^2 = 0,877, indicando que o modelo explica 87,90% da variabilidade dos dados (MONTGOMERY *et al.*, 2009).

Existem várias interpretações incorretas do R^2. Em geral, R^2 não mede a magnitude da inclinação da linha de regressão e um valor alto de R^2 não implica em uma inclinação acentuada. Além disso, R^2 não avalia a adequação do modelo, pois pode ser artificialmente elevado pela adição de termos polinomiais de ordens superiores. Mesmo que X e Y estejam relacionados de forma não-linear, R^2 pode ser alto (MONTGOMERY *et al.*, 2009). Por exemplo, na equação de regressão abaixo, o R^2 pode ser relativamente alto apesar de uma aproximação linear ser inadequada. Finalmente, mesmo com um R^2 elevado, isso não garante que o modelo de regressão faça previsões precisas para observações futuras (MONTGOMERY *et al.*, 2009).(colocar imagem)

2.4.3 MAPE

MAPE (Mean Absolute Percentage Error) é uma métrica usada para avaliar a precisão de um modelo de previsão. Ele mede a precisão do modelo em termos de erro percentual médio absoluto entre os valores previstos e os valores reais.

A fórmula para o MAPE é dada por:

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \times 100$$

onde:

- A_t é o valor real na observação t,
- F_t é o valor previsto na observação t,
- n é o número de observações.

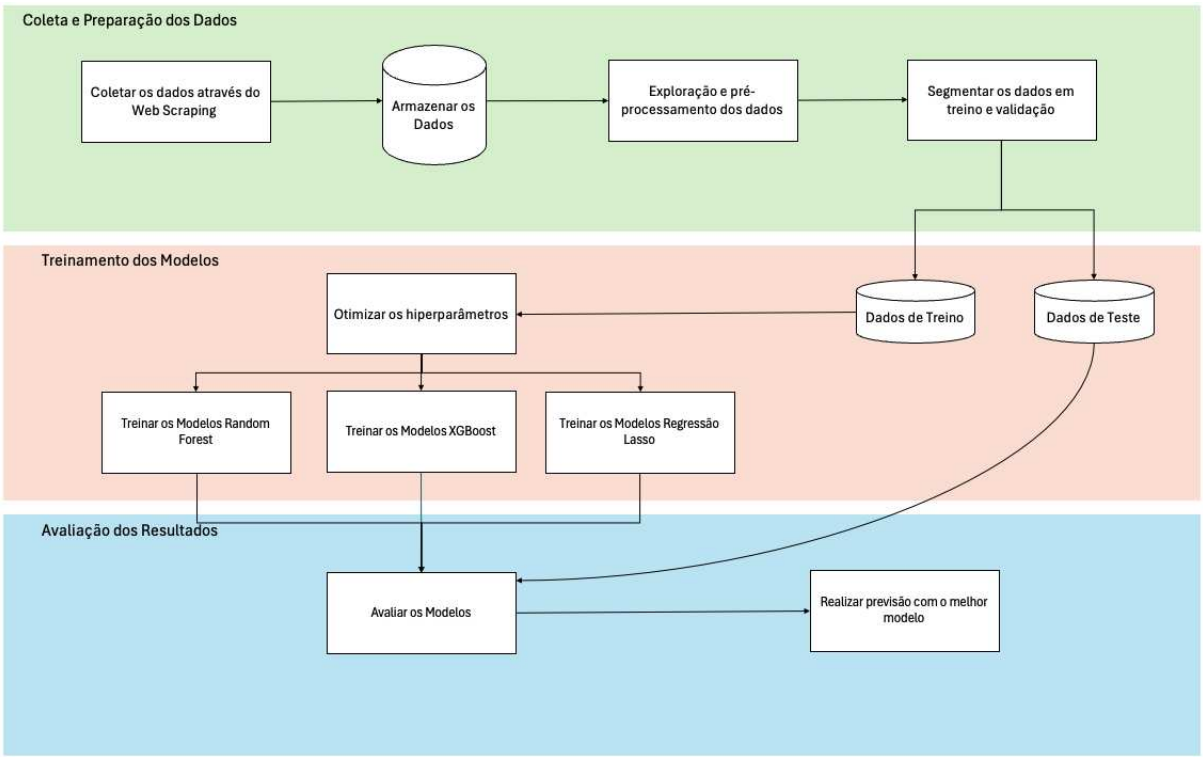
O MAPE é expresso como uma porcentagem e quanto menor for o valor, melhor é a precisão do modelo de previsão. Um MAPE de 0% indica previsões perfeitas.

3 METODOLOGIA

Este trabalho de conclusão de curso, classificado pela ABEPRO na área de Pesquisa Operacional e subárea de Modelagem, Simulação e Otimização, adota uma abordagem quantitativa. Utilizando dados extraídos dos anúncios para solucionar um problema previamente identificado, caracteriza-se como uma metodologia de natureza aplicada.

Este capítulo discute o enquadramento da pesquisa, os materiais e métodos empregados no desenvolvimento do trabalho, além dos procedimentos metodológicos e seus detalhes. A figura a seguir ilustra um fluxograma que descreve em detalhes o passo a passo dessa metodologia, dividida em três etapas principais: coleta e preparação dos dados, treinamento dos modelos e avaliação dos resultados.

Figura 8 – Fluxograma com visão geral do projeto.



Fonte: Adaptado de Potrich (2024)

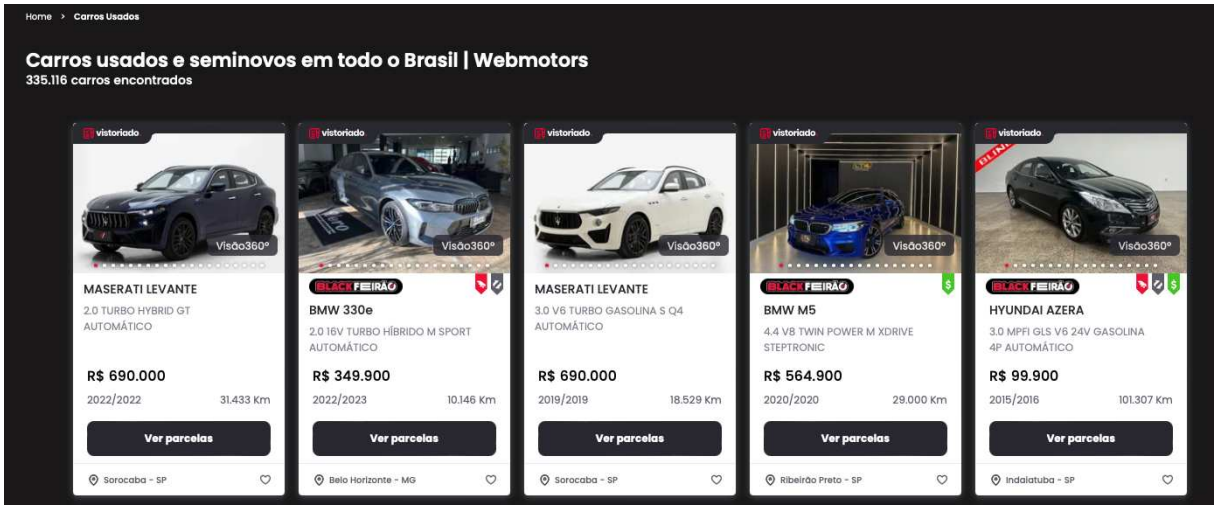
3.1 COLETA DE DADOS

Os dados deste estudo foram obtidos através da extração de anúncios de automóveis usados disponíveis no site WebMotors, abrangendo todo o Brasil, com base na quantidade de veículos listados no portal durante o mês de outubro de 2023. Durante a coleta, não foi aplicada nenhuma ordenação no site, a fim de evitar vieses na

seleção dos veículos com base em faixas de preço específicas. Essa estratégia foi escolhida para garantir que o conjunto de dados fosse o mais representativo possível, abrangendo uma maior dispersão e variabilidade nos preços dos veículos. No entanto, essa abordagem resultou em uma base de dados não homogênea, o que foi explorado, analisado e devidamente tratado ao longo do estudo.

Abaixo, está a imagem de como as informações estavam dispostas na página inicial, incluindo dados sobre o modelo, marca, tamanho do motor, ano de fabricação e do modelo, além da quilometragem e do local de venda.

Figura 9 – Site Webmotors.



Fonte: (WEBMOTORS, 2024)

Para coletar dados reais do mercado automotivo no site WebMotors, foi desenvolvido um algoritmo de web scraping que segue os links das páginas da web, utilizando as ferramentas requests e BeautifulSoup. A biblioteca requests é responsável por realizar a comunicação com o protocolo HTTP, adaptando-o para a semântica orientada a objetos do Python. Já a biblioteca BeautifulSoup facilita a formatação da página HTML, convertendo-a em objetos Python facilmente iteráveis que representam estruturas XML. As informações coletadas são processadas e armazenadas em um arquivo XLSX, o que facilita o acesso e a manipulação dos dados. Em seguida, esses dados são analisados no ambiente virtual do VSCode.

3.1.1 WebScraping

Nesta etapa da extração de dados de um site para coletar informações sobre veículos, realiza-se a requisição da URL para acessar os dados da página de interesse. Para isso, utiliza-se a ferramenta `ZenRowsClient`, que contorna medidas de proteção

como CAPTCHAs e JavaScript. A URL da página é fornecida e, em seguida, o código realiza uma requisição GET, retornando os dados em formato JSON.

```
url = data_dict.get('url')
r = client.get(url)
datas = json.loads(r.text).get('SearchResults')
```

Na sequência, os dados são extraídos do JSON retornado, acessando as informações de cada veículo, como marca, modelo, ano de fabricação, quilometragem, preço, entre outras. Cada informação é extraída individualmente a partir das chaves correspondentes dentro da estrutura JSON, e os dados de cada veículo são armazenados em um dicionário.

```
make = data.get('Specification').get('Make').get('Value')
model = data.get('Specification').get('Model').get('Value')
yearf = data.get('Specification').get('YearFabrication')
price = data.get('Prices').get('Price')
```

Após a extração, os dados são organizados em um formato estruturado e adicionados a uma lista. Essa lista é constantemente atualizada à medida que os dados de cada página são processados.

Por fim, após a coleta de todos os dados, os resultados são convertidos em um DataFrame do `pandas`, facilitando a organização e análise. Esse DataFrame é então exportado para um arquivo Excel, o que permite ao usuário visualizar e analisar os dados de maneira estruturada.

```
df = pd.DataFrame(cmp)
df.to_excel(f'{start}_{end}_data.xlsx',index=False)
```

3.2 ANÁLISE EXPLORATÓRIA

Esta etapa inicial é fundamental para entender e resumir as principais características do conjunto de dados em estudo. Durante a análise exploratória, diversas técnicas estatísticas e visuais foram empregadas para identificar padrões, tendências, anomalias e possíveis relações entre as variáveis. Esse processo possibilitou uma compreensão aprofundada dos dados antes de sua aplicação em modelos de machine learning. A análise exploratória foi realizada para examinar a distribuição das variáveis, identificar valores ausentes ou discrepantes, explorar em maior profundidade as relações entre as variáveis e, em última instância, orientar as decisões sobre as técnicas analíticas mais adequadas para atingir os objetivos da pesquisa.

Dentro dessa abordagem, foi realizada uma análise detalhada dos dados com o objetivo de identificar padrões e características importantes dos anúncios dos veículos.

Primeiramente, foi analisado o volume de anúncios por região, o que ajudou a entender onde a oferta de veículos é mais expressiva. Em seguida, foi utilizado um gráfico de Pareto para identificar as marcas de veículos com maior quantidade de anúncios, destacando as mais frequentes. Esse exercício foi complementado com o cálculo do preço médio por marca, proporcionando uma visão sobre os valores médios dos veículos anunciados de acordo com o fabricante.

Além disso, a comparação do preço médio entre as regiões Sul e Sudeste foi realizada, já que essas regiões apresentaram os maiores volumes de anúncios, permitindo observar se havia diferenças significativas nos preços praticados. A análise foi aprofundada com histogramas, que ilustraram a distribuição dos anúncios em relação ao ano de fabricação, quilometragem, ano do modelo e preço dos veículos. Por fim, foi investigado qual o tamanho de motor mais comum entre os anúncios, identificando as categorias mais representativas no mercado. Essas etapas ajudaram a criar uma visão abrangente e detalhada dos dados, fornecendo insights valiosos sobre o mercado de veículos.

3.3 PRÉ-PROCESSAMENTO DE DADOS

O pré-processamento de dados é uma etapa crucial na análise, pois visa preparar os dados brutos para as fases seguintes de análise e modelagem. Durante esse processo, são realizadas várias tarefas essenciais, como a remoção de duplicatas, a eliminação de outliers e o tratamento de valores ausentes, com o objetivo de melhorar a qualidade dos dados e garantir resultados mais precisos e significativos em modelos analíticos ou de machine learning. A organização e a estruturação adequadas dos dados são fundamentais para minimizar vieses ou distorções nos resultados finais, permitindo que as análises sejam mais confiáveis e representativas. Abaixo o código em python da retirada de duplicatas de

O código realiza três etapas importantes no pré-processamento de dados: a remoção de duplicatas e a eliminação de outliers.

```
print(f"Quantidade de linhas no df bruto: {len(df)}")
quantidade_duplicadas = df.duplicated().sum()
print(f"Quantidade de linhas duplicadas: {quantidade_duplicadas}")
df = df.drop_duplicates() # Remover as linhas duplicadas do DataFrame
print(f"Quantidade de linhas no df limpo: {len(df)}")

Q1 = df['Price'].quantile(0.25)
Q3 = df['Price'].quantile(0.75)
IQR = Q3 - Q1
df = df[(df['Price'] >= Q1 - 1.5 * IQR) &
```

```
    (df['Price'] <= Q3 + 1.5 * IQR)]
len(df)

df.fillna(0, inplace=True)
```

Primeiramente, ele imprime a quantidade de linhas do DataFrame bruto, proporcionando uma visão inicial do tamanho do conjunto de dados. Em seguida, ele calcula a quantidade de duplicatas presentes no DataFrame usando a função `duplicated()` e imprime essa informação. Para garantir que os dados não contenham registros redundantes, o código remove as linhas duplicadas com o método `drop_duplicates()` e exibe o número de linhas após a remoção.

Depois, o código aplica um filtro para remover outliers na coluna Price. Utilizando o método quantile(), ele calcula o primeiro (Q1) e o terceiro (Q3) quartis do preço e, a partir do intervalo interquartílico (IQR), filtra os registros de acordo com uma margem de 1.5 vezes o IQR, removendo assim os valores extremos que possam distorcer a análise.

Por fim, devido à presença de carros elétricos, algumas linhas na coluna "tamanho motor"ficaram com valores ausentes. Para lidar com isso, o código trata os valores ‘NaN‘ no DataFrame substituindo-os por zero, utilizando o método ‘fillna()‘. Essa abordagem é adotada para garantir que os valores ausentes não impactem negativamente nas análises subsequentes ou em modelos de machine learning.

3.3.1 Remoção de duplicatas

No mercado automotivo online, é frequente que o mesmo veículo seja anunciado em várias ocasiões, o que resulta em registros duplicados dentro da mesma plataforma. Essa duplicidade pode prejudicar a análise dos dados, gerando distorções nos resultados, como o aumento artificial no volume de anúncios de determinadas marcas ou modelos. Para melhorar a qualidade dos dados e garantir que as análises fossem mais precisas, foi realizado o processo de identificação e remoção dessas duplicidades. Para isso, utilizou-se a função duplicated() do Python, que é capaz de verificar se cada linha do conjunto de dados é uma duplicata de qualquer linha anterior, considerando todas as colunas. Ao identificar essas duplicatas, elas foram removidas, deixando o conjunto de dados mais limpo e consistente, facilitando a análise subsequente e a construção de modelos mais eficazes. Esse procedimento é essencial para garantir que os resultados obtidos não sejam influenciados por repetições indesejadas e que a análise seja representativa da realidade do mercado.

3.3.2 Remoção de outliers

Para detectar os outliers, utilizou-se o método do intervalo interquartil (IQR) para analisar e definir a faixa adequada de valores na variável ‘Price‘. Inicialmente, foram calculados o primeiro quartil (Q1) e o terceiro quartil (Q3), que delimitam a concentração central dos dados. A partir deles, estabeleceu-se um limite de corte — conhecido como threshold — que considera valores dentro de 1,5 vezes o IQR além dos quartis. Qualquer valor fora desse limite foi classificado como outlier e removido do conjunto de dados. Essa estratégia visa aprimorar a integridade e a precisão dos dados para análises futuras.

3.3.3 Dados Faltantes

Excluir exemplos de treinamento com valores ausentes é uma das abordagens mais simples para lidar com dados faltantes, consistindo em remover a linha (ou coluna) onde há alguma observação ausente. No entanto, ao remover muitos dados, corremos o risco de perder informações valiosas que o modelo precisa para distinguir os atributos. No nosso caso, o número de dados faltantes é muito baixo em comparação com a grande quantidade de dados obtidos através do web scraping.

3.4 FEATURE ENGINNERING

Feature engineering é o processo de criar e transformar variáveis (ou "features") a partir dos dados brutos para melhorar o desempenho dos modelos de machine learning. Este processo envolve selecionar, modificar ou combinar características para fornecer ao modelo informações mais relevantes e úteis. Técnicas comuns incluem a normalização e padronização de dados, a criação de variáveis derivadas, e a aplicação de métodos estatísticos para reduzir a dimensionalidade. O objetivo principal do feature engineering é extrair o máximo de informação útil dos dados disponíveis, ajudando a construir modelos mais precisos e eficientes. É uma etapa crucial que pode ter um impacto significativo na performance do modelo, muitas vezes mais do que a escolha do próprio algoritmo.

A exploração dos dados teve um papel essencial no aprimoramento e detalhamento do conjunto de informações. Primeiramente, foi criada a coluna "tamanho motor", extraída das descrições dos anúncios dos veículos, pois o tamanho do motor está relacionado à potência do veículo, o que pode influenciar diretamente no preço do automóvel. Em seguida, gerou-se a coluna "região", com base nos estados indicados nos anúncios, permitindo uma análise espacial dos dados, uma vez que os preços dos veículos podem variar de acordo com a região do país. Por fim, a coluna "idade do veículo"foi calculada a partir do ano de fabricação, o que possibilitou uma melhor compreensão da relação entre a idade do veículo e o seu preço.

3.4.1 Tratamento de variáveis categóricas

Nesta etapa, as variáveis categóricas foram transformadas em variáveis dummy, também chamadas de variáveis indicadoras. Esse processo é fundamental para preparar os dados para serem utilizados em modelos de regressão e outras técnicas analíticas que exigem variáveis numéricas. Ao converter cada categoria de uma variável em uma nova variável binária (0 ou 1), é possível representar de forma eficiente informações qualitativas em um formato quantitativo. Essa transformação permite que os modelos analíticos tratem as categorias de maneira independente, sem implicar uma ordem ou hierarquia entre elas, além de melhorar a performance e a precisão das análises, especialmente em algoritmos de machine learning. Essa conversão também facilita a detecção de padrões e relações entre as variáveis, proporcionando uma melhor interpretação dos dados.

O Tratamento de variáveis categóricas envolveu a transformação das variáveis categóricas em variáveis dummy, utilizando o método One-Hot Encoding através do `OneHotEncoder`. Isso converteu variáveis como *Make*, *Model*, *Transmission*, *City*, *State* e *regiao_pais* em colunas binárias, onde cada categoria foi representada por um valor de 0 ou 1. Essa transformação é essencial para que essas variáveis possam ser utilizadas de forma eficaz em modelos de machine learning, que exigem entradas numéricas.

Além disso, as variáveis numéricas, como *Vehicle_Age*, *Year Model*, *Milage*, *tamanho motor*, *flex*, *eletrico* e *hibrido*, passaram por um processo de normalização utilizando o `StandardScaler`. Esse método ajusta os dados para que possuam uma média igual a zero e um desvio padrão igual a um, garantindo que as variáveis numéricas estejam na mesma escala. A normalização é fundamental para melhorar o desempenho de modelos de machine learning, evitando que variáveis com escalas maiores dominem o processo de aprendizado.

Abaixo o código onde é realizado o pré-processamento

```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ]
)
```

A principal razão para utilizar o StandardScaler é a unificação da escala das variáveis. Em muitos casos, os dados podem ter escalas muito diferentes entre si. Por exemplo, uma variável como "idade do veículo"pode variar de 0 a 100, enquanto a variável "preço"pode variar de milhares a milhões. Se essas variáveis forem usadas em modelos de machine learning sem qualquer transformação, a variável com os valores

maiores pode dominar o aprendizado do modelo, distorcendo a análise e levando a um desempenho ruim. Com o uso do StandardScaler, as variáveis passam a ter a mesma importância e a análise é realizada de maneira mais

A padronização de um conjunto de dados é uma exigência comum para muitos estimadores de machine learning: eles podem ter um desempenho ruim se as características individuais não se assemelharem a dados distribuídos normalmente (por exemplo, uma distribuição Gaussiana com média zero e variância unitária) (SCIKIT-LEARN, 2024).

3.5 TREINAMENTO DOS MODELOS

A segmentação dos dados em conjuntos de treino e teste é uma etapa essencial no desenvolvimento de modelos de machine learning. Esse processo consiste em dividir o conjunto de dados disponível em duas partes: uma para o treinamento e outra para o teste. Os dados de treino são usados para ajustar e calibrar o modelo, permitindo que ele identifique padrões e relações nos dados. Já os dados de teste, que permanecem separados durante o treinamento, são utilizados para avaliar o desempenho do modelo em dados desconhecidos. Essa separação é fundamental para verificar a capacidade de generalização do modelo, evitando que ele apenas "decore"os dados de treino. Um balanceamento adequado entre as proporções de treino e teste, como 70-30 ou 80-20, assegura uma avaliação precisa, ajudando a identificar problemas como overfitting ou underfitting.

A separação dos dados para treinamento e teste foi realizada utilizando a técnica de Holdout, sem estratificação. Primeiramente, as variáveis preditoras (features) foram definidas, incluindo informações como "Make", "Model", "Milage", "Transmission", entre outras, e a variável alvo foi definida como "Price". Isso foi feito com o seguinte código:

```
features = [
    "Make", "Model", "Year Febrication", "Year Model", "Milage",
    "Transmission", "City", "State", "tamanho motor", "flex",
    "eletrico", "hibrido", "regiao_pais"
]
target = "Price"
```

Após isso, os dados foram filtrados para garantir que apenas anos com pelo menos dois anúncios fossem mantidos, utilizando o seguinte código:

```
year_counts = df['Year Febrication'].value_counts()
valid_years = year_counts[year_counts >= 2].index
filtered_df = df[df['Year Febrication'].isin(valid_years)]
```

Com a base de dados filtrada, as variáveis independentes (X) e dependentes (y) foram separadas com o código:

```
X = filtered_df[features]
y = filtered_df[target]
```

Por fim, o conjunto de dados foi dividido em dois subconjuntos: um para treinamento (X_train e y_train) e outro para teste (X_test e y_test). A divisão foi feita de forma aleatória, com 20% dos dados alocados para teste e 80% para treinamento, utilizando a função `train_test_split` do Scikit-learn:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

É comum usar 80% dos dados para treinamento e reter 20% para teste. No entanto, isso depende do tamanho do conjunto de dados: se ele contiver 10 milhões de instâncias, reter 1% significa que seu conjunto de teste conterá 100.000 instâncias, provavelmente mais do que o suficiente para obter uma boa estimativa do erro de generalização (GÉRON, 2022).

Após separar os dados em conjuntos de treinamento e validação, passamos para o treinamento dos modelos de Machine Learning. Foram selecionados três modelos candidatos: XGBoost, Random Forest e Lasso Regression, para identificar os padrões nos dados.

Os três modelos foram treinados, cada modelo foi colocado em uma pipeline e o desempenho de cada modelo foi avaliado utilizando a função `evaluate_model`. A seguir estão os parâmetros utilizados para cada modelo, mesmo que sejam os valores padrão:

```
models = {
    "Lasso Regression (Default)": Lasso(random_state=42),
    "Random Forest (Default)": RandomForestRegressor(random_state=42),
    "XGBoost (Default)": XGBRegressor(random_state=42)
}
```

- **Lasso Regression (Default)**:
  - `alpha`: 1.0 (valor padrão, controle da regularização)
  - `max_iter`: 1000 (valor padrão, número máximo de iterações)

  - `random_state`: 42 (valor padrão, para reprodutibilidade)

- **Random Forest (Default)**:
  - `n_estimators`: 100 (valor padrão, número de árvores na floresta)
  - `max_depth`: None (valor padrão, profundidade máxima das árvores)
  - `min_samples_split`: 2 (valor padrão, número mínimo de amostras necessárias para dividir um nó)
  - `min_samples_leaf`: 1 (valor padrão, número mínimo de amostras em uma folha)
  - `random_state`: 42 (valor padrão, para reprodutibilidade)

- **XGBoost (Default)**:
  - `n_estimators`: 100 (valor padrão, número de árvores no modelo)
  - `learning_rate`: 0.1 (valor padrão, taxa de aprendizado)
  - `max_depth`: 6 (valor padrão, profundidade máxima das árvores)
  - `random_state`: 42 (valor padrão, para reprodutibilidade)

Então foi criada uma função `evaluate_model` para treinar e avaliar os modelos. Esta função treina o modelo, faz previsões sobre o conjunto de testes e calcula três métricas de avaliação: RMSE (Root Mean Squared Error), R^2 (coeficiente de determinação) e MAPE (Mean Absolute Percentage Error):

```
def evaluate_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    r2 = r2_score(y_test, y_pred)
    mape = mean_absolute_percentage_error(y_test, y_pred)
    return rmse, r2, mape
```

Assim, os modelos foram treinados e avaliados de maneira simples, sem qualquer ajuste de hiperparâmetros, e seus desempenhos foram consolidados para análise.

O processo de validação cruzada foi utilizado no `RandomizedSearchCV` para otimizar os hiperparâmetros dos modelos de regressão. A validação cruzada é uma técnica para avaliar a performance de um modelo em um conjunto de dados de forma mais robusta, evitando problemas como overfitting ou underfitting.

No código, o processo de validação cruzada é executado dentro do `RandomizedSearchCV`, como mostrado a seguir:

```
search = RandomizedSearchCV(
    pipeline, param_distributions, n_iter=5, cv=2, scoring='neg_mean_squared_error'
    random_state=42, n_jobs=-1
)
```

O parâmetro `cv=2` dentro do `RandomizedSearchCV` indica o número de divisões (folds) na validação cruzada. Isso significa que os dados de treinamento foram divididos em 2 partes (folds), onde, a cada iteração, o modelo é treinado em uma parte e testado na outra. Esse processo é repetido para garantir que cada parte dos dados seja usada tanto para treino quanto para teste.

Dessa forma, a validação cruzada com 2 folds foi escolhida, e o modelo foi ajustado e avaliado de forma mais confiável em termos de desempenho geral, já que as métricas de desempenho são calculadas com base em múltiplas divisões dos dados.

3.6 OTIMIZAÇÃO DE HIPERPARÂMETROS

A otimização de hiperparâmetros é uma etapa importante no processo de construção de modelos de aprendizado de máquina. Ela envolve a busca pelos melhores valores para os parâmetros do modelo, que não são aprendidos diretamente a partir dos dados, mas definidos antes do treinamento. A escolha dos hiperparâmetros pode influenciar significativamente a performance do modelo, como sua capacidade de generalização, precisão e eficiência.

Para aplicar essa técnica, utilizamos os parâmetros padrão dos modelos: param_distributions_rf para o Random Forest, param_distributions_xgb para o XGBoost e param_distributions_lasso para a Regressão Lasso. O método Random Search é então utilizado para encontrar os melhores hiperparâmetros para cada um desses modelos.

Após a definição das distribuições de parâmetros, é criada uma pipeline que integra o pré-processamento dos dados e o modelo de aprendizado de máquina, o que assegura uma execução organizada e eficiente.

Por fim, o `RandomizedSearchCV` é configurado para realizar a busca aleatória pelos melhores hiperparâmetros, considerando as distribuições especificadas. Esse processo busca otimizar o modelo de forma eficiente, levando em conta a combinação ideal de parâmetros para o melhor desempenho.

Abaixo, segue o código com cada uma dessas etapas mencionadas.

```
# Random Search para Random Forest, XGBoost e Lasso
optimized_results = []
for model_name, param_distributions, model in [
    ("Random Forest (Optimized)", param_distributions_rf,
```

```
    RandomForestRegressor(random_state=42)),

    ("XGBoost (Optimized)", param_distributions_xgb,
    XGBRegressor(random_state=42)),

    ("Lasso Regression (Optimized)", param_distributions_lasso,
    Lasso(random_state=42))
]:
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                               ('regressor', model)])

    search = RandomizedSearchCV(
        pipeline, param_distributions, n_iter=5, cv=2,
        scoring='neg_mean_squared_error', random_state=42, n_jobs=-1
    )

    search.fit(X_train, y_train)
    best_model = search.best_estimator_

    rmse, r2, mape = evaluate_model(best_model, X_train, y_train,
    X_test, y_test)

    optimized_results.append([model_name, rmse, r2, mape])
```

3.7 VERIFICAÇÃO DA ACURÁCIA DOS MODELOS

Na etapa dedicada à interpretação dos modelos, buscamos compreender como as decisões de precificação eram tomadas com base nos atributos e identificamos os fatores mais importantes que influenciam o preço dos veículos. Após realizar os ajustes nas etapas anteriores, a análise dos resultados envolveu a comparação das métricas obtidas, como RMSE, R^2 e MAPE, para selecionar o modelo com a melhor capacidade de previsão. Além de comparar diretamente os modelos, também avaliamos o impacto dos ajustes de hiperparâmetros em relação aos modelos não ajustados, permitindo identificar a eficácia das modificações e escolher o modelo com o melhor desempenho geral.

Foi desenvolvida também uma aplicação prática para ilustrar a utilização dos modelos. Selecionamos três veículos distintos e inserimos suas características. Com base nessas informações, realizamos previsões de preços para cada carro utilizando o modelo treinado.

Em seguida, comparamos essas previsões com anúncios reais de veículos

semelhantes disponíveis no mercado, avaliando a precisão e confiabilidade das estimativas. Essa comparação foi fundamental para testar o desempenho do modelo em situações do mundo real.

4 RESULTADOS

Neste capítulo, serão exibidos os resultados de cada fase da metodologia. Todos os modelos de aprendizado de máquina foram ajustados utilizando o mesmo conjunto de dados, contemplando as etapas de treinamento, teste e validação. Além disso, será feita uma avaliação da relevância de cada atributo em relação à variável alvo, analisada em diferentes modelos. Os resultados serão apresentados por meio de gráficos, mostrando a diferença percentual entre os valores previstos e os observados, além de uma tabela com indicadores de desempenho, como RMSE, R^2 e MAPE.

4.1 COLETA DE DADOS

Os dados foram coletados utilizando técnicas de *web scraping* no site Webmotors, selecionado por oferecer um maior volume de anúncios e uma gama mais ampla de atributos disponíveis para extração, quando comparado a outras plataformas semelhantes. O objetivo desse projeto foi extrair informações detalhadas sobre veículos listados no site e organizar esses dados em um arquivo Excel para análise posterior. Para realizar a coleta, o código utilizou diversas bibliotecas, como `requests` e `BeautifulSoup`, para envio de requisições HTTP e manipulação de HTML, além de `pandas` para organizar e armazenar os dados em um formato tabular. A API `ZenRows` também foi integrada ao processo, permitindo contornar bloqueios do site e aumentar a eficiência da coleta.

Primeiramente, a função de coleta de dados foi configurada para acessar URLs específicas de páginas de resultados, de onde extraía informações relevantes sobre cada veículo. A função analisava dados retornados no formato JSON e extraía atributos-chave, como marca, modelo, versão, ano de fabricação, quilometragem, cidade do vendedor e preço. Cada veículo era então armazenado em uma lista de dicionários, onde cada entrada continha as informações completas de um anúncio individual.

Em seguida, o código solicita ao usuário uma URL inicial e a converte para um formato utilizável pela API. Com essa URL, o número total de páginas contendo anúncios de veículos é calculado, e o usuário pode definir um intervalo específico de páginas a serem raspadas. Para cada página dentro do intervalo especificado, uma URL é gerada e adicionada a uma lista que contém todos os links a serem processados, facilitando a coleta sequencial.

Para otimizar a velocidade de processamento, o código implementa o uso de múltiplas threads através da função `ThreadPoolExecutor`, realizando requisições simultâneas a várias páginas de anúncios. O processamento paralelo permitiu uma coleta mais rápida e eficiente, reduzindo significativamente o tempo necessário para coletar um grande volume de dados.

Por fim, os dados coletados são organizados em um `DataFrame` da biblioteca

`pandas` e exportados para um arquivo Excel, nomeado de acordo com o intervalo de páginas coletadas. Esse processo completo de coleta foi estruturado para ser robusto e eficiente, evitando bloqueios e otimizado para acessar e extrair dados de múltiplas páginas de anúncios. A abordagem paralela e o uso da API `ZenRows` foram fatores críticos para garantir o sucesso e a escalabilidade da operação.

Coluna	**Valores não nulos**	**Tipo de Variável**	**Descrição**
Make	440280	object	Marca
Model	440280	object	Modelo
Verification	440280	object	Descrição
Year Febrication	440280	int64	Ano de fabricação
Year Model	440280	int64	Ano do modelo
Milage	440280	int64	Quilometragem rodada
Transmission	440280	object	Tipo de transmissão
Price	440280	float64	Preço
City	440280	object	Cidade do anúncio
State	440280	object	Estado do anúncio

Tabela 1 – Informações das colunas do DataFrame

4.2 PRÉ-PROCESSAMENTO DOS DADOS

Neste processo, foi realizada uma limpeza de dados com o objetivo de remover duplicidades e outliers, melhorando a qualidade do conjunto de dados para futuras análises.

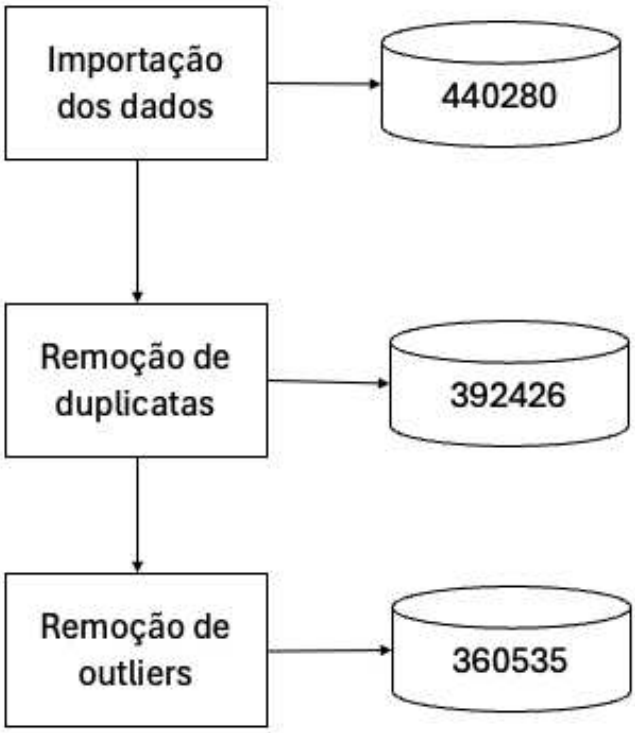
Primeiramente, foi verificada a quantidade total de registros no *DataFrame* bruto, contendo 440.280 linhas. Em seguida, calculou-se o número de linhas duplicadas, totalizando 47.854 registros repetidos. Esses registros duplicados foram então removidos, resultando em um *DataFrame* com 392.426 linhas, agora sem duplicidades.

Após a remoção das duplicatas, foi realizada uma análise de *outliers* para a variável de interesse, `Price`. Utilizou-se a técnica do intervalo interquartílico (IQR) para identificar e excluir valores anômalos. Foram calculados o primeiro quartil (Q1) e o terceiro quartil (Q3) dos preços, e o intervalo interquartílico foi definido como IQR = $Q3 - Q1$. Em seguida, os registros foram filtrados para manter apenas os valores de `Price` que estivessem dentro do intervalo $[Q1 - 1.5 \times \text{IQR}, Q3 + 1.5 \times \text{IQR}]$.

Após a exclusão dos *outliers*, o número final de registros no *DataFrame* foi reduzido para 360.535 linhas. Esta etapa de limpeza de dados foi essencial para garantir que o conjunto de dados esteja consistente e livre de influências extremas, que poderiam distorcer análises e modelos preditivos subsequentes.

A partir da categoria "Make", foi possível realizar um novo agrupamento dos dados, identificando as principais marcas de carro. Entre as 119 marcas analisadas, apenas 11 delas representam 80% dos anúncios online.

Figura 10 – Fluxograma com visão do tratamento de dados.



Fonte: Autor (2024)

No processo de preparação dos dados, foram criadas colunas adicionais para enriquecer as informações contidas no *DataFrame* e permitir análises mais detalhadas sobre as características dos veículos.

Primeiramente, foi criada uma nova coluna chamada `tamanho motor`. Esta coluna foi preenchida com informações extraídas da coluna `Verification`, onde valores numéricos com casas decimais, que representam o tamanho do motor, foram identificados e extraídos por meio de uma expressão regular.

Em seguida, foram criadas três colunas binárias para indicar o tipo de combustível dos veículos: `flex`, `eletrico` e `hibrido`. A coluna `flex` recebeu o valor 1 quando a palavra "FLEX"estava presente na coluna `Verification` e 0 caso contrário, indicando se o veículo possui motorização flexível. De maneira semelhante, a coluna `eletrico` foi preenchida com o valor 1 quando a palavra "ELÉTRICO"estava presente, e 0 caso contrário, sinalizando se o veículo é elétrico. Da mesma forma, a coluna `hibrido` foi configurada para indicar a presença de motorização híbrida, atribuindo o valor 1 se "HÍBRIDO"estivesse presente e 0 caso contrário.

Por fim, foi criada uma coluna adicional chamada `regiao_pais` para representar a região geográfica correspondente ao estado de origem do veículo. A coluna `State`

foi primeiramente ajustada para extrair a sigla do estado, que estava entre parênteses, utilizando uma expressão regular. Em seguida, um dicionário foi aplicado para mapear cada sigla de estado para sua respectiva região do país, como Norte, Nordeste, Sudeste, Sul e Centro-Oeste.

Essas novas colunas enriqueceram o *DataFrame*, permitindo uma análise mais granular e detalhada das características dos veículos, incluindo tipo de combustível, tamanho do motor e região geográfica.

Para analisar a distribuição dos estados brasileiros de acordo com suas regiões, foi realizada uma contagem dos anúncios por região utilizando a coluna `regiao_pais` do *DataFrame*. Esta coluna permitiu identificar quantos anúncios pertencem a cada uma das cinco grandes regiões do Brasil: Norte, Nordeste, Sudeste, Sul e Centro-Oeste.

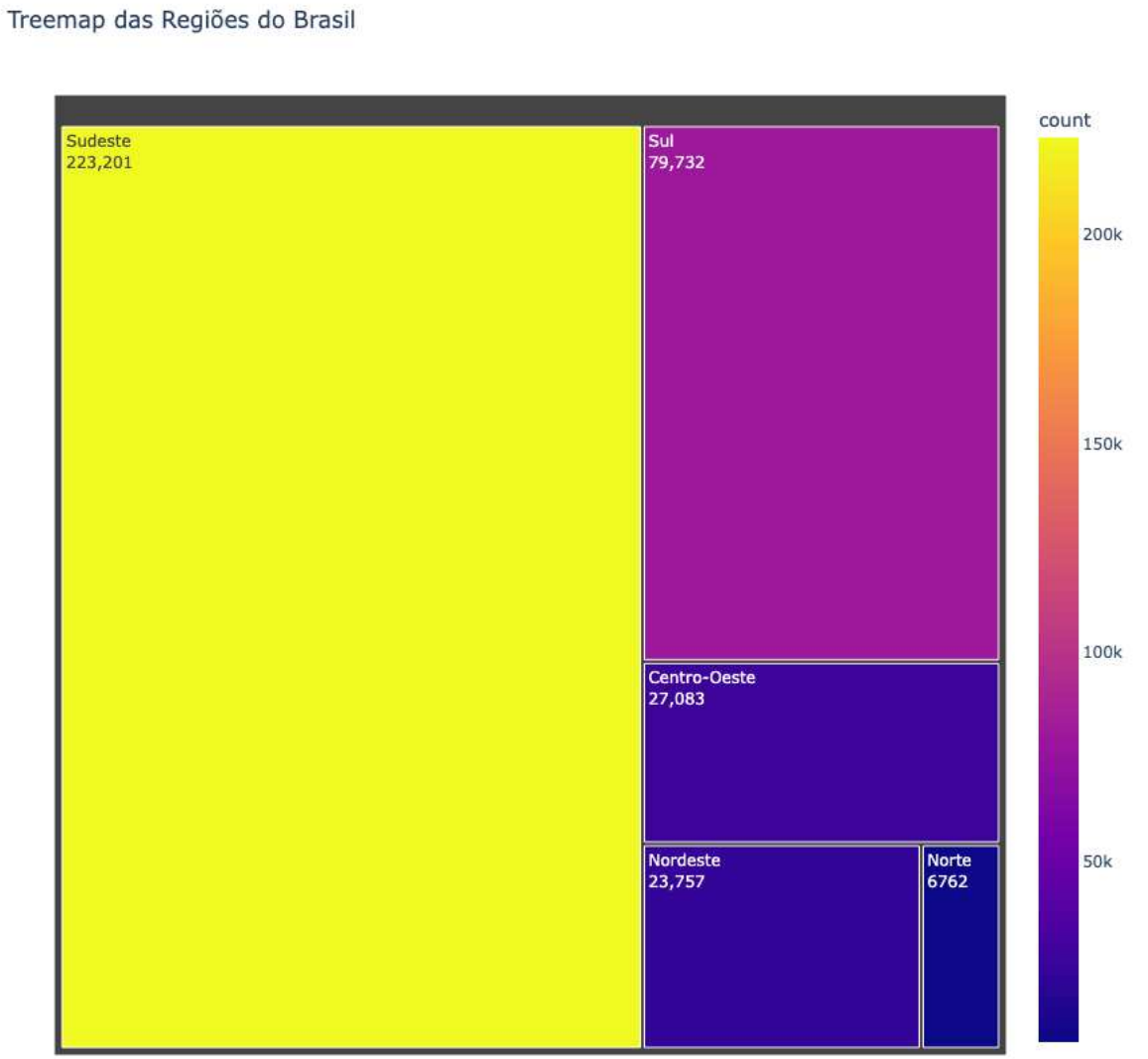
A região Sudeste destaca-se como a área com a maior concentração de veículos, totalizando 223.201 registros. Esse retângulo é o maior do gráfico e está representado em um tom de amarelo brilhante, indicando a predominância da região em relação à quantidade de veículos. Em seguida, a segunda maior concentração de veículos está no Sul, com 79.732 registros. Embora significativamente menor do que o Sudeste, a região Sul ainda ocupa uma área expressiva no gráfico, com uma cor intermediária entre o amarelo e o roxo.

O Centro-Oeste aparece em terceiro lugar, com 27.083 registros, ocupando um espaço bem menor em comparação às regiões Sudeste e Sul, o que reflete uma menor participação no número total de veículos. A região Nordeste, com 23.757 registros, possui uma participação ainda menor e é representada em uma cor mais escura, evidenciando a menor quantidade de veículos em relação às outras grandes regiões. Por fim, a região Norte, com 6.762 registros, apresenta a menor concentração de veículos entre todas as regiões e é representada em um tom de azul escuro, ocupando o menor espaço no gráfico.

Abaixo, o gráfico 12 ilustra a distribuição de veículos por marca, destacando as marcas que, juntas, acumulam até 80% do total de veículos no conjunto de dados. Observa-se que a marca Volkswagen ocupa a primeira posição, representando 16,3% do total, seguida por Chevrolet (12,8%) e Fiat (12,7%). Essas três marcas somadas já compreendem uma parte substancial do total de veículos.

À medida que percorremos o gráfico da esquerda para a direita, vemos uma diminuição gradual na porcentagem de cada marca, enquanto a linha vermelha que representa o percentual acumulativo sobe até atingir 80%. As marcas listadas nas posições finais, como Honda (4,5%) e Nissan (3,2%), completam esse limite de 80%. Este gráfico permite visualizar rapidamente quais marcas possuem maior representatividade e como elas contribuem cumulativamente para o total, fornecendo uma visão clara da concentração de veículos por marca no conjunto de dados.

Figura 11 – Treemap das Regiões do Brasil.

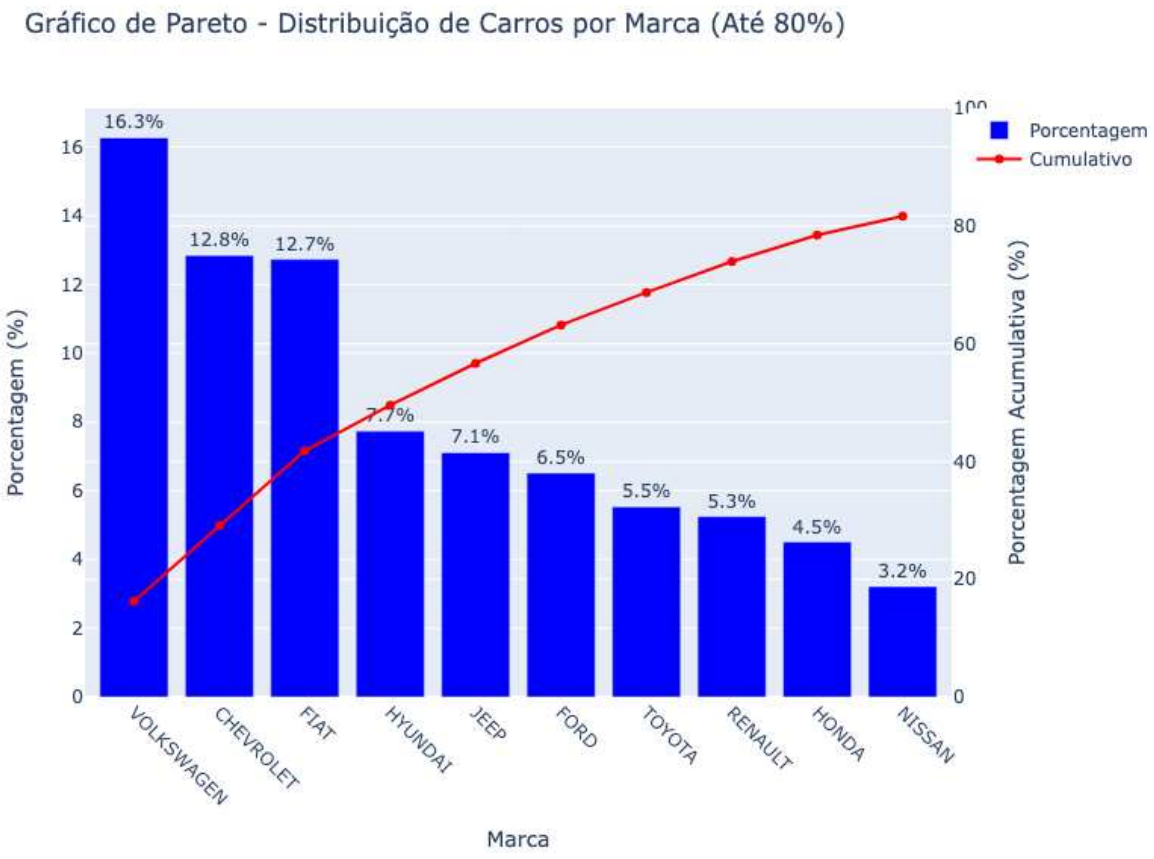


Fonte: Autor (2024)

P gráfico 13 apresenta uma comparação do preço médio de veículos das principais marcas nas regiões brasileiras com maior número de anúncios, o Sul e o Sudeste. As barras no gráfico são segmentadas por cor, onde a cor azul representa a região Sudeste e a cor laranja representa a região Sul. O gráfico destaca as 10 marcas mais populares, mostrando o preço médio, além dos valores máximos, mínimos e o desvio padrão de cada uma nas regiões Sul e Sudeste. Isso proporciona uma visão clara das similaridades e diferenças de preços entre essas localidades, permitindo uma análise mais detalhada da variação de preços para cada marca.

No topo da lista, a marca BMW apresenta o preço médio mais elevado, com

Figura 12 – Distribuição de Carros por Marca.



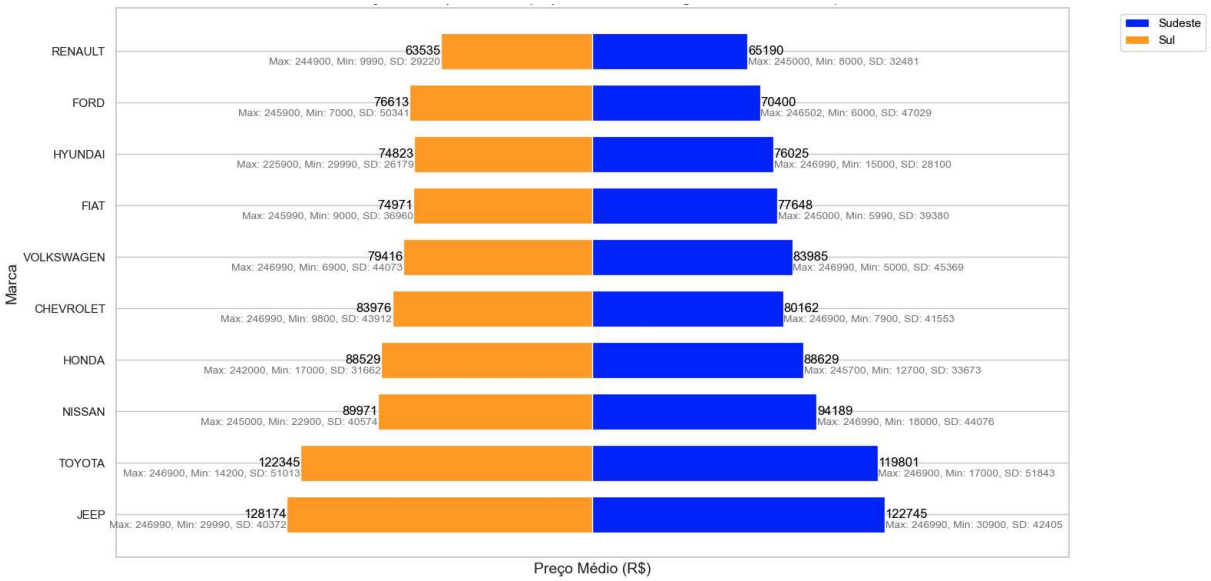
Fonte: Autor (2024)

R$ 654.969 no Sul e R$ 410.990 no Sudeste, indicando uma diferença considerável de preços entre as regiões. Outras marcas de alto valor médio incluem Chevrolet e Renault, com o preço médio no Sudeste (R$ 418.873 e R$ 344.954, respectivamente) sendo superior ao da região Sul.

Observa-se que algumas marcas populares, como Honda, Ford, Volkswagen, e Nissan, têm preços médios mais baixos em comparação às marcas de luxo, com variações menos expressivas entre as regiões. Marcas como Hyundai e Fiat também mostram uma disparidade significativa de preços entre Sul e Sudeste, sugerindo uma variação na demanda ou oferta regional para esses veículos. Este gráfico fornece uma visão clara das diferenças de preço por marca entre as duas regiões, sendo uma ferramenta útil para identificar tendências e disparidades regionais no mercado de veículos.

A Figura X apresenta uma série de histogramas que ilustram a distribuição das principais variáveis do conjunto de dados de veículos analisados nesta pesquisa. Os histogramas foram dispostos em uma estrutura 2x3, permitindo uma visão geral das

Figura 13 – Preço Médio por Marca (Top 10 Marcas - Regiões Sul e Sudeste).



Fonte: Autor (2024)

características dos veículos, incluindo ano de fabricação, quilometragem, ano-modelo, preço e tamanho do motor. Essa disposição facilita a análise comparativa entre as variáveis e permite identificar padrões gerais no conjunto de dados.

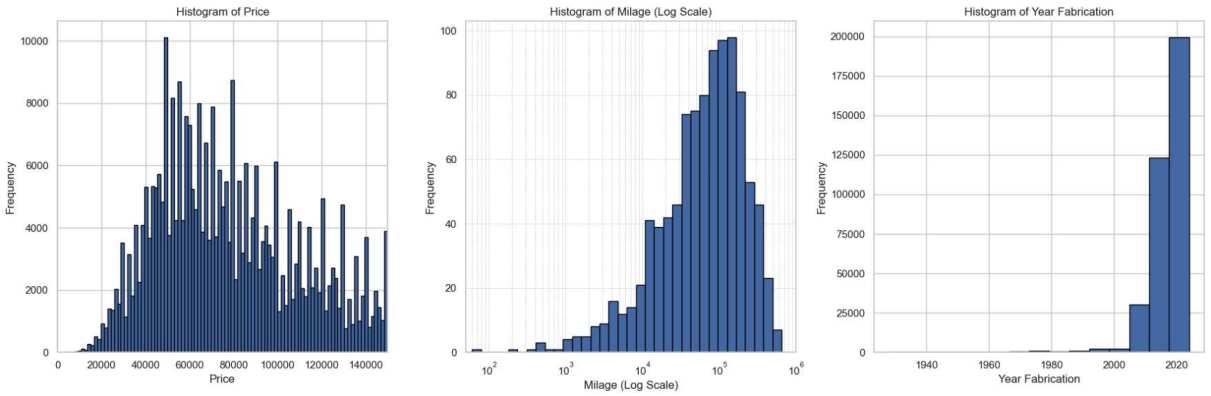
Em relação às características dos veículos, os histogramas mostram uma concentração expressiva de modelos mais recentes, com baixa quilometragem e preços majoritariamente acessíveis, embora existam algumas exceções com valores mais altos. A distribuição do ano de fabricação e do ano-modelo indica que a maioria dos veículos disponíveis são de fabricação recente, o que sugere uma preferência do mercado por veículos mais novos. A análise da quilometragem mostra uma maior frequência de veículos com pouco uso, embora existam *outliers* que indicam veículos com quilometragem mais alta.

Quanto ao preço, observa-se uma predominância de veículos com valores na faixa mais baixa, o que pode refletir a realidade do mercado de veículos populares, embora o conjunto de dados também inclua uma pequena quantidade de veículos de maior valor, que formam uma cauda longa na distribuição de preços.

Esses histogramas oferecem uma visão abrangente das principais variáveis dos veículos analisados, destacando tendências e padrões que ajudam a compreender o perfil dos veículos disponíveis no mercado. Esses *insights* servem de base para as análises subsequentes e contribuem para uma melhor interpretação das preferências e demandas do mercado automotivo.

A análise do tamanho do motor revela uma preferência por veículos com motores de menor cilindrada, possivelmente influenciada por fatores econômicos e de eficiência

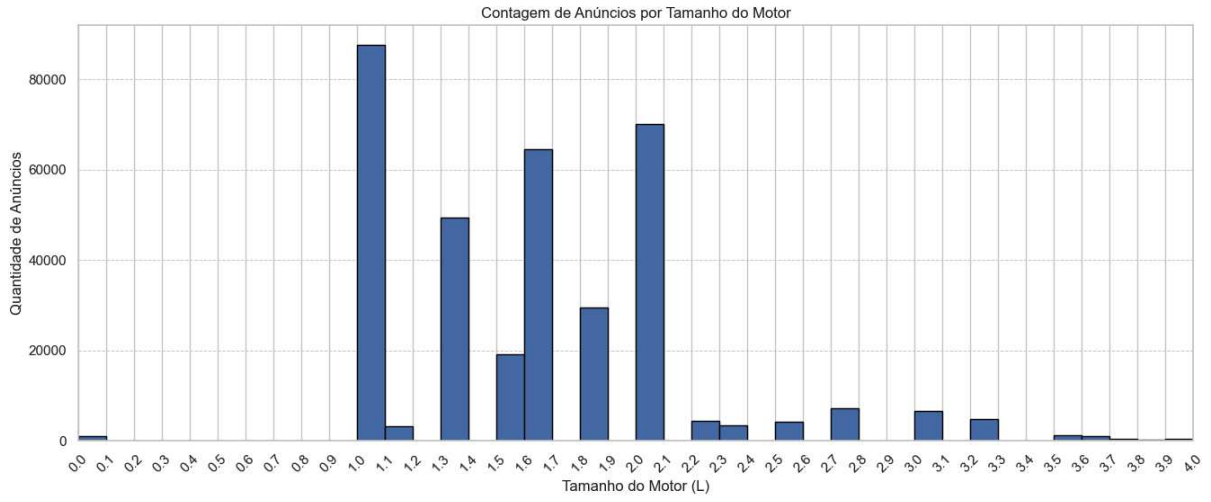
Figura 14 – Histograma das variáveis.



Fonte: Autor (2024)

de combustível.

Figura 15 – Distribuição de anúncios por tamanho do motor.



Fonte: Autor (2024)

4.2.1 SEPARAÇÃO DOS DADOS EM TREINO E TESTE

Para a execução dos experimentos de modelagem, os dados foram divididos em dois conjuntos: treino e teste. Essa divisão foi realizada com o intuito de garantir que o modelo pudesse ser treinado e posteriormente validado em uma amostra que não fosse utilizada durante o processo de ajuste, proporcionando uma avaliação imparcial de seu desempenho.

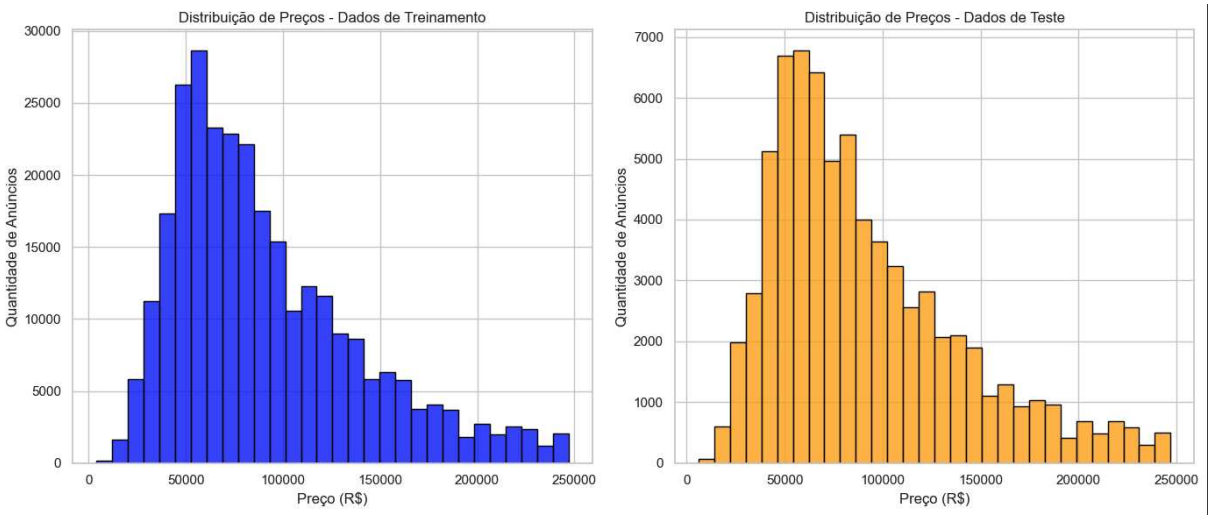
Durante esse processo, as variáveis independentes, que representam as características utilizadas para prever o preço do veículo, foram atribuídas ao conjunto X,

enquanto a variável dependente, o preço, foi atribuída ao conjunto *y*. A divisão seguiu uma proporção de 80% dos dados para o conjunto de treino e 20% para o conjunto de teste, garantindo que o modelo dispusesse de uma quantidade suficiente de dados para aprendizado, sem prejudicar a avaliação futura.

Para assegurar a reprodutibilidade dos resultados, a divisão foi feita de forma aleatória, mas com a definição de um parâmetro de estado fixo. Isso permite que a mesma divisão seja reproduzida em execuções futuras, facilitando a comparação de resultados e proporcionando uma análise consistente do desempenho dos modelos em relação aos dados.

A separação em conjuntos de treino e teste é crucial para uma validação robusta do modelo. O conjunto de treino permite que o modelo ajuste seus parâmetros e identifique padrões a partir dos dados. Por outro lado, o conjunto de teste simula o comportamento do modelo em dados não vistos anteriormente, refletindo sua capacidade de generalização e seu potencial de aplicação em cenários reais.

Figura 16 – Distribuição dos dados de treino e teste.



Fonte: Autor (2024)

Na imagem, observamos duas distribuições de preços para os dados de treino e de teste. Ambas as distribuições seguem um padrão semelhante, com maior concentração de anúncios em preços mais baixos, especialmente na faixa de 50.000 a 100.000 reais. Esse comportamento similar indica que a separação dos dados foi realizada de forma adequada, garantindo uma representatividade equivalente entre as duas amostras. Assim, podemos inferir que o modelo a ser treinado nos dados de treino terá uma boa capacidade de generalização ao ser aplicado aos dados de teste.

4.2.2 OTIMIZAÇÃO DE HIPERPARÂMETROS

Para melhorar o desempenho dos modelos, foi aplicada a técnica de otimização de hiperparâmetros por meio do *Randomized Search Cross-Validation* (*RandomizedSearchCV*). A otimização de hiperparâmetros permite ajustar os parâmetros dos modelos de forma que eles possam alcançar um melhor ajuste aos dados e, consequentemente, melhorar as previsões.

Neste estudo, três modelos foram otimizados: *Random Forest*, *XGBoost* e *Lasso Regression*. Para cada um deles, foi definida uma distribuição de valores para os principais hiperparâmetros, que foram então testados em diferentes combinações aleatórias. A técnica de *Random Search* foi escolhida em vez do tradicional *Grid Search* devido à sua eficiência em explorar uma gama maior de valores de hiperparâmetros em menos tempo, o que é especialmente útil para modelos complexos como *Random Forest* e *XGBoost*.

Os hiperparâmetros considerados para cada modelo foram os seguintes:

- **Random Forest**: Para o modelo *Random Forest*, foram testados os seguintes valores de hiperparâmetros:
  - Número de estimadores (*n_estimators*): valores variando entre 50 e 150.
  - Profundidade máxima da árvore (*max_depth*): 10 e 20.
  - Número mínimo de amostras para divisão (*min_samples_split*): 2 e 5.
  - Número mínimo de amostras por folha (*min_samples_leaf*): 1 e 2.
  - Máximo de características usadas em cada divisão (*max_features*): 'auto' e 'sqrt'.
  - Uso de *bootstrap*: [True, False].
- **XGBoost**: Para o modelo *XGBoost*, foram otimizados os seguintes hiperparâmetros:
  - Número de estimadores (*n_estimators*): valores variando entre 50 e 150.
  - Profundidade máxima da árvore (*max_depth*): 3, 10 e 20.
  - Taxa de aprendizado (*learning_rate*): 0.01, 0.1 e 0.2.
  - Fração de amostras usadas em cada árvore (*subsample*): 0.7, 0.8 e 1.0.
- **Lasso Regression**: Para o modelo *Lasso Regression*, foi ajustado o parâmetro:
  - *Alpha*: valores variando entre 0.001 e 10 (distribuição uniforme).

A otimização foi realizada com *RandomizedSearchCV* configurado para 5 iterações de busca aleatória com validação cruzada de duas dobras (*cv=2*). Para cada

modelo otimizado, selecionou-se a melhor combinação de hiperparâmetros com base na métrica de erro quadrático médio negativo (*neg_mean_squared_error*). Em seguida, os modelos ajustados com os melhores hiperparâmetros foram avaliados nos dados de teste.

Este processo permitiu identificar configurações mais precisas para cada modelo, visando um desempenho superior e uma previsão mais confiável.

4.2.3 ANÁLISE DOS RESULTADOS

Neste estudo, avaliamos o desempenho de diversos modelos de *Machine Learning* para prever o valor de veículos com base em características como idade, tipo de transmissão, quilometragem, entre outras variáveis. Utilizamos três algoritmos: *Lasso Regression*, *Random Forest* e *XGBoost*, tanto em suas configurações padrão quanto em versões otimizadas, para entender o impacto da escolha do modelo e da otimização de hiperparâmetros sobre a precisão das previsões.

Para a configuração de pré-processamento, as variáveis numéricas foram padronizadas para garantir que todas tivessem uma escala semelhante, facilitando o aprendizado dos modelos. Além disso, as variáveis categóricas passaram por transformações de *One-Hot Encoding*, uma técnica que converte cada categoria dessas variáveis em colunas binárias separadas. Isso significa que cada categoria é representada por uma nova coluna que indica a presença ou ausência dessa categoria com os valores 1 ou 0, respectivamente. Essa transformação é essencial para que os modelos de aprendizado de máquina possam lidar adequadamente com dados categóricos. Após o pré-processamento, os dados foram divididos em conjuntos de treino e teste, e utilizamos o *Random Search* para otimizar os hiperparâmetros dos modelos. Em seguida, os modelos foram treinados e avaliados, e os resultados de cada configuração foram consolidados e apresentados.

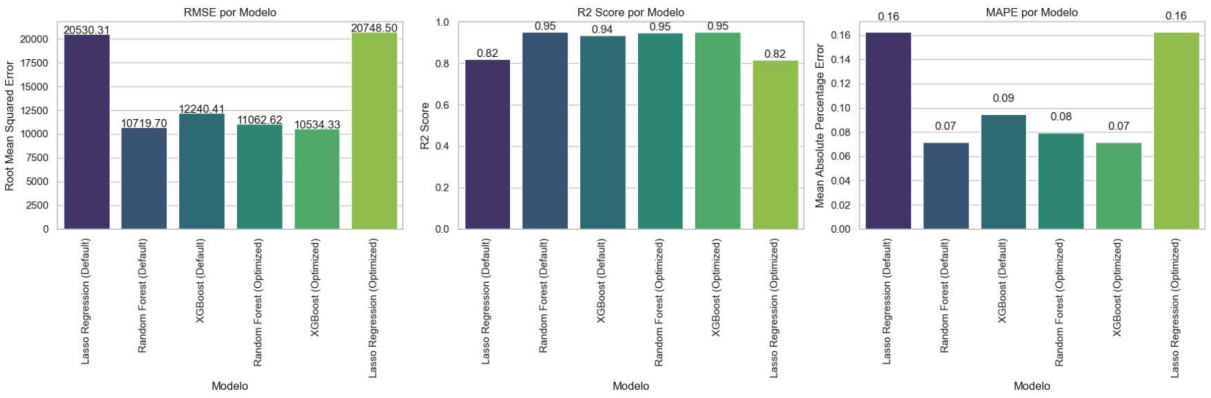
Model	**RMSE**	**R2**	**MAPE**
Lasso Regression (Default)	20530.31	0.82	16%
Random Forest (Default)	10719.70	0.95	7%
XGBoost (Default)	12240.41	0.94	9%
Random Forest (Optimized)	11062.62	0.95	8%
XGBoost (Optimized)	10534.33	0.95	7%
Lasso Regression (Optimized)	20748.50	0.82	16%

Tabela 2 – Resultados dos modelos de regressão para RMSE, R2 e MAPE

A fim de facilitar a compreensão dos desempenhos dos modelos, apresenta-se o gráfico 17.

Para avaliar a confiabilidade dos modelos apresentados, analisamos três métricas de desempenho: RMSE (*Root Mean Squared Error*), R^2 e MAPE (*Mean Absolute Percentage Error*).

Figura 17 – Desempenho dos Modelos: RMSE, R2 e MAPE.



Fonte: Autor (2024)

O RMSE mede o erro absoluto médio das previsões em relação aos valores reais; quanto menor o RMSE, melhor o ajuste do modelo aos dados, pois indica que os valores previstos estão mais próximos dos valores reais. Modelos com RMSE altos apresentam previsões com maior desvio em relação ao valor real, o que é indesejável. Nos resultados, os modelos *Random Forest* (Default e Otimizado) e *XGBoost* (Otimizado) apresentaram os menores valores de RMSE, com 10719.70 e 10534.33, respectivamente, sugerindo maior precisão e ajuste aos dados. Em contraste, o *Lasso Regression* teve RMSE mais alto, indicando menor exatidão nas previsões.

O R^2 Score indica a proporção da variação total dos dados explicada pelo modelo. Valores próximos de 1 são desejáveis, pois indicam que o modelo explica bem as variações dos dados, enquanto valores mais baixos indicam um ajuste inferior. Nos resultados, tanto o *Random Forest* quanto o *XGBoost* apresentaram R^2 Score acima de 0.95, reforçando sua alta capacidade de explicação e ajuste. Em comparação, o *Lasso Regression* mostrou desempenho inferior nesta métrica, indicando que explica menos variabilidade dos dados.

O MAPE representa o erro percentual médio, com valores menores indicando previsões mais precisas e, portanto, melhores. Um MAPE alto é indesejável, pois indica grandes discrepâncias percentuais entre as previsões e os valores reais. Nos resultados, o *Random Forest* (Default) e o *XGBoost* (Otimizado) apresentaram os menores valores de MAPE, evidenciando uma alta precisão nas previsões.

Portanto, o modelo *XGBoost* (Otimizado) se destaca como a melhor escolha, equilibrando baixo erro e alta explicabilidade, seguido de perto pelo *Random Forest* (Default).

4.2.4 EXEMPLOS DE APLICAÇÃO

Após treinar e otimizar um modelo XGBoost, que obteve o melhor desempenho, o próximo passo foi aplicá-lo para prever o valor de um veículo específico, no caso, um Toyota Corolla de 2022. As informações fornecidas sobre o carro incluíam: marca "Toyota", modelo "Corolla", transmissão "Automática", cidade "Florianópolis", estado "SC", idade do veículo de 3 ano, milhagem de 23.000 km, motor de 2.0 litros, combustível flex, e sem características de ser elétrico ou híbrido.

O modelo XGBoost otimizado foi então utilizado para realizar a previsão do valor do veículo, e o valor estimado foi de R$ 147.142. Para comparar, ao se realizar uma pesquisa sobre o preço médio de anúncios de carros com características semelhantes, como a marca, modelo, ano e outras especificações, encontrou-se uma média de R$ 152.850. Essa diferença entre o valor previsto e o valor médio dos anúncios pode ser um indicativo de que o modelo ainda possui margem para ajustes ou pode ser influenciado por fatores que não foram considerados nos dados utilizados para a previsão. Esse procedimento, no entanto, mostrou como o modelo pode ser útil para estimar o valor de veículos com base em suas características.

Figura 18 – Anúncios Toyota Corolla.

Make	Model	Verification	ar Fabric	Year Mo	Milage	Transmission	Price	City	State
TOYOTA	COROLLA	2.0 VVT-IE FLEX GR-S DIRECT SHIFT	2021	2022	28000	Automática	148900	Florianópolis	Santa Catarina (SC)
TOYOTA	COROLLA	2.0 VVT-IE FLEX ALTIS DIRECT SHIFT	2021	2022	18913	Automática	156800	Florianópolis	Santa Catarina (SC)
TOYOTA	COROLLA	2.0 VVT-IE FLEX ALTIS DIRECT SHIFT	2021	2022	18913	Automática	156800	Florianópolis	Santa Catarina (SC)

Fonte: Autor (2024)

O segundo teste foi realizado para o veículo com as seguintes características: marca “VOLKSWAGEN”, modelo “GOL”, transmissão “Manual”, cidade “Florianópolis”, estado “SC”, idade do veículo de 2 ano, milhagem de 14.000 km, motor de 1.0 litro, combustível flex, e sem características de ser elétrico ou híbrido. Ao realizar a pesquisa sobre o preço médio de anúncios de carros com essas características, a média encontrada foi de R$ 66.396. O modelo XGBoost otimizado foi utilizado para prever o valor desse veículo, resultando em um valor estimado de R$ 67.080. Essa previsão ficou ligeiramente acima da média dos anúncios, o que pode sugerir uma avaliação mais alta para o veículo de acordo com o modelo utilizado.

O terceiro e último teste foi realizado para o veículo com as seguintes características: marca “FIAT”, modelo “UNO”, transmissão “Manual”, cidade “Florianópolis”, estado “SC”, idade do veículo de 3 anos, milhagem de 52.000 km, motor de 1.0 litro, combustível flex, e sem características de ser elétrico ou híbrido. A média de preço dos anúncios de veículos com essas características foi de R$ 51.250. Utilizando o modelo XGBoost otimizado, o valor previsto para esse veículo foi de R$ 51.756, ficando

Figura 19 – Anúncios Volkswagen Gol.

Make	Model	Verification	ar Fabric	Year Mo	Milage	Transmission	Price	City	State
VOLKSWAGEN	GOL	1.0 12V MPI TOTALFLEX 4P MANUAL	2022	2023	9400	Manual	63300	Florianópolis	Santa Catarina (SC)
VOLKSWAGEN	GOL	1.0 12V MPI TOTALFLEX 4P MANUAL	2022	2023	14400	Manual	69900	Florianópolis	Santa Catarina (SC)
VOLKSWAGEN	GOL	1.0 12V MPI TOTALFLEX 4P MANUAL	2022	2023	20366	Manual	65990	Florianópolis	Santa Catarina (SC)

Fonte: Autor (2024)

bastante próximo da média dos anúncios, o que reforça a precisão do modelo para estimar valores de veículos com base em suas especificações.

Figura 20 – Anúncios Fiat Uno.

Make	Model	Verification	ar Fabric	Year Mo	Milage	Transmission	Price	City	State
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	54179	Manual	51490	Florianópolis	Santa Catarina (SC)
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	41338	Manual	49990	Florianópolis	Santa Catarina (SC)
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	45376	Manual	51970	Florianópolis	Santa Catarina (SC)
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	64490	Manual	50790	Florianópolis	Santa Catarina (SC)
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	62156	Manual	50900	Florianópolis	Santa Catarina (SC)
FIAT	UNO	1.0 FIRE FLEX ATTRACTIVE MANUAL	2021	2021	48050	Manual	52360	Florianópolis	Santa Catarina (SC)

Fonte: Autor (2024)

5 CONCLUSÃO

O objetivo deste estudo foi desenvolver um modelo de precificação para anúncios de carros usados, com base em dados coletados. Esse procedimento foi essencial para compreender o impacto de cada atributo na previsão dos preços e possibilitar a interpretação detalhada dos modelos. Os resultados obtidos mostraram um desempenho significativo em diferentes abordagens, com destaque para o modelo *Random Forest* (padrão e otimizado) e o *XGBoost* otimizado.

Entre os modelos testados, o *Random Forest* (Default) apresentou um RMSE de 10,719, um R^2 de 0,951 e um MAPE de 0,0719, evidenciando sua precisão e capacidade de explicar a variabilidade dos dados de forma eficaz. O RMSE (Root Mean Squared Error) representa a média das diferenças quadradas entre os valores reais e previstos, indicando, portanto, o erro médio absoluto das previsões. O R^2 reflete a proporção da variabilidade dos dados que é explicada pelo modelo, com valores próximos a 1 indicando alta capacidade explicativa. Já o MAPE (Mean Absolute Percentage Error) revela o erro percentual médio em relação aos valores reais, proporcionando uma medida do erro relativo das previsões. O *XGBoost* (Otimizado) também se destacou, com um RMSE ainda mais baixo de 10,534, R^2 de 0,953 e MAPE de 0,0715, consolidando-se como o modelo mais robusto neste estudo.

Os resultados confirmam que modelos de aprendizado de máquina como *Random Forest* e *XGBoost*, quando ajustados corretamente, são altamente eficazes na precificação de veículos usados. Por outro lado, a *Lasso Regression* apresentou desempenho inferior, com RMSE de 20,530 e 20,748 nas versões padrão e otimizada, respectivamente, sugerindo menor precisão e capacidade de captar a complexidade dos dados.

Essa conclusão é reforçada pela análise apresentada na Tabela 3, que compara as previsões do modelo com a média dos valores de mercado para cada veículo, além de exibir o erro percentual de previsão. Observa-se que o modelo *XGBoost* otimizado produziu estimativas próximas aos preços médios de mercado, com erros percentuais relativamente baixos: 4% para o Toyota Corolla, 1% para o Volkswagen Gol e 1% para o Fiat Uno, o que evidencia sua capacidade de oferecer previsões consistentes e alinhadas com as tendências do mercado.

Esse erro percentual reduzido sugere que o modelo consegue capturar bem as tendências do mercado, produzindo estimativas de preço consistentes para veículos com diferentes características. A precisão do modelo ao se aproximar dos valores de mercado reforça a utilidade dessa abordagem para realizar estimativas de preço em cenários reais.

Como recomendação para futuras pesquisas, apesar das limitações na disponibilidade de dados automotivos detalhados, sugere-se a inclusão de variáveis adi-

Modelo do Carro	Média Mercado	Previsão	Erro Percentual de Previsão
Toyota Corolla	R$152.850,00	R$ 147.142,00	4%
Volkswagen Gol	R$ 66.396,00	R$ 67.080,00	1%
Fiat Uno	R$ 51.250,00	R$ 51.756,00	1%

Tabela 3 – Comparação entre valores de mercado e previsões

cionais que possam capturar com mais precisão as preferências dos consumidores. Entre essas variáveis, incluem-se fatores socioeconômicos, como a renda média da região, a sazonalidade e a popularidade de determinados modelos. A consideração de informações específicas do veículo, como histórico de manutenção, número de proprietários anteriores e características opcionais não incluídas nos anúncios originais (por exemplo, sistemas de assistência de direção e histórico de acidentes), também pode contribuir para maior precisão nas previsões. Além dessas variáveis, recomenda-se alterar a milhagem para uma faixa de valores em vez de um valor único. Também é importante realizar a verificação dos intervalos de confiança para avaliar a confiabilidade do modelo.

A incorporação dessas variáveis permitirá uma visão mais aprofundada do mercado, aprimorando a capacidade do modelo de captar a complexidade dos fatores que afetam os preços dos veículos.

REFERÊNCIAS

ADETUNJI, Abigail Bola; AKANDE, Oluwatobi Noah; AJALA, Funmilola Alaba; OYEWO, Ololade; AKANDE, Yetunde Faith; OLUWADARA, Gbenle. **House price prediction using random forest machine learning technique**. *Procedia Computer Science*, Elsevier, v. 199, p. 806–813, 2022.

AJZEN, Icek. **Consumer attitudes and behavior**. *In*: *HANDBOOK of consumer psychology*. [*S.l.*]: Routledge, 2018. p. 529–552.

ANFAVEA2023. **Anuario da Industria Automobilistica Brasileira**. [*S.l.*: *s.n.*], 2023. Disponível em: https://anfavea.com.br/site/wp- content/uploads/2023/05/anuario-ATUALIZADO-2023-ALTA_compressed.pdf. Acesso em: 12 de dezembro 2023.

ASGHAR, Muhammad; MEHMOOD, Khalid; YASIN, Samina; KHAN, Zimal Mehboob. **Used cars price prediction using machine learning with optimal features**. *Pakistan Journal of Engineering and Technology*, v. 4, n. 2, p. 113–119, 2021.

B3. **Financiamento de veículos cresce 10% em 2023**. [*S.l.*: *s.n.*], 2023. Acesso em: 10 nov. 2024. Disponível em: https://www.b3.com.br/pt_br/noticias/financiamento-de-veiculos-cresce-10- em-2023.htm.

BEHRENS, John T; YU, Chong-ho. **Exploratory data analysis.** *Handbook of psychology*, v. 2, p. 33–64, 2003.

BUDIONO, Daniel Aprillio; UTOMO, Kevin Sander; WIBOWO, Kenny Jinhiro; WIRADINATA, Marcell Jeremy. **Used car price prediction model: a machine learning approach.** *International Journal of Computer and Information System (IJCIS),* v. 5, n. 1, p. 59–66, 2024.

CASOTTI, Bruna Pretti; GOLDENSTEIN, Marcelo. **Panorama do setor automotivo: as mudanças estruturais da indústria e as perspectivas para o Brasil**. *Banco Nacional de Desenvolvimento Econômico e Social*, 2008.

CHEN, Chuancan; HAO, Lulu; XU, Cong. **Comparative analysis of used car price evaluation models**. *In*: *AIP PUBLISHING, 1. AIP Conference Proceedings*. [*S.l.*: *s.n.*], 2017. v. 1839.

CHEN, Tianqi; GUESTRIN, Carlos. **Xgboost: A scalable tree boosting system**. *In*: *PROCEEDINGS of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [*S.l.*: *s.n.*], 2016. p. 785–794.

CHINTHAPATLA, Saikrishna. **Data Engineering Excellence in the Cloud: An In-Depth Exploration**. [*S.l.*]: ResearchGate, 2024.

CLOUD, Google. **What is a Data Warehouse?** [*S.l.*: *s.n.*], 2024. Acc essed: 2024-11-14. Disponível em: https://cloud.google.com/learn/what-is-a-data-warehouse?hl=en.

CUNNINGHAM, Pádraig; CORD, Matthieu; DELANY, Sarah Jane. **Supervised learning**. *In*: *MACHINE learning techniques for multimedia: case studies on organization and retrieval*. [*S.l.*]: Springer, 2008. p. 21–49.

CUTLER, Adele; CUTLER, D Richard; STEVENS, John R. **Random forests. Ensemble machine learning: Methods and applications**, Springer, p. 157–175, 2012.

DHAR, Vasant. **Data science and prediction**. *Communications of the ACM*, ACM New York, NY, USA, v. 56, n. 12, p. 64–73, 2013.

DOGUCU, Mine; ÇETINKAYA-RUNDEL, Mine. **Web scraping in the statistics and data science curriculum: Challenges and opportunities**. *Journal of Statistics and Data Science Education*, Taylor & Francis, v. 29, sup1, s112–s122, 2021.

FENAUTO2024. **A FENAUTO divulgou seu relatório mensal com as vendas de veículos seminovos e usados no mês de março com notícias positivas para o segmento**. [*S.l.*: *s.n.*], 2024. Disponível em: https://www.fenauto.org.br/news/vendas-de-carros-usados-ja-passam-dos-35- milhoes-no-trimestre. Acesso em: 9 de Junho 2024.

FUNDAÇÃO INSTITUTO DE PESQUISAS ECONÔMICAS. **Tabela Fipe – Consulta de preços de veículos**. [*S.l.*: *s.n.*], 2024. [Acessado em: 10 de novembro de 2024]. Disponível em: https://veiculos.fipe.org.br.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. 3rd. [*S.l.*]: *O'Reilly Media*, 2022. ISBN 978-1098125974.

GHAHRAMANI, Zoubin. **Unsupervised learning**. *In*: *SUMMER school on machine learning.* [*S.l.*]: Springer, 2003. p. 72–112.

GLEZ-PEÑA, Daniel; LOURENÇO, Anália; LÓPEZ-FERNÁNDEZ, Hugo; REBOIRO-JATO, Miguel; FDEZ-RIVEROLA, Florentino. **Web scraping technologies in an API world**. *Briefings in bioinformatics*, Oxford University Press, v. 15, n. 5, p. 788–797, 2014.

GOLDSTEIN, Benjamin A; NAVAR, Ann Marie; CARTER, Rickey E. **Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges.** *European heart journal*, Oxford University Press, v. 38, n. 23, p. 1805–1814, 2017.

HIGH, Robin. **Dealing with 'outliers': How to maintain your data's integrity**. *Computing News*, v. 15, n. 3, p. 14–16, 2000.

IBGE. **Frota de veículos**. [*S.l.*: *s.n.*], 2023. Disponível em: https://cidades.ibge.gov.br/brasil/pesquisa/22/0. Acesso em: 06 de maio 2023.

INFOMONEY. **Como a depreciação afeta o valor de carros seminovos?** [*S.l.*: *s.n.*], 2024. Disponível em: https://www.infomoney.com.br/colunistas/o-mundo-sobre- muitas-rodas/venda-de-carros-avanca-10-no-1o-trimestre/. Acesso em: 28 de Agosto 2024.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert *et al.* **An introduction to statistical learning**. [*S.l.*]: *Springer*, 2013. v. 112.

KAELBLING, Leslie Pack; LITTMAN, Michael L; MOORE, Andrew W. **Reinforcement learning: A survey**. *Journal of artificial intelligence research*, v. 4, p. 237–285, 1996.

KUHN, M. Applied Predictive Modeling. [*S.l.*]: *Springer*, 2013.

LI, Chenguang. **Machine Learning-Based Models for Accurate Car Prices Prediction**. *Highlights in Business, Economics and Management*, v. 40, p. 416–421, 2024.

MAHESH, Batta. **Machine learning algorithms-a review**. *International Journal of Science and Research (IJSR).[*Internet], v. 9, n. 1, p. 381–386, 2020.

MONTGOMERY, GOUGLAS *et al.* **Estatıstica aplicada e probabilidade para engenheiros**. LTC, 2009.

NAGEL, Stefan. **Machine learning in asset pricing**. [*S.l.*]: *Princeton University Press*, 2021. v. 1.

NAGLE, Thomas T. **The strategy and tactics of pricing.** 6th. [*S.l.*]: *Routledge, Taylor & Francis Group*, 2020.

NASTESKI, Vladimir. **An overview of the supervised machine learning methods**. *Horizons. b*, v. 4, n. 51-62, p. 56, 2017.

NIELSEN, Didrik. **Tree boosting with xgboost-why does xgboost win"every"machine learning competition?** 2016. Diss. (Mestrado) – NTNU.

NVIDIA. **XGBoost**. [*S.l.*: *s.n.*], 2024. Acessado: 20-nov-2024. Disponível em: https://www.nvidia.com/en-us/glossary/xgboost/.

PERSSON, Emil. **Evaluating tools and techniques for web scraping**. [*S.l.*: *s.n.*], 2019.

POTRICH, Yuri Balczareki. **Precificação de Imóveis em Florianópolis Utilizando Técnicas de Aprendizado de Máquina**. Florianópolis: [*s.n.*], 2024.

PROVOST, Foster; FAWCETT, Tom. **Data science and its relationship to big data and data-driven decision making**. *Big data*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 1, n. 1, p. 51–59, 2013.

R7. **Mesmo no pós pandemia, vendas de carro pela internet crescem no país. [*S.l.*: *s.n.*], 2024**. Disponível em: https://noticias.r7.com/prisma/autos-carros/mesmo-no-pos-pandemia-vendas- de-carro-pela-internet-crescem-no-pais-20082024/. Acesso em: 25 de Agosto 2024.

RAHM, Erhard; DO, Hong Hai *et al.* **Data cleaning: Problems and current approaches**. *IEEE Data Eng. Bull.*, v. 23, n. 4, p. 3–13, 2000.

SAMMUT-BONNICI, Tanya; CHANNON, Derek F. **Pricing strategy**. *Wiley encyclopedia of management*, Wiley Online Library, p. 1–3, 2015.

SARMENTO, Rui; COSTA, Vera. **Introduction to Linear Regression**. *In*: [*S.l.*: *s.n.*], jan. 2017. ISBN 9781522519898. DOI: 10.4018/978-1-68318-016-6.ch006.

SCIKIT-LEARN. **StandardScaler**. [*S.l.*: *s.n.*], 2024. https://scikit-learn.org/dev/modules/generated/sklearn.preprocessing.StandardScaler.html. Acessado: 2024-11-15.

SEGUI, Santi; IGUAL, Laura. **Introduction to Data Science A Python Approach to Concepts, Techniques and Applications**. [*S.l.*]: *springer publication*, 2017.

SERASA. **Pesquisa: a Relação do Brasileiro com o Automóvel**. [*S.l.*: *s.n.*], 2023. Disponível em: https://www.serasa.com.br/carteira-digital/blog/pesquisa-a- relacao-do-brasileiro-com-o-automovel/. Acesso em: 14 de Julho 2024.

SGATOYOTA. **Como a depreciação afeta o valor de carros seminovos?** [*S.l.*: *s.n.*], 2024. Disponível em: https://blog.sgatoyota.com.br/como-a-depreciacao-afeta- o-valor-de-carros-seminovos/. Acesso em: 28 de Agosto 2024.

SHAPRAPAWAD, Snehit; BORUGADDA, Premkumar; KOSHIKA, Nirmala. **Car Price Prediction: An Application of Machine Learning.** *In*: *IEEE. 2023 International Conference on Inventive Computation Technologies* (ICICT). [*S.l.*: *s.n.*], 2023. p. 242–248.

SHARMA, Hemlata; HARSORA, Hitesh; OGUNLEYE, Bayode. **An Optimal House Price Prediction Algorithm: XGBoost**. *Analytics*, MDPI, v. 3, n. 1, p. 30–45, 2024.

SHENDE, Vikram. **Analysis of research in consumer behavior of automobile passenger car customer**. *International Journal of Scientific and Research Publications*, v. 4, n. 2, p. 1–8, 2014.

TALKS, Turing. **Regressão de Ridge e Lasso.** [*S.l.*: *s.n.*], 2020. Acessado: 2024-11-20. Disponível em: https://medium.com/turing-talks/turing-talks-20- regress%C3%A3o-de-ridge-e-lasso-a0fc467b5629.

TIBSHIRANI, Robert. **Regression shrinkage and selection via the lasso**. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, Oxford University Press, v. 58, n. 1, p. 267–288, 1996.

TUFTE, Edward R; GRAVES-MORRIS, Peter R. **The visual display of quantitative information**. [*S.l.*]: *Graphics press Cheshire*, CT, 1983. v. 2.

TUKEY, John W. **Exploratory data analysis**. *Reading/Addison-Wesley*, 1977.

VANDERPLAS, Jake. **Python data science handbook: Essential tools for working with data**. [*S.l.*]: "*O'Reilly Media, Inc*.", 2016.

WEBMOTORS. **Carros Usados - Estoque**. [*S.l.*: *s.n.*], 2024. Acesso em: 10 nov. 2024. Disponível em: https://www.webmotors.com.br/carros-usados/estoque?tipoveiculo=carros-usados.

WICKHAM, Hadley; ÇETINKAYA-RUNDEL, Mine; GROLEMUND, Garrett. **R for data science**. [*S.l.*]: "*O'Reilly Media, Inc*.", 2023.

WU, Jia; CHEN, Xiu-Yun; ZHANG, Hao; XIONG, Li-Dong; LEI, Hang; DENG, Si-Hao. **Hyperparameter optimization for machine learning models based on Bayesian optimization**. *Journal of Electronic Science and Technology*, Elsevier, v. 17, n. 1, p. 26–40, 2019.

ZHANG, Ping; JIA, Yiqiao; SHANG, Youlin. **Research and application of XGBoost in imbalanced data**. *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, v. 18, n. 6, p. 15501329221106935, 2022.

ZHANG, Xiaoqing; YUAN, Xigang; WANG, Min; WANG, Yongjian; ZHANG, Dalin. **Pricing strategy for the automobile producer considering consumer anxiety behavior and policy substitution effect**. *Journal of Cleaner Production*, Elsevier, v. 446, p. 141414, 2024.

APÊNDICE A – CÓDIGOS EM PYTHON

IMPLEMENTAÇÃO DE WEB SCRAPING EM PYTHON

```
client = ZenRowsClient("<SEU_ZenRowsClient")

cmp = []

h = {
    'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
}

def get_data(data_dict):
    url = data_dict.get('url')
    ind = data_dict.get('ind')

    #r = requests.get(url,headers=h,proxies=proxyDict)
    r = client.get(url)

    datas = json.loads(r.text).get('SearchResults')

    for data in datas:
        make = data.get('Specification').get('Make').get('Value')
        model = data.get('Specification').get('Model').get('Value')
        ver = data.get('Specification').get('Version').get('Value')
        yearf = data.get('Specification').get('YearFabrication')
        yearm = data.get('Specification').get('YearModel')
        mil = data.get('Specification').get('Odometer')
        trans = data.get('Specification').get('Transmission')

        city = data.get('Seller').get('City')
        state = data.get('Seller').get('State')

        price = data.get('Prices').get('Price')

        fnl = {
            'Make':make,
            'Model':model,
            'Verification':ver,
            'Year Febrication':yearf,
```

```
            'Year Model':yearm,
            'Milage':mil,
            'Transmission':trans,
            'Price':price,
            'City':city,
            'State':state
        }

        cmp.append(fnl)

    print('======================== > > ',ind,'/',end)

url1 = input('Please Enter Url :')

uu1 = url1.replace('https://www.webmotors.com.br/','').strip()
url2 = "https://www.webmotors.com.br/"+quote(uu1,safe='')
url = f"https://www.webmotors.com.br/api/search/car?url={url2}&actualPage=1&display

links = []

r = client.get(url)
count = int(json.loads(r.text).get('Count')/24)
print(f'Total Links are {count} how many you want to scrape for now :')
start = int(input('Enter Start :'))
end = int(input('Enter End :'))
for i in range(start,end):
    url = f"https://www.webmotors.com.br/api/search/car?url={url2}&actualPage={i}&d

    links.append({
        'url':url,
        'ind':i
    })

with ThreadPoolExecutor() as executor:
    executor.map(get_data, links)
df = pd.DataFrame(cmp)
df.to_excel(f'{start}_{end}_data.xlsx',index=False)
```

SEPARAÇÃO DOS DADOS EM TREINO E TESTE

```
# Definindo as features e o target
features = [
    "Make", "Model", "Year Febrication", "Year Model", "Milage",
    "Transmission", "City", "State", "tamanho motor", "flex",
    "eletrico", "hibrido", "regiao_pais", "Vehicle_Age"
]
target = "Price"

# Filtrar anos com pelo menos dois anúncios
year_counts = df['Year Febrication'].value_counts()
valid_years = year_counts[year_counts >= 2].index
filtered_df = df[df['Year Febrication'].isin(valid_years)]

# Separando features e target
X = filtered_df[features]
y = filtered_df[target]

# Divisão Holdout sem estratificação
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

TREINANDO O MODELO E AVALIANDO ERROS

```
# Definir variáveis categóricas e numéricas para pré-processamento
categorical_features = ['Make', 'Model', 'Transmission', 'City', 'State',
                        'regiao_pais']
numeric_features = ['Vehicle_Age', 'Year Model', 'Milage',
                'tamanho motor', 'flex', 'eletrico', 'hibrido']

# Pré-processador
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'),
                                categorical_features)
    ]
)
```

```
# Função para treinar e avaliar o modelo
def evaluate_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    r2 = r2_score(y_test, y_pred)
    mape = mean_absolute_percentage_error(y_test, y_pred)
    return rmse, r2, mape

# Modelos sem otimização
models = {
    "Lasso Regression (Default)": Lasso(random_state=42),
    "Random Forest (Default)": RandomForestRegressor(random_state=42),
    "XGBoost (Default)": XGBRegressor(random_state=42)
}

# Resultados dos modelos sem otimização
results_default = []
for name, model in models.items():
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                        ('regressor', model)])
    rmse, r2, mape = evaluate_model(pipeline, X_train, y_train,
                                                X_test, y_test)
    results_default.append([name, rmse, r2, mape])

# Otimização de hiperparâmetros
param_distributions_rf = {
    'regressor__n_estimators': randint(50, 150),
    'regressor__max_depth': [10, 20],
    'regressor__min_samples_split': [2, 5],
    'regressor__min_samples_leaf': [1, 2],
    'regressor__max_features': ['auto', 'sqrt'],
    'regressor__bootstrap': [True, False]
}

param_distributions_xgb = {
    'regressor__n_estimators': randint(50, 150),
    'regressor__max_depth': [3, 10, 20],
```

```
    'regressor__learning_rate': [0.01, 0.1, 0.2],
    'regressor__subsample': [0.7, 0.8, 1.0],
}

param_distributions_lasso = {
    'regressor__alpha': uniform(0.001, 10)
}

# Random Search para Random Forest, XGBoost e Lasso
optimized_results = []
for model_name, param_distributions, model in [
    ("Random Forest (Optimized)", param_distributions_rf,
                    RandomForestRegressor(random_state=42)),
    ("XGBoost (Optimized)", param_distributions_xgb,
                    XGBRegressor(random_state=42)),
    ("Lasso Regression (Optimized)", param_distributions_lasso,
                    Lasso(random_state=42))
]:
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                        ('regressor', model)])
    search = RandomizedSearchCV(
        pipeline, param_distributions, n_iter=5, cv=2,
                        scoring='neg_mean_squared_error',
        random_state=42, n_jobs=-1
    )
    search.fit(X_train, y_train)
    best_model = search.best_estimator_
    rmse, r2, mape = evaluate_model(best_model, X_train,
                                y_train, X_test, y_test)
    optimized_results.append([model_name, rmse, r2, mape])

# Consolidar os resultados
results = results_default + optimized_results
results_df = pd.DataFrame(results, columns=["Model", "RMSE",
                                        "R2 Score", "MAPE"])
print(results_df)
```

APLICAÇÃO DOS MODELOS

Toyota Corolla

```
new_data = {
    'Make': 'Toyota',
    'Model': 'Corolla',
    'Transmission': 'Automatic',
    'City': 'Florianópolis',
    'State': 'SC',
    'regiao_pais': 'Sul',
    'Vehicle_Age': 3,
    'Year Model': 2022,
    'Milage': 23000,
    'tamanho motor': 2.0,
    'flex': 1,
    'eletrico': 0,
    'hibrido': 0
}

# Inicializar a variável para o modelo XGBoost otimizado
xgb_best_model = None

# Parâmetros de distribuição para o XGBoost
param_distributions_xgb = {
    'regressor__n_estimators': randint(50, 150),
    'regressor__max_depth': [3, 10, 20],
    'regressor__learning_rate': [0.01, 0.1, 0.2],
    'regressor__subsample': [0.7, 0.8, 1.0],
}

# Pipeline para o XGBoost
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('regressor', XGBRegress

# Random Search para o XGBoost
search = RandomizedSearchCV(
    pipeline, param_distributions_xgb, n_iter=5, cv=2, scoring='neg_mean_squared_er
    random_state=42, n_jobs=-1
)
search.fit(X_train, y_train)
```

```
# Armazenar o melhor modelo XGBoost
xgb_best_model = search.best_estimator_

# Verificar se o modelo XGBoost (Optimized) foi armazenado
if xgb_best_model is not None:
    # Preparar os dados para a previsão
    new_data_df = pd.DataFrame([new_data])

    # Realizar a previsão com o modelo XGBoost otimizado
    predicted_value = xgb_best_model.predict(new_data_df)
    print("Predicted Value:", predicted_value[0])
else:
    print("O modelo XGBoost (Optimized) não foi encontrado.")
```

VOLKSWAGEN GOL

```
new_data = {
    'Make': 'VOLKSWAGEN',
    'Model': 'GOL',
    'Transmission': 'Manual',
    'City': 'Florianópolis',
    'State': 'SC',
    'regiao_pais': 'Sul',
    'Vehicle_Age': 2,
    'Year Model': 2023,
    'Milage': 14000,
    'tamanho motor': 1.0,
    'flex': 1,
    'eletrico': 0,
    'hibrido': 0
}

# Inicializar a variável para o modelo XGBoost otimizado
xgb_best_model = None

# Parâmetros de distribuição para o XGBoost
param_distributions_xgb = {
    'regressor__n_estimators': randint(50, 150),
    'regressor__max_depth': [3, 10, 20],
```

```
    'regressor__learning_rate': [0.01, 0.1, 0.2],
    'regressor__subsample': [0.7, 0.8, 1.0],
}

# Pipeline para o XGBoost
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('regressor', XGBRegress

# Random Search para o XGBoost
search = RandomizedSearchCV(
    pipeline, param_distributions_xgb, n_iter=5, cv=2, scoring='neg_mean_squared_er
    random_state=42, n_jobs=-1
)
search.fit(X_train, y_train)

# Armazenar o melhor modelo XGBoost
xgb_best_model = search.best_estimator_

# Verificar se o modelo XGBoost (Optimized) foi armazenado
if xgb_best_model is not None:
    # Preparar os dados para a previsão
    new_data_df = pd.DataFrame([new_data])

    # Realizar a previsão com o modelo XGBoost otimizado
    predicted_value = xgb_best_model.predict(new_data_df)
    print("Predicted Value:", predicted_value[0])
else:
    print("O modelo XGBoost (Optimized) não foi encontrado.")
```

FIAT UNO

```
new_data = {
    'Make': 'FIAT',
    'Model': 'UNO',
    'Transmission': 'Manual',
    'City': 'Florianópolis',
    'State': 'SC',
    'regiao_pais': 'Sul',
    'Vehicle_Age': 3,
    'Year Model': 2021,
    'Milage': 52000,
```

```
    'tamanho motor': 1.0,
    'flex': 1,
    'eletrico': 0,
    'hibrido': 0
}

# Inicializar a variável para o modelo XGBoost otimizado
xgb_best_model = None

# Parâmetros de distribuição para o XGBoost
param_distributions_xgb = {
    'regressor__n_estimators': randint(50, 150),
    'regressor__max_depth': [3, 10, 20],
    'regressor__learning_rate': [0.01, 0.1, 0.2],
    'regressor__subsample': [0.7, 0.8, 1.0],
}

# Pipeline para o XGBoost
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('regressor', XGBRegress

# Random Search para o XGBoost
search = RandomizedSearchCV(
    pipeline, param_distributions_xgb, n_iter=5, cv=2, scoring='neg_mean_squared_er
    random_state=42, n_jobs=-1
)
search.fit(X_train, y_train)

# Armazenar o melhor modelo XGBoost
xgb_best_model = search.best_estimator_

# Verificar se o modelo XGBoost (Optimized) foi armazenado
if xgb_best_model is not None:
    # Preparar os dados para a previsão
    new_data_df = pd.DataFrame([new_data])

    # Realizar a previsão com o modelo XGBoost otimizado
    predicted_value = xgb_best_model.predict(new_data_df)
    print("Predicted Value:", predicted_value[0])
else:
```

```
print("O modelo XGBoost (Optimized) não foi encontrado.")
```