UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
ENGENHARIA DE PRODUÇÃO

Vitória Lima de Melo

ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A PREVISÃO DE PREÇOS DE VENDA DE FLATS EM JOÃO PESSOA - PB

João Pessoa – PB
2025

Vitória Lima de Melo

ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A PREVISÃO DE PREÇOS DE VENDA DE FLATS EM JOÃO PESSOA - PB

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia de Produção da Universidade Federal da Paraíba, como requisito parcial à obtenção do título de Engenheiro de Produção.

Orientador: Prof. Dr. Luciano Carlos Azevedo da Costa.

JOÃO PESSOA – PB

2025

Catalogação na publicação
Seção de Catalogação e Classificação

M528a Melo, Vitoria Lima de.
Algoritmos de Aprendizado de Máquina Para a Previsão de Preços de Venda de Flats em João Pessoa - PB / Vitoria Lima de Melo. - João Pessoa, 2025.
62 f. : il.

Orientação: Luciano Carlos Azevedo da, Costa.
TCC (Graduação) - UFPB/CT.

1. Mercado Imobiliário. 2. Precificação de Imóveis. 3. Aprendizado de Máquina. 4. Previsão de Preços. I. Costa, Luciano Carlos Azevedo da. II. Título.

UFPB/CT/BSCT CDU 658.5(043.2)

Elaborado por ONEIDA DIAS DE PONTES - CRB-CRB15/198

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

FOLHA DE APROVAÇÃO

Aluna: VITÓRIA LIMA DE MELO

Título do trabalho: ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA PREVISÃO DE PREÇOS DE VENDA DE FLATS EM JOÃO PESSOA-PB

Trabalho de Conclusão do Curso defendido e aprovado em 05/05/2025 pela banca examinadora:

gov.br Documento assinado digitalmente
LUCIANO CARLOS AZEVEDO DA COSTA
Data: 06/05/2025 15:55:20-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Luciano Carlos Azevedo Da Costa (Orientador)

Universidade Federal da Paraíba (UFPB)

gov.br Documento assinado digitalmente
HUGO HARRY FREDERICO RIBEIRO KRAMER
Data: 06/05/2025 15:02:32-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Hugo Harry Frederico Ribeiro Kramer (Examinador)

Universidade Federal da Paraíba (UFPB)

gov.br Documento assinado digitalmente
LUCAS GUEDES DE OLIVEIRA
Data: 05/05/2025 22:21:37-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Lucas Guedes de Oliveira (Examinador)

Universidade Federal da Paraíba (UFPB)

“Porque dEle, e por meio dEle, e para Ele são todas as coisas. A Ele seja a glória para sempre. Amém”

Romanos 11.36

RESUMO

O mercado imobiliário brasileiro tem apresentado crescimento contínuo, sendo esse cenário especialmente notável em João Pessoa–PB, onde o setor da construção civil tem se valorizado significativamente nos últimos anos. Nesse contexto, a precificação adequada de imóveis torna-se fundamental para garantir negociações justas que beneficiem tanto compradores quanto vendedores. Este trabalho propõe a implementação e avaliação de modelos preditivos para estimar os preços de venda de flats na cidade de João Pessoa, utilizando algoritmos de aprendizado de máquina. A partir da coleta de dados de anúncios do site Viva Real, foram aplicados e comparados sete modelos: Regressão Linear, Regressão por Vetores de Suporte, Árvore de Decisão, Floresta Aleatória, *XGBoost*, *LightGBM* e Rede Neural. Os resultados obtidos foram analisados com base em métricas de desempenho, a fim de identificar o modelo mais eficaz na previsão dos preços dos flats.

Palavras-chave: Mercado Imobiliário; Precificação de Imóveis; Aprendizado de Máquina; Previsão de Preços.

ABSTRACT

The Brazilian real estate market has shown continuous grown, and this scenario is especially notable in João Pessoa – PB, where the construction sector has significantly appreciated in recent years. In this context, accurate property pricing becomes essential to ensure fair negotiations that benefit both buyers and sellers. This study proposes the implementation and evaluation of predictive models to estimate the selling prices of flats in the city of João Pessoa, using machine learning algorithms. Based on data collected from the Viva Real website, seven models were applied and compared: Linear Regression, Support Vector Regression, Decision Tree, Random Forest, XGBoost, LightGBM and Neural Network. The results were analyzed using performance metrics to identify the most effective model for predicting flat prices.

Keywords: Real Estate Market; Property Pricing; Machine Learning; Price Prediction.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

A cidade de João Pessoa, na Paraíba, tem se destacado não apenas por suas belezas naturais e qualidade de vida, mas também pelo crescimento populacional e pelos avanços econômicos. De acordo com o Índice FipeZAP de Venda Residencial, divulgado pelo DATAZAP, João Pessoa ocupou a terceira posição entre as capitais brasileiras com maior valorização imobiliária em 2024 (Fipe, 2024, p. 15).

Segundo Luan Pereira de Souza, corretor de imóveis e economista, o mercado imobiliário local enfrenta um déficit habitacional que vem sendo gradualmente reduzido graças ao avanço da construção civil e ao crescente interesse de investidores. Essa demanda é impulsionada tanto por especuladores quanto por novos proprietários, muitos deles beneficiados por subsídios habitacionais (Souza, 2023).

A busca por qualidade de vida, o custo de vida relativamente baixo, a infraestrutura urbana e as oportunidades de emprego são fatores que contribuem para o aumento da procura por imóveis, impulsionando o setor da construção civil e aquecendo o mercado imobiliário. João Pessoa vive, assim, um ciclo virtuoso que tem movimentado a economia local. A execução de obras estruturantes pelos setores público e privado tem ampliado a capacidade da cidade de atrair novos investimentos, turistas e moradores.

Adquirir um imóvel, seja para moradia ou investimento, envolve uma série de desafios, sobretudo no que diz respeito à correta estimativa do seu valor de mercado. A falta de conhecimento técnico pode comprometer a decisão de compra. Como destaca Alencar (2022, p. 14), “é preciso cautela na hora da decisão de compra, já que o principal fator para ter um bom retorno realizando este tipo de investimento é comprar o imóvel por um valor justo ou abaixo do mercado”. Para garantir um retorno satisfatório sobre o capital investido, é essencial mensurar com precisão o valor real do imóvel negociado.

A precificação de imóveis envolve diversas variáveis como localização, tamanho, número de cômodos, facilidades do empreendimento, entre outros. Na prática, atribuir um valor a um imóvel torna-se subjetivo. Nesse cenário, surge a necessidade de ferramentas mais precisas para apoiar a precificação de imóveis, uma das abordagens mais promissoras nesse sentido é o uso do Aprendizado de Máquina (*Machine Learning*). Segundo Etham (2014, p. 3), o Aprendizado de Máquina consiste

em programar computadores para que melhorem seu desempenho com base em dados históricos, permitindo tanto a previsão de resultados futuros quanto a extração de conhecimento a partir das informações analisadas.

Ao utilizar bases de dados com registros detalhados sobre imóveis e suas características, os algoritmos de Aprendizado de Máquina são capazes de identificar padrões e relações entre variáveis que influenciam diretamente o preço de mercado, contribuindo para estimativas mais precisas e consistentes. Diversos estudos demonstram a eficácia dessa abordagem em diferentes contextos do mercado imobiliário, como na previsão de preços de aluguel, venda ou avaliação de terrenos.

Lauth (2023), por exemplo, em seu trabalho Modelo Preditivo para Preço de Aluguel de Apartamentos em Blumenau, aplicou diversos algoritmos de aprendizado de máquina, como Regressão Linear, *Random Forest*, *XGBoost* e Rede Neural, concluindo que a Regressão Linear apresentou o melhor desempenho para seu conjunto de dados. De forma semelhante, Potrich (2024), na pesquisa Precificação de Imóveis em Florianópolis Utilizando Técnicas de Aprendizado de Máquina, utilizou dados do site Viva Real e aplicou os modelos Lasso *Regression*, *Random Forest* e *XGBoost*, sendo este último o mais eficaz na previsão dos preços analisados.

Esses exemplos demonstram a relevância do Aprendizado de Máquina como ferramenta de apoio à tomada de decisão no setor imobiliário, possibilitando uma precificação mais justa, transparente e baseada em dados reais. Cabe destacar que o desempenho dos modelos pode variar de acordo com o contexto, a base de dados utilizada e as variáveis consideradas, não existindo um algoritmo único ideal para todos os cenários.

Diante desse cenário, este trabalho propõe a implementação e comparação de diferentes algoritmos de Aprendizado de Máquina para a previsão dos preços de venda de flats na cidade de João Pessoa (PB). A metodologia envolve a coleta de dados de anúncios do site Viva Real, o pré-processamento e a transformação dessas informações, seguidos da aplicação dos modelos Regressão Linear, SVR, Árvore de Decisão, Random Forest, *XGBoost*, *LightGBM* e Rede Neural. O objetivo é identificar qual modelo apresenta o melhor desempenho preditivo, contribuindo para uma abordagem mais eficiente e precisa na precificação de imóveis residenciais na região.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo principal deste trabalho é implementar e testar modelos de previsão de preços de *flats* à venda na cidade de João Pessoa/Paraíba, utilizando algoritmos de aprendizado de máquina. Para isso, serão analisados os dados de anúncio de *flats* à venda no site da Viva Real no período de 28 de janeiro de 2025 a 23 de fevereiro de 2025.

1.2.2 Objetivos Específicos

Para se alcançar o objetivo geral deste trabalho será necessário:

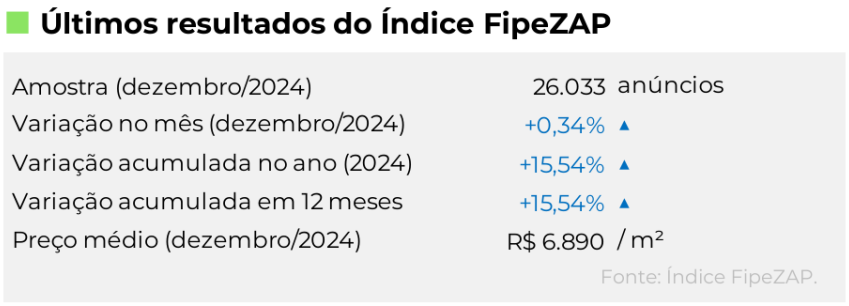
- Coletar informações consideradas importantes de preços históricos de *flats*, como a metragem, características do empreendimento, preço e outros, através do Viva Real, *site* de pesquisa imobiliária.
- Realizar o pré-processamento dos dados coletados.
- Implementar e treinar modelos de previsão de preços.
- Comparar os resultados obtidos para cada modelo.

1.3 Justificativa

A precificação de imóveis é uma tarefa complexa que envolve a análise de uma ampla gama de variáveis, tanto quantitativas — como área útil, número de quartos e valor do metro quadrado — quanto qualitativas, como localização, percepção de segurança e infraestrutura do entorno. Em João Pessoa (PB), essa complexidade é intensificada pelo forte dinamismo do mercado imobiliário local.

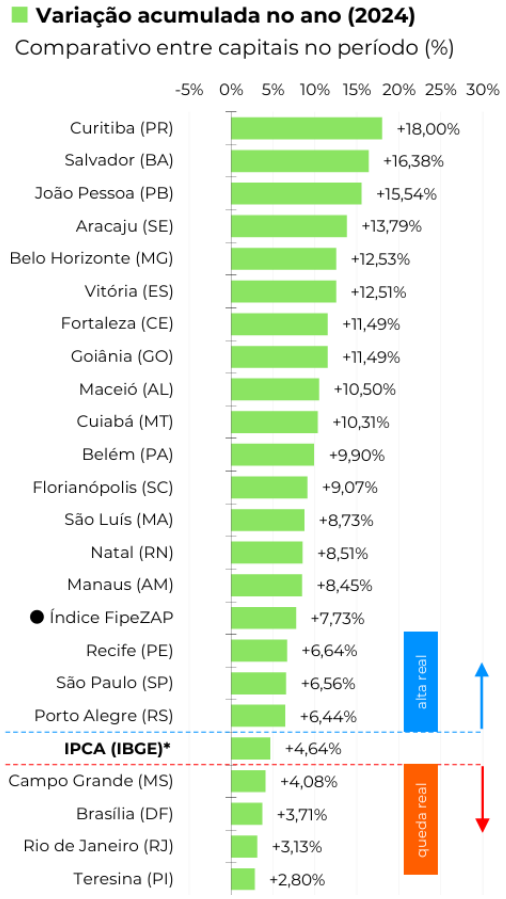
De acordo com o Índice FipeZAP, a capital paraibana registrou uma valorização acumulada de 15,54% no ano de 2024 (Figura 1), posicionando-se entre as três capitais brasileiras com maior crescimento nos preços residenciais, atrás apenas de Salvador e Curitiba (Figura 2). Esse aumento é ainda mais expressivo quando comparado ao índice médio nacional (+7,73%) e à inflação oficial medida pelo IPCA (+4,64%), evidenciando um cenário de alta real no preço dos imóveis (FipeZAP, 2024).

Figura 1: Resultados de João Pessoa do índice FipeZAP em 2024.



Fonte: Fundação Instituto de Pesquisas Econômicas, 2024.

Figura 2: Variação acumulada do preço de venda residencial entre capitais no ano de 2024.

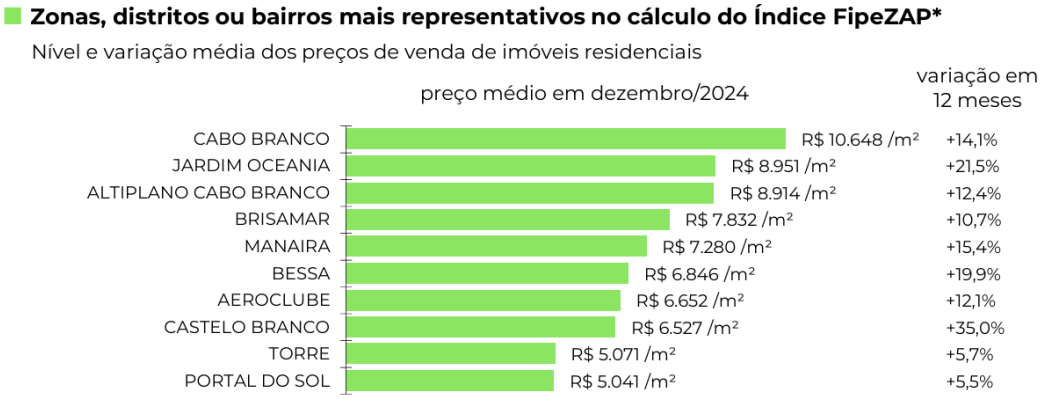


Fonte: Fundação Instituto de Pesquisas Econômicas, 2024.

Além disso, o preço médio do metro quadrado em João Pessoa atingiu R$ 6.890,00 em dezembro de 2024 (Figura 1), com bairros como Cabo Branco e Jardim Oceania ultrapassando os R$ 10 mil/m² (Figura 3). Esses dados revelam não apenas o crescimento do mercado, mas também uma alta heterogeneidade espacial nos valores, o que torna a avaliação ainda mais desafiadora. Em paralelo, a construção

civil nacional cresceu 4,3% no mesmo período (CBIC, 2025), refletindo um ambiente de constante transformação e expansão urbana.

Figura 3: Preço médio dos preços de vendas de imóveis residenciais em dezembro de 2024.



Fonte: Fundação Instituto de Pesquisas Econômicas, 2024.

Apesar da existência de normativas como a NBR 14653-2 (ABNT, 2011), que buscam padronizar critérios técnicos na avaliação de imóveis urbanos, sua aplicação prática encontra barreiras, sobretudo devido à subjetividade inerente a algumas variáveis. A percepção de segurança, a qualidade de vida e a facilidade de acesso a serviços, por exemplo, são aspectos difíceis de mensurar de forma objetiva, o que pode comprometer a consistência das estimativas de valor e impactar a confiança nas negociações.

Imóveis bem precificados são fundamentais para o funcionamento saudável do mercado. Uma avaliação justa e precisa promove a transparência, atrai investimentos e favorece o planejamento urbano eficiente. Em contrapartida, a precificação incorreta pode acarretar problemas como a ociosidade de unidades, perda de capital, distorções de mercado e desequilíbrios entre oferta e demanda.

Diante dessa realidade, torna-se operacionalmente relevante adotar tecnologias que contribuam para avaliações mais confiáveis, padronizadas e ajustadas à realidade do mercado. O Aprendizado de Máquina apresenta-se como uma ferramenta promissora nesse cenário, permitindo a análise de grandes volumes de dados e a detecção de padrões complexos, muitas vezes imperceptíveis pelos métodos tradicionais. Modelos baseados em Aprendizado de Máquina são capazes de lidar com múltiplas variáveis, reduzir a subjetividade e oferecer previsões mais precisas de preços imobiliários.

Dessa forma, este trabalho justifica-se pela necessidade de aplicação de soluções tecnológicas inovadoras ao contexto imobiliário de João Pessoa, especialmente diante do crescimento acentuado dos preços e da diversidade de fatores que influenciam o valor dos imóveis. Ao aplicar e comparar diferentes algoritmos de Aprendizado de Máquina na previsão de preços de venda de flats, esta pesquisa busca oferecer uma abordagem mais precisa e eficiente para a precificação, apoiando decisões de compra, venda e investimento com base em dados concretos e análises técnicas.

1.4 Estrutura do Trabalho

O restante deste Trabalho de Conclusão de Curso está organizado como segue: o Capítulo 2 aborda a fundamentação teórica, apresentando os principais conceitos relacionados à avaliação de imóveis, aprendizado de máquina, modelos de aprendizado supervisionado e métricas para análise de desempenho dos algoritmos. No Capítulo 3, é apresentada a metodologia adotada, incluindo os procedimentos de coleta, descrição e processamento dos dados, bem como a implementação e parametrização dos algoritmos utilizados. O Capítulo 4 aborda a análise e discussão dos resultados obtidos, com a comparação do desempenho dos modelos preditivos. Por fim, o Capítulo 5 apresenta as considerações finais, destacando as contribuições e limitações do trabalho e sugestões para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Mercado Imobiliário

O mercado imobiliário é o local onde as transações com imóveis são realizadas. Elas podem envolver a compra e a venda de imóveis, incorporações, locações, administração de condomínios, administração de shoppings, entre outras (Kremer, 2008). De acordo com Matos e Bartkiw (2013), o mercado imobiliário é formado por diversos agentes, incluindo imobiliárias, corretores autônomos, proprietários, empreiteiras, empresas da construção civil e prestadoras de serviços em propaganda e marketing. Esses atores desempenham papéis fundamentais na administração e comercialização de empreendimentos imobiliários, contribuindo para o funcionamento e desenvolvimento do setor.

Segundo os dados divulgados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o setor da construção civil registrou um crescimento de 4,3% em 2024, com um Produto Interno Bruto (PIB) de R$ 359,523 bilhões. Esse crescimento refletiu na geração de empregos formais, com a criação de 110.133 novas vagas, elevando o total de trabalhadores no setor para 2,858 milhões. O mercado imobiliário também registrou avanços expressivos, com um aumento de 20,9% nas vendas de apartamentos novos e um crescimento de 18,6% nos lançamentos (CBIC, 2025). Esse cenário é reflexo de uma maior disponibilidade de linhas de crédito para construção e do aumento de interesse em investir no mercado imobiliário.

Além de sua contribuição direta para o PIB e a geração de empregos, o mercado imobiliário influencia outros setores econômicos, como o financeiro, varejista e de serviços, devido à sua ampla cadeia produtiva. Investimentos em infraestrutura e desenvolvimento urbano, estimulados pelo mercado imobiliário, também promovem melhorias na qualidade de vida da população e no crescimento sustentável das cidades.

2.1.1 Avaliação de Bem Imóvel

Segundo a NBR 14653-1, avaliação de bens é definida como uma “análise técnica, realizada por engenheiro de avaliações, para identificar o valor de um bem, de seus custos, frutos e direitos, assim como determinar indicadores da viabilidade de sua utilização econômica, para uma determinada finalidade, situação e data” (Associação Brasileira de Normas Técnicas, 2001).

Ainda de acordo com a NBR 14653-2, a avaliação imobiliária deve seguir critérios técnicos e metodológicos para garantir precisão e confiabilidade nos resultados. Para isso, podem ser utilizados métodos de avaliação imobiliária, sendo os principais o método comparativo de mercado, o método de renda, o método involutivo e o evolutivo, todos estes citados pela norma técnica.

A avaliação imobiliária desempenha um papel essencial na economia e no planejamento urbano, sendo fundamental para a precificação justa dos imóveis e para a redução da subjetividade nas transações imobiliárias. A avaliação de bens imóveis não apenas garante segurança e transparência para o mercado imobiliário, mas também contribui para o equilíbrio econômico e social das cidades, promovendo negociações mais justas e investimentos mais seguros.

2.1.2 Métodos Tradicionais de Avaliação

A NBR 14653-2 cita métodos para identificar o valor de um bem, de seus frutos e direitos, sendo o primeiro deles o Método Comparativo Direto de Dados de Mercado (MCDDM), onde “identifica o valor de mercado do bem por meio de tratamento técnico dos atributos dos elementos comparáveis, constituintes da amostra” (NBR 14653-1, 2001). Para a identificação do valor são levadas em consideração variáveis relevantes para explicar a tendência da formação de valor e possíveis relações com a variável dependente.

De acordo com a NBR 14653-2, o método involutivo considera a valorização do imóvel a partir do estudo de um projeto hipotético de desenvolvimento imobiliário, levando em conta os custos de construção, despesas administrativas, impostos, lucro esperado pelo empreendedor e o valor final de comercialização das unidades produzidas (Associação Brasileira de Normas Técnicas, 2011). O valor decorrente de sua aplicação é um reflexo direto da capacidade de utilização do imóvel, pois neste método o profissional avaliador procura determinar seu valor por meio de estudo das condições eficientes de aproveitamento do terreno (Patrimônio da União, 2024).

O método involutivo “se baseia no custo de reprodução do bem, ou seja, no valor necessário para reproduzir a edificação em condições similares às do imóvel avaliado, e é utilizado quando os dados amostrais semelhantes ao objeto avaliando é escasso (Patrimônio da União, 2024). Já o método da renda, “é aplicável aos imóveis suscetíveis de produzir renda, por meio de aluguéis ou arrendamentos, da produção

ou de atividade de negócio, uma vez que se baseia na estimativa do valor presente dos fluxos de renda futuros gerados pelo bem" (Patrimônio da União, 2024).

2.2 Aprendizado de Máquina

A precificação de bens imóveis é um desafio complexo que envolve a análise de múltiplos fatores, como localização, características do imóvel, demanda do mercado e condições econômicas. O uso de ferramentas estatísticas é essencial para garantir uma avaliação precisa e objetiva, reduzindo a subjetividade e minimizando erros na determinação do valor de um imóvel.

Segundo Alpaydin (2014), o aprendizado de máquina consiste em programar computadores para otimizar um critério de desempenho com base em dados de exemplo ou experiências passadas. Nesse processo, um modelo é definido com certos parâmetros, e o aprendizado ocorre quando um programa ajusta esses parâmetros utilizando os dados de treinamento ou informações anteriores. O modelo resultante pode ter um caráter preditivo, permitindo realizar previsões futuras, descritivo, auxiliando na extração de conhecimento a partir dos dados, ou até mesmo ambas as funções simultaneamente.

Segundo Shalev-Shwartz e Ben-David (2014), o aprendizado de máquina recebe como entrada um conjunto de dados de treinamento, que representa a experiência adquirida, e gera como saída um modelo computacional capaz de desempenhar uma determinada tarefa, incorporando assim o conhecimento extraído dos dados. Essa ferramenta é amplamente aplicada em áreas científicas, como bioinformática, medicina e astronomia.

"O aprendizado de máquina pode ser dividido em três tipos principais: o supervisionado, o não supervisionado e o aprendizado por reforço. Cada tipo de aprendizado possui técnicas e abordagens específicas, para diferentes problemas" (Lauth, 2023).

2.2.1 Aprendizado Supervisionado

"No aprendizado supervisionado, o objetivo é prever o valor de uma variável resposta (*outcome*) a partir de variáveis preditoras (*inputs*)" (Morettin; Singer, 2019). Segundo James et al. (2013, p. 26), "Muitos métodos clássicos de aprendizado estatístico, como regressão linear e regressão logística, assim como abordagens mais

modernas, como Modelos Aditivos Generalizados (GAM), B*oosting* e Máquinas de Vetores de Suporte (SVM), operam no domínio do aprendizado supervisionado".

A variável resposta, que é o alvo desejado, pode ser uma variável quantitativa ou qualitativa. Quando se trata de variáveis quantitativas, um dos modelos mais utilizados é o de regressão. No contexto de previsão de preços, pode ser empregado para prever o preço de ativos financeiros, imóveis ou produtos com base em características históricas.

No caso de variáveis qualitativas (categóricas ou discretas), com um conjunto de valores finitos, a classificação é o modelo mais utilizado. Ela pode ser aplicada na previsão de preços de imóveis de forma indireta, utilizando uma abordagem baseada em categorias de preço em vez de prever um valor exato. Em vez de prever o preço específico de um imóvel, um modelo de classificação pode atribuir o imóvel a uma faixa de preço ou categoria (por exemplo, "baixo", "médio", "alto"), com base em características como localização, tamanho, número de quartos, entre outras.

2.2.2 Aprendizado Não-Supervisionado

"No caso de aprendizado não-supervisionado, temos apenas um conjunto de variáveis preditoras (*inputs*) e o objetivo é descrever associações e padrões entre essas variáveis. Nesse caso, não há uma variável resposta" (Morettin; Singer, 2019, p. 4). No aprendizado não-supervisionado o objetivo não é prever valores de saída.

De acordo com James et al. (2013, p. 373), existem "dois tipos particulares de aprendizado não supervisionado: a análise de componentes principais (PCA), uma ferramenta utilizada para visualização de dados ou pré-processamento dos dados antes da aplicação de técnicas supervisionadas, e a classificação (*clustering*), uma ampla classe de métodos para descobrir subgrupos desconhecidos nos dados".

No caso da previsão de preços de imóveis, o aprendizado não-supervisionado pode ser utilizado a classificação para agrupar os imóveis em *clusters* com base nas suas características similares como localização, número de quartos etc. E a análise de componentes principais (PCA), pode ser utilizada para reduzir a dimensionalidade dos dados antes de aplicar em outros modelos. Por exemplo, em um conjunto de dados de imóveis com muitas variáveis, é possível utilizar a PCA para identificar as principais variáveis que influenciam no preço dos imóveis.

2.2.3 Aprendizado por Reforço

“Existe também um cenário intermediário de aprendizado em que, embora os exemplos de treinamento contenham mais informações do que os exemplos de teste, o aprendiz é exigido a prever ainda mais informações para os exemplos de teste” (Shalev-Shwartz; Ben-David, 2014, p. 23). Segundo Alpaydin (2014), o modelo de aprendizado por reforço envolve a avaliação e aprimoramento contínuo das ações com base em recompensas ou *feedback*.

Esse tipo de aprendizado é amplamente utilizado em aplicações que exigem tomada de decisão em ambientes dinâmicos. Na previsão de preços, o aprendizado por reforço pode ser utilizado para estratégias de negociação automatizada (*trading*), onde um agente aprende a comprar e vender ativos de forma otimizada com base no comportamento do mercado.

2.3 Aprendizado Supervisionado

2.3.1 Regressão Linear

“A regressão linear é uma ferramenta estatística comum para modelar a relação entre algumas variáveis "explicativas" e um resultado com valor real” (Shalev-Shwartz e Ben-David, 2014, p. 123). O objetivo da regressão linear é descrever o comportamento de uma variável dependente (Y) em relação às variáveis independentes (X). O modelo de regressão pode ser dividido em linear simples e linear múltiplo. O modelo simples possui apenas duas variáveis, sendo uma dependente (Y), e uma independente (X). Já no caso do modelo de regressão linear múltiplo, há duas ou mais variáveis independentes (X_p).

A equação de uma regressão linear múltipla com variáveis independentes $X_1, X_2, X_3, \dots, X_p$ e uma variável dependente Y tem a forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

onde β_0 é o intercepto (valor de Y quando $\mathrm{X} = 0$), $\beta_1, \beta_2, \dots, \beta_p$ são os coeficientes de regressão em que cada coeficiente representa a variação esperada em Y para uma unidade de aumento em X_p e ε é o termo de erro aleatório (diferença entre o valor real e o valor previsto).

Segundo Montgomery, Peck e Vining (2012), modelos de regressão linear múltipla são frequentemente utilizados como modelos empíricos ou funções de

aproximação. A verdadeira relação entre as variáveis pode ser desconhecida, mas através do modelo de regressão linear é possível apresentar uma aproximação adequada da função real, ainda que desconhecida.

2.3.2 Regressão por Vetores de Suporte (SVR)

O modelo Máquina de Vetores de Suporte (SVM) é um algoritmo de aprendizado de máquina utilizado para classificação e para regressão, identificado pela primeira vez por Vladimir Vapnik (1992). A regressão por vetores de suporte (SVR) é uma adaptação do modelo SVM para resolver problemas de regressão. Enquanto o modelo SVM prevê valores discretos, o modelo SVR prevê valores contínuos.

Conforme a biblioteca do *Scikit-learn* (Pedregosa et al.), o modelo gerado através da regressão por vetores de suporte depende apenas de um subconjunto da base de dados de treino, pois a função ignora as amostras cuja previsão está próxima ao alvo. Segundo Lin e Chen (2011), o modelo SVR também pode lidar com dados não lineares e possui apenas uma única solução ótima para cada conjunto de parâmetros de *kernel* e parâmetro de margem flexível (*soft margin*).

Diferente da regressão linear tradicional, o SVR pode ser aplicado em situações em que a relação entre as variáveis não forma uma linha reta. Através da função kernel, é possível ajustar curvas e gerar uma única solução ótima.

2.3.3 Árvore de Decisão

Os modelos de árvore de decisão são de aprendizado supervisionado não paramétrico, utilizado tanto para classificação como para regressão. O seu objetivo é criar um modelo capaz de prever o valor de uma variável alvo através da aprendizagem de regras de decisão simples, que são obtidas a partir do banco de dados. “Os métodos baseados em árvores são simples e úteis para interpretação. No entanto, geralmente não são competitivos em relação aos melhores métodos de aprendizado supervisionado” (James et al., 2013, p. 303).

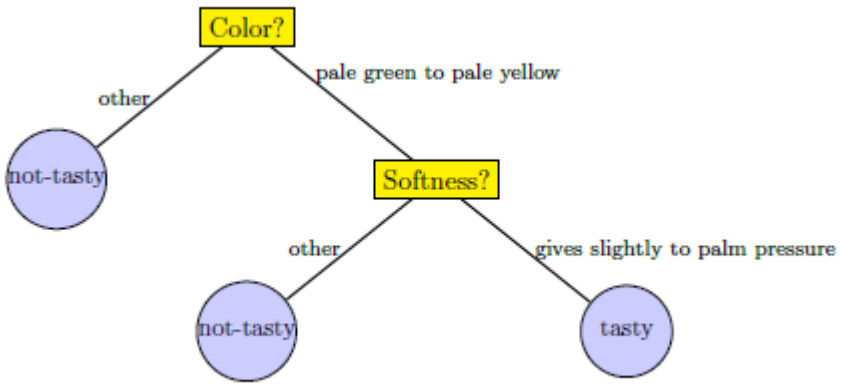
De acordo com Izbicki e Santos (2020, p. 77), uma árvore de decisão é construída por particionamentos recursivos no espaço das covariáveis. Ou seja, o conjunto de dados é dividido em grupos menores, esse processo é repetido várias vezes, cada vez dentro de uma subdivisão anterior. Cada particionamento é chamado de nó e cada resultado final recebe o nome de folha.

Segundo James et al. (2013, p. 306), o processo de construção da árvore de regressão se dá basicamente em duas etapas:

1. Divisão do espaço preditor, ou seja, divisão do conjunto de dados com base nas variáveis preditoras, tais como $X_1, X_2, X_3, \ldots, X_p$. O objetivo é segmentar o espaço dos dados em J regiões distintas, $R_1, R_2, R_3, \ldots, R_J$.
2. Previsão com a média das observações em cada região R_J, para cada nova observação é verificada a previsão, que é a média dos valores de resposta Y.

Para Alpaydin (2014, p. 213), uma árvore de decisão é um modelo hierárquico onde o processo de previsão ocorre por meio de uma sequência de divisões sucessivas (particionamentos recursivos) no espaço das variáveis independentes. Essas divisões são feitas de forma a reduzir gradualmente o conjunto de dados até alcançar uma região mais específica e precisa.

Figura 4:Exemplo de estrutura de uma árvore de regressão.



Fonte: Shalev-Shwartz e Ben-David, 2014.

De acordo com Shalev-Shwartz e Ben-David (2014, p. 252), o crescimento de uma árvore de decisão segue um processo iterativo, cujo objetivo é melhorar a capacidade preditiva do modelo com base na estrutura dos dados de treinamento. Esse processo pode ser dividido em três etapas: inicialização com a raiz, iterações com tentativas de divisão e realização ou não da divisão.

2.3.4 *Random Forest* (Floresta Aleatória)

O modelo *Random Forest*, ou Floresta Aleatória, consiste na criação de várias árvores de decisão para realizar predições, proposto por Leo Breiman em 2001. Segundo Breiman (2001), *Random Forest* é um classificador composto por uma

coleção de classificadores estruturados em forma de árvore $\{h(x,k), k = 1, \ldots\}$, onde os vetores $\{k\}$ são variáveis aleatórias independentes e identicamente distribuídas. Cada árvore realiza um voto unitário para a classe mais frequente correspondente à entrada x.

"O Random Forest é uma extensão da Árvore de Decisão, que cria muitas árvores de decisão de maneira aleatória, formando o que podemos enxergar como uma floresta e suas previsões são combinadas para produzir uma previsão final" (Lauth, 2023, p. 23). O modelo *Random Forest* traz uma melhoria em relação ao modelo *Bagging*, que é outra extensão de Árvores de Decisão, através da aleatoriedade na escolha de atributos das árvores.

Durante a construção de cada árvore, a cada nó, o algoritmo seleciona aleatoriamente um subconjunto de m preditores dentre os k preditores totais. Esse processo de amostragem de m variáveis é repetido em cada nó da árvore, onde geralmente $m \approx \sqrt{k}$, ou seja, o número de variáveis selecionadas em cada divisão da árvore é aproximadamente igual à raiz quadrada do número total de variáveis.

A predição do modelo *Random Forest* é feita com base em uma votação majoritária entre as árvores. Para a classificação, a classe escolhida pela maioria das árvores é a predição final, já para a regressão, é calculada a média das previsões individuais. "Os resultados do modelo tendem a ser pouco sensíveis ao número de variáveis escolhidas para cada nó. Normalmente, um ou duas variáveis já alcançam um resultado ótimo" (Breiman, 2001, p. 6).

2.3.5 XGBoost

Segundo Sicsú, Samartini e Barth (2023, p. 274), o *XGBoost* (*Extreme Gradient Boosting*) é uma extensão do modelo *Gradient Boosting*, um dos mais poderosos algoritmos de Aprendizado de Máquina atualmente devido sua acurácia, versatilidade na aplicação de problemas e rapidez de processamento. Chen e Guestrin (2016, p. 785) mostram a popularidade e a força do modelo *XGBoost* na ciência de dados ao citar que, entre as 29 soluções vencedoras de um desafio de Aprendizado de Máquina promovido pela *Kaggle* em 2015, 17 utilizaram o *XGBoost*. E, dentre as 17 soluções, 8 utilizaram exclusivamente o XGBoost para treinar o modelo.

"Os resultados obtidos pelo modelo *XGBoost* mostram a sua eficiência para a resolução de diversos tipos de problemas, desde previsão de vendas até classificação

de *softwares* maliciosos” (Chen e Guestrin, 2016, p. 785). O principal fator do sucesso do algoritmo é a sua escalabilidade em todos os cenários. “Em vez de treinar o melhor modelo possível nos dados, treina-se milhares de modelos em vários subconjuntos do conjunto de dados de treinamento e vota-se no modelo de melhor desempenho” (Neptune AI, 2023).

O treinamento do modelo se dá através da busca pelos melhores parâmetros, ou seja, os coeficientes θ, que melhor se ajustam aos dados de treino X_i e valores alvo Y_i .Para treinar o modelo, é necessário definir uma função objetivo, que vai medir o desempenho do modelo em relação aos dados de treinamento. A função objetivo é definida na seguinte forma:

$$obj(\theta) = L(\theta) + R(\theta)$$

onde L é a função de perda específica para cada tipo de tarefa, como o erro quadrático médio, e R é o termo de regularização, que penaliza modelos muito complexos, evitando o *overfiting* (XGBoost Developers, 2025).

“O *XGBoost* é capaz de resolver problemas em escala real usando uma quantidade mínima de recursos” (Chen e Guestrin, 2016, p. 794). A sua alta precisão, velocidade e capacidade de lidar com grandes volumes de dados o torna uma das ferramentas mais importantes da ciência de dados, sendo assim, referência em aplicações práticas e acadêmicas.

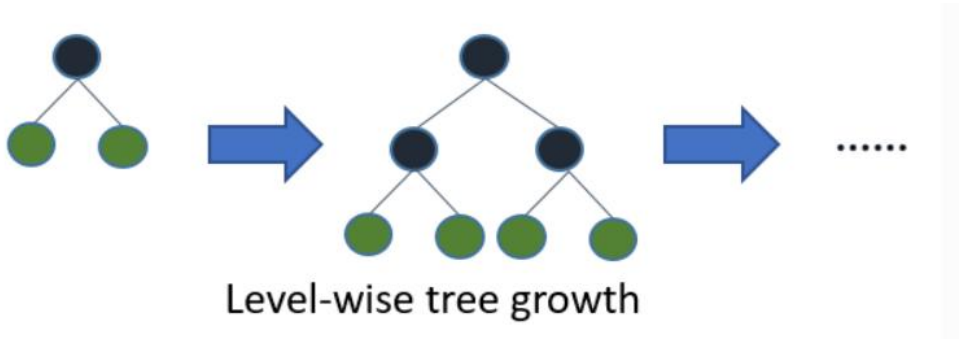
2.3.6 *Light* GBM

O modelo *Light* GBM (*Light Gradient Boosting Machine*), assim como o *XGBoost*, é baseado no *Gradient Boosting*. “É um modelo de *Gradient Boosting* que utiliza algoritmos de aprendizado baseados em árvores” (*Microsoft*, 2025). O *Light* GBM foi desenvolvido pela equipe de pesquisa da *Microsoft* devido à uma necessidade de algoritmos de *Gradient Boosting* mais rápidos e escaláveis para grandes volumes de dados e ambientes distribuídos.

“Light GBM tem como prefixo *Light* devido a sua velocidade de processamento. É possível lidar com grandes volumes de dados e ocupar menos memória para ser executado” (Banarjee, 2025). Segundo a Microsoft, o Light GBM utiliza algoritmos baseados em histogramas que transformam os valores contínuos dos atributos (*features*) em intervalos discretos (*bins*).

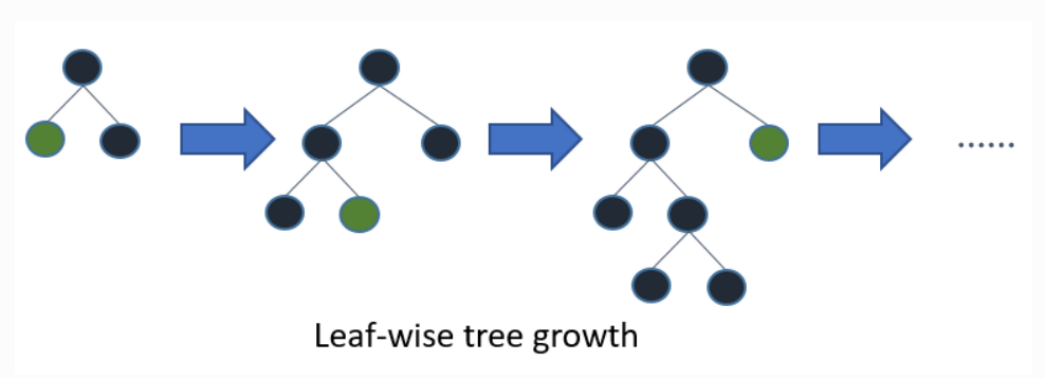
O principal diferencial do modelo está na forma como ele constrói as árvores, em vez do crescimento tradicional nível a nível (*level-wise*), ele utiliza uma abordagem chamada crescimento por folhas (*leaf-wise growth*). Nesse caso, o algoritmo escolhe a folha com maior redução de perda para crescer. Quando o número de folhas é mantido fixo, os algoritmos *leaf-wise* tendem a alcançar uma perda menor do que os algoritmos que crescem nível a nível (*level-wise*).

Figura 5: Exemplo do crescimento tradicional nível a nível



Fonte: LightGBM Documentation, Microsoft, 2025.

Figura 6: Exemplo do crescimento por folha.



Fonte: LightGBM Documentation, Microsoft, 2025.

O *Light* GBM é amplamente adotado por profissionais de ciência de dados por sua eficiência computacional aliada a uma ótima capacidade preditiva, sendo uma escolha de destaque em competições e soluções de Aprendizado de Máquina. O modelo surgiu como uma solução moderna e otimizada desenvolvida pela Microsoft, com o objetivo de melhorar a performance do *Gradient Boosting* clássico, especialmente em problemas de larga escala, sem comprometer a qualidade das previsões.

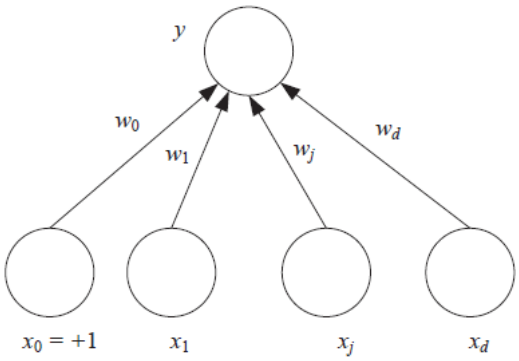
2.3.7 Redes Neurais Artificiais

“As Redes Neurais Artificias são modelos computacionais inspirados no funcionamento do cérebro” (Shalev-Shwartz e Ben-David, 2014). “Neurologistas, juntamente com pesquisadores de outras áreas, tais como eletrônica, automação, biofísica, matemática, desejavam produzir um modelo que descrevesse a rede neural biológica” (Medeiros, 1999, p. 71).

“O modelo de Redes Neurais Artificias foi introduzido em 1943 por Warren McCulloch e Walter Pitts, o modelo propõe elementos computacionais retirados das propriedades fisiológicas de um neurônio biológico e de suas conexões” (James, 2020). A engenharia está interessada em Redes Neurais Artificias pois acredita que o modelo pode ajudar a criar melhores sistemas computacionais.

Uma Rede Neural Artificial é formada por diversas unidades de processamento, que são os neurônios artificias. Cada unidade de processamento é interligada por canais de comunicação ou conexões sinápticas. “Uma rede neural artificial pode ser vista como um grafo onde os nós são os neurônios e as ligações fazem a função das sinapses” (Ferneda, 2006). Segundo Shalev-Shwartz e Ben-David (2014, p. 268), uma Rede Neural pode ser descrita como um grafo direcionado, cujos nós correspondem a neurônios e as arestas correspondem às conexões entre eles.

Figura 7: Exemplo do grafo direcionado de uma Rede Neural Artificial.

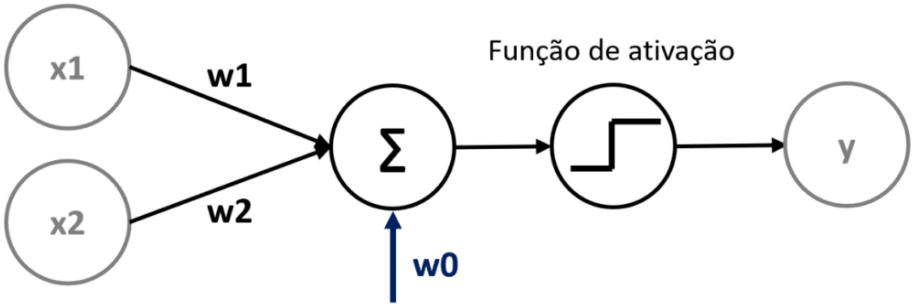


Fonte: Alpaydin, 2014.

Cada conexão possui um valor associado, denominado peso, que representa a importância daquela entrada específica na ativação do neurônio seguinte. Esses pesos são ajustáveis e fundamentais para o aprendizado da rede, pois determinam a influência de cada entrada no resultado final. O processamento ocorre por meio de

uma soma ponderada entre as entradas e seus respectivos pesos, seguida da aplicação de uma função de ativação, que define a saída do neurônio.

Figura 8: Exemplificação do perceptron.



Fonte: Araújo, 2023.

"O *perceptron* é o elemento básico do processamento de uma Rede Neural. Ele recebe entradas que podem ser provenientes do ambiente (variáveis de entrada) ou das saídas de outros *perceptrons*" (Alpaydin, 2014). Para cada entrada, $x_1, x_2, \dots, x_j$, está associado um peso sináptico, $w_1, w_2, \dots, w_j$, que representa a influência daquela entrada sobre a saída, onde $j = 1, \dots, d$. A saída y é dada pela soma ponderada das entradas com seus pesos, ou seja, um produto escalar entre o vetor de entradas e o vetor de pesos. A função tem o seguinte formato:

$$y = \sum_{j=1}^{d} w_j x_j + w_0$$

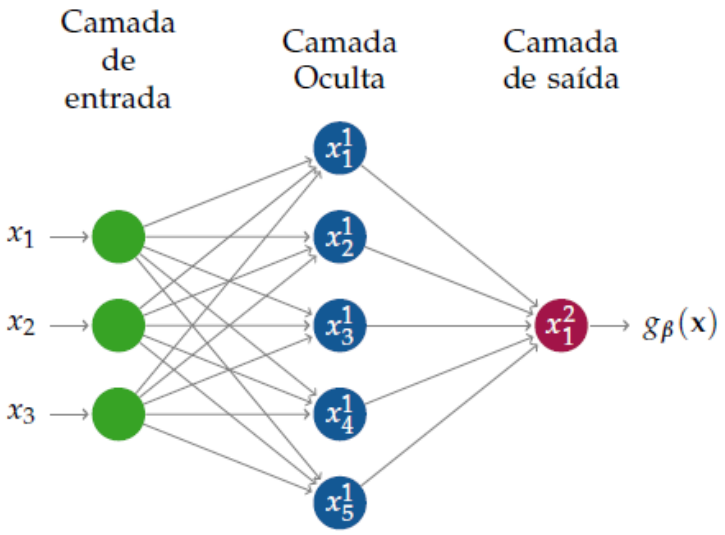
onde w_0, denominado como viés (*bias*), é um termo adicional que permite à Rede Neural ajustar-se melhor aos dados e aumentar sua flexibilidade. Ou seja, é uma contante adicionada à soma ponderada das entradas do neurônio.

Segundo Shalev-Shwartz e Ben-David (2014), é comum assumir que a estrutura da rede está organizada em camadas. Isso significa que os nós (neurônios) da rede podem ser agrupados em subconjuntos distintos e não vazios, formando diferentes níveis ou camadas. Uma Rede Neural geralmente é dividida em três principais tipos de camadas:

1. Camada de entrada (*input layer*): é a primeira camada, entrada de dados brutos;
2. Camadas ocultas (*hideen layers*): fica entre a camada de entrada e a camada de saída, responsáveis pelo processamento e extração de padrões dos dados;

3. Cada de saída (*output layer*): é a última camada, que gera a previsão para classificação ou regressão.

Figura 9: Exemplo de rede neural com uma camada oculta.



Fonte: Izbicki e Santos, 2020.

"De forma geral, uma rede neural pode ter várias camadas ocultas e um número diferente de neurônios em cada camada. Essas são escolhas feitas pelo usuário" (Izbicki e Santos, 2020). As Redes Neurais Artificias são divididas em diversos tipos, cada uma utilizada para um problema específico, sendo os principais: *Perceptron* Simples, *Perceptron* Multicamadas (MLP), Rede Neural Convolucional (CNN), Rede Neural Recorrente (RNN), Rede de Função de Base Radial (RBF), *Autoencoders*, Redes Neurais Generativas (GANs) e Transformers.

As redes neurais representam um dos pilares mais poderosos da inteligência artificial moderna, capazes de aprender padrões complexos a partir de grandes volumes de dados. Sua estrutura em camadas e variedade de arquiteturas permite aplicações em diversas áreas. Cada novo tipo de rede neural é um avanço para a ciência de dados, sendo possível solucionar problemas cada vez mais de formas inteligentes e inovadoras.

2.4 Métricas de Desempenho

"A avaliação da qualidade dos algoritmos de regressão geralmente utiliza valores de erros em relação ao que o modelo previu e aos dados da vida real. Esses erros são medidos usando-se métricas de desempenho" (Netto e Maciel, 2021, p.

380). Abaixo estão as principais métricas de desempenho utilizadas para quantificar o quão bem um algoritmo de aprendizado está executando uma tarefa.

2.4.1 Erro Médio Absoluto (MAE)

O Erro Médio Absoluto, ou *Mean Absolute Error* (MAE), é a diferença absoluta média entre os valores observados e os valores previstos pelo modelo. “Outra maneira de pensar sobre o desempenho do modelo é considerar quão distante, em média, sua previsão estava do valor real” (Lantz, 2015, p. 198). O Erro Médio Absoluto é dado por:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - f(x_i)|$$

2.4.2 Erro Quadrático Médio (MSE)

O Erro Quadrático Médio, também conhecido como MSE (*Mean Squared Error*), é a média ao quadrado da diferença entre os dados originais e os dados previstos. “Quando se trata de regressão, é a medida mais utilizada para quantificar até que ponto o valor de resposta previsto para uma determinada observação está próximo do valor real da resposta” (James et al., 2023). O Erro Quadrático Médio é dado por:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - f(x_i))^2$$

onde $f(x_i)$ é o valor previsto pelo modelo para a observação i, y_i é o valor real da observação e n é o número total de observações.

Quanto menor o MSE, mais próximo estão as respostas previstas das respostas reais, enquanto quanto maior o MSE, mais as respostas previstas se diferem das respostas reais. O uso do quadrado dos erros penaliza fortemente desvios maiores, o que pode ser útil se você quiser evitar grandes erros individuais.

2.4.3 Raiz do Erro Quadrático Médio (RMSE)

A Raiz do Erro Quadrático Médio, também conhecido como RMSE (*Root Mean Squared Error*), é uma métrica estatística padrão utilizada para medir o desempenho de modelos em diversas áreas. Segundo Willmott e Matsuura (2005, p. 80), o cálculo

do RMSE envolve três etapas: calcular o erro quadrático total, calcular o erro quadrático médio (MSE) e tirar a raiz quadrada. Esse cálculo é dado por:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2} = \sqrt{MSE}$$

onde $f(x_i)$ é o valor previsto pelo modelo para a observação i, y_i é o valor real da observação e n é o número total de observações.

Como a Raiz do Erro Quadrático Médio eleva os erros ao quadrado, erros maiores são mais penalizados, o que é interessante para evitar grandes desvios. Contudo, se a base de dados tiver *outliers* extremos, o RMSE pode não ser ideal. Assim como o MSE, quanto menor o seu valor, melhor o desempenho do modelo.

2.4.4 Coeficiente de Determinação Médio (R² médio)

O Coeficiente de Determinação, também chamado de estatística de R-quadrado ou R2, varia de 0 a 1 e mede a proporção de variância da variável dependente que é explicada pelas variáveis independentes no modelo. Quanto mais próximo de 1, melhor o modelo explica os dados (Scikit-learn, 2025). Em termos matemáticos, o R² é definido como:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

onde SS_{res} é a soma dos quadrados dos resíduos e SS_{tot} é a soma total dos quadrados.

Contudo, calcular o R² com base apenas nos dados de treinamento pode induzir um *overfitting*, que é quando o modelo se ajusta excessivamente aos dados de treino, mas tem baixo desempenho em novos dados. Nesse caso, é comum utilizar as técnicas de Validação Cruzada (*cross-validation*), que segundo Netto e Maciel (2021, p. 154), consiste em separar as partes de treino e teste em subconjuntos distintos que abranjam todo o conjunto de dados, o modelo é ajustado e avaliado várias vezes, obtendo-se então o Coeficiente de Determinação Médio (R² médio), chamado também de *mean cross-validated*.

Um dos métodos mais utilizados é a validação cruzada k-fold, que consiste em dividir os dados em k partes iguais. O modelo é treinado em k – 1 partes e testado na parte restante, repetindo esse processo k vezes. A média dos coeficientes de determinação obtidos em cada *fold* é chamada de Coeficiente de Determinação Médio (R² médio).

2.5 Trabalhos Correlatos

O uso de técnicas de Aprendizado de Máquina na previsão de preços ganhou destaque com o avanço das tecnologias de dados e a crescente necessidade de soluções analíticas para problemas complexos. Inicialmente aplicadas com sucesso em áreas como o setor financeiro, marketing e e-commerce (Alpaydin, 2014, p.3). Essas técnicas passaram a ser exploradas também no mercado imobiliário, com o objetivo de reduzir a subjetividade nas avaliações e aumentar a precisão na estimativa de valores.

A popularização de bases de dados públicas e privadas, aliada à maior capacidade computacional, possibilitou que algoritmos preditivos fossem utilizados para modelar a relação entre características dos imóveis e seus respectivos preços (JAMES et al., 2013, p.20). Com isso, o Aprendizado de Máquina tornou-se uma ferramenta poderosa para substituir ou complementar métodos tradicionais de avaliação imobiliária, como o comparativo direto de dados de mercado, especialmente em contextos urbanos com alta variabilidade de preços.

Com base nesse cenário, diversos estudos têm aplicado modelos de aprendizado de máquina à previsão de preços imobiliários, tanto no Brasil quanto no exterior. Na cidade de Aracaju–SE, Sousa (2023) aplicou o algoritmo *Random Forest* para estimar o valor de venda de imóveis. O banco de dados utilizado foi coletado via *web scrapping* em cinco *sites* de imobiliárias diferentes. O modelo apresentou um bom desempenho, sendo especialmente eficaz para imóveis com valores de até R$ 1.000.000,00. O estudo mostrou que o tamanho da área construída foi a variável mais influente na precificação.

Também no nordeste brasileiro, Silva (2019) analisou diversos modelos de Aprendizado de Máquina para a precificação de imóveis na cidade de Fortaleza–CE, tais como Regressão Linear, Regressão Gaussiana, *Random Forest*, Redes Neurais Artificiais e *Gradient Boosting Machine*. O trabalho concluiu que todos os modelos simulados são adequados para a previsão de preços, visto que todos possuem um erro abaixo de 15%, sendo os modelos de *Random Forest* e *Gradient Boosting Machine* os de melhor desempenho.

Lauth (2023) desenvolveu um modelo preditivo para o preço de aluguel de apartamentos em Blumenau–SC utilizando algoritmos como Regressão Linear, *Random Forest*, *XGBoost* e Redes Neurais. Os resultados indicaram que a Regressão Linear obteve o melhor desempenho preditivo, com erro absoluto médio inferior a 5%. Já em Goiânia–GO, Oliveira Filho (2023) comparou modelos como Regressão Linear, Ridge e Lasso para determinar o melhor algoritmo na precificação de imóveis. Os dados foram coletados através da plataforma *Kaggle* e foram observados 13031 dados. A sua pesquisa concluiu que o modelo Lasso obteve os menores erros médios na previsão dos preços, mantendo apenas duas variáveis no modelo, sendo elas o número de quartos e de banheiros.

Além da precificação de imóveis já edificados, Dias (2023) em seu estudo propôs o uso de modelos de precificação hedônicos aliados ao aprendizado de máquina para prever os preços de terrenos em Brasília–DF. Foram utilizados mais de 30 algoritmos de Aprendizado de Máquina, sendo a Árvore de Decisão Baseada em Histograma (*HistGradientBoostingRegressor*) o modelo com melhor resultado na previsão de preços dos terrenos. No trabalho de Potrich (2024), foi utilizada uma base de dados da plataforma Viva Real para prever os preços de imóveis em Florianópolis, aplicando os modelos Lasso *Regression*, *Random Forest* e *XGBoost*, em que o *XGBoost* foi o modelo com melhor desempenho.

Miranda, Zuviollo e Pugliesi (2023) propuseram a criação de um modelo preciso de previsão de preços de aluguel de imóveis nas cidades de São Paulo, Rio de Janeiro, Porto Alegre, Belo Horizonte e Campinas, coletando os dados na plataforma *Kaggle* com 10962 observações. Neste estudo, todos os algoritmos desenvolveram modelos confiáveis, sendo eles Regressão por Vetores de Suporte, Regressão Linear e *Random Forest*.

Por fim, no contexto internacional, Francisco (2024) analisou dados habitacionais de Singapura durante o período entre janeiro de 1990 e dezembro de 2023, obtendo 915.374 registros de transações de vendas de imóveis. Concluiu-se com seu estudo que as variáveis com maior impacto no preço das casas em Singapura são: o ano de transação da venda, o tamanho do imóvel, número de quartos e a cidade. Foram comparados os modelos Lasso e *Random Forest*, sendo o *Random Forest* o mais eficaz para a previsão dos preços.

Quadro 1: Resumo dos trabalhos citados.

Autor/Ano	Localidade	Tipo de Imóvel	Modelos Utilizados	Melhor Desempenho
Sousa (2023)	Aracaju/SE	Imóveis à venda	Random Forest	Random Forest
Silva (2019)	Fortaleza/CE	Imóveis residenciais	Regressão Linear, Regressão Gaussiana, Random Forest, Redes Neurais Artificiais e Gradient Boosting Machine	Random Forest e Gradient Boosting Machine
Lauth (2023)	Blumenau/SC	Apartamentos para aluguel	Regressão Linear, Random Forest, XGBoost e Redes Neurais	Regressão Linear
Oliveira Filho (2023)	Goiânia/GO	Imóveis residenciais	Regressão Linear, Ridge, Lasso	Regressão Lasso
Dias (2023)	Brasília/DF	Terrenos	+30 modelos	Árvore de Decisão Baseada em Histograma
Potrich (2024)	Florianópolis/SC	Imóveis à venda	Lasso Regression, Random Forest e XGBoost	XGBoost
Miranda, Zuviollo e Pugliesi (2023)	Cidades de São Paulo, Rio de Janeiro, Porto Alegre, Belo Horizonte e Campinas	Imóveis para aluguel	Regressão por Vetores de Suporte, Regressão Linear e Random Forest	Regressão Linear
Francisco (2024)	Singapura	Imóveis vendidos	Lasso Regression, Random Forest	Random Forest

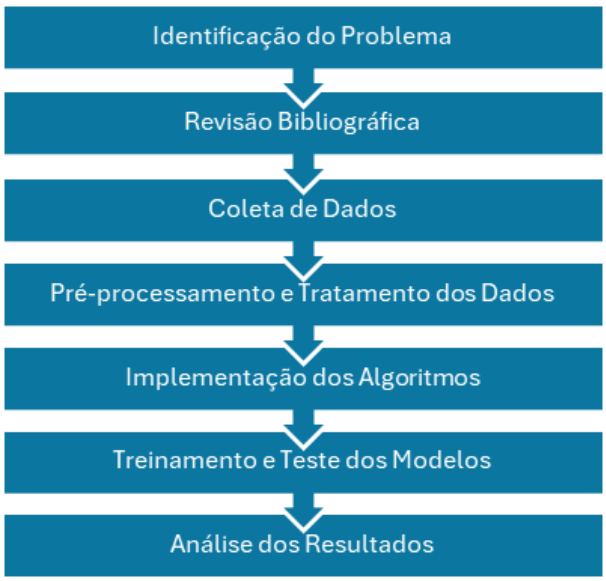
Fonte: Elaborado pelo autor (2025).

3 METODOLOGIA

A metodologia adotada neste trabalho visa o desenvolvimento e a avaliação de modelos preditivos para estimar os preços de venda de flats na cidade de João Pessoa–PB, utilizando algoritmos de aprendizado de máquina. Para isso, foram seguidas etapas sistemáticas que incluem: a coleta de dados de anúncios imobiliários, o pré-processamento e tratamento dos dados, a seleção e implementação de algoritmos de aprendizado supervisionado e, por fim, a avaliação do desempenho dos modelos preditivos com base em métricas estatísticas. Esse processo foi desenvolvido com o uso da linguagem Python e de bibliotecas especializadas em ciência de dados.

Este trabalho tem natureza aplicada, pois busca resolver um problema prático – a precificação de imóveis – por meio de métodos de Aprendizado de Máquina. A pesquisa possui abordagem quantitativa, baseada na análise de dados numéricos, através das métricas de desempenho. Na Figura 10 estão representadas as etapas de desenvolvimento deste trabalho.

Figura 10: Etapas de desenvolvimento do trabalho.



Fonte: Elaborado pelo autor (2025).

3.1 Coleta de Dados

O conjunto de dados coletados foi referente aos imóveis anunciados no site Viva Real entre os dias 28/01/2025 e 23/02/2025. Para as buscas, foram indicadas as opções “Comprar” e “*Flat*” (tipo de imóvel) na cidade de João Pessoa –PB (“Onde deseja morar?”). A busca retornou, em média, 1800 anúncios de *flats* em João Pessoa,

dos quais foram coletados 1000 dados. Dentre os 1000 dados coletados, não havia dados duplicados, porém havia anúncios que não informavam as informações necessárias para o preenchimento completo dos dados analisados.

Os dados foram coletados anúncio a anúncio, uma vez que as variáveis escolhidas para esta análise nem sempre eram informadas nas características do imóvel, tornando-se inviável a utilização de *web scrapping*. Para a obtenção dos dados coletados, foi necessário, além de analisar as características listadas pelo anunciante, ler a descrição do imóvel e visualizar as imagens para uma captação mais completa. Os dados obtidos em cada anúncio foram organizados em uma planilha no E*xcel*, separados pelo código do anúncio no *site*.

A Figura 11 representa o cabeçalho do conjunto de dados iniciais coletados.

Figura 11: Cabeçalho do conjunto de dados iniciais coletados.

CÓD	ÁREA	PISCINA	ESPAÇO GOURMET	LAVANDERIA	COWORKING	ACADEMIA	PLAYGROUND/ SALÃO DE JOGOS	SALÃO DE FESTAS	BAIRRO	SITUAÇÃO	VALOR
2506580979	20	S	S	S	S	S	S	S	CABO BRANCO	FINALIZADO	R$ 475.000,00
2774013145	23								TAMBAÚ	EM CONSTRUÇÃO	R$ 368.880,00
2773396414	27								JARDIM OCEANIA	FINALIZADO	R$ 300.000,00
2609616397	30	S	S	S	S	S	S	S	CABO BRANCO	EM CONSTRUÇÃO	R$ 520.000,00
2758140231	22	S	S	S	S	S	S	N	JARDIM OCEANIA	EM CONSTRUÇÃO	R$ 299.800,00
2773789713	47	S	N	N	S	N	N	S	MANAÍRA	FINALIZADO	R$ 280.000,00
2743765192	26	S	S	S	S	S	S	S	MANAÍRA	FINALIZADO	R$ 319.000,00
2722691241	40								JARDIM OCEANIA	FINALIZADO	R$ 590.000,00
2753794107	23	S	S	S	S	S	N	N	JARDIM OCEANIA	EM CONSTRUÇÃO	R$ 329.603,00
2755379810	24	S	S	S	N	S	N	N	JARDIM OCEANIA	FINALIZADO	R$ 369.000,00

Fonte: Elaborado pelo autor (2025).

3.2 Descrição dos Dados

Para dar início à etapa de modelagem preditiva, é necessário realizar o adequado processamento dos dados. Esta etapa é fundamental para garantir a qualidade das informações que serão utilizadas na construção dos modelos, bem como para assegurar que os resultados obtidos reflitam de maneira fidedigna a realidade do mercado imobiliário. O conjunto de dados utilizado neste trabalho refere-se a flats localizados na cidade de João Pessoa e inclui variáveis que representam tanto características físicas dos imóveis quanto aspectos relacionados à infraestrutura do empreendimento e à sua localização.

Essas informações foram selecionadas com base em sua relevância para a definição do valor de venda, considerando os critérios comumente observados por compradores em potencial. As construtoras de empreendimentos do tipo *flats* em João Pessoa buscam proporcionar uma experiência completa ao comprador, agregando ambientes de lazer e facilidades para o inquilino. Áreas como *coworking*, lavanderia,

academia, espaço *gourmet*, *pet place*, mercado e spa estão presentes nos imóveis da cidade de João Pessoa.

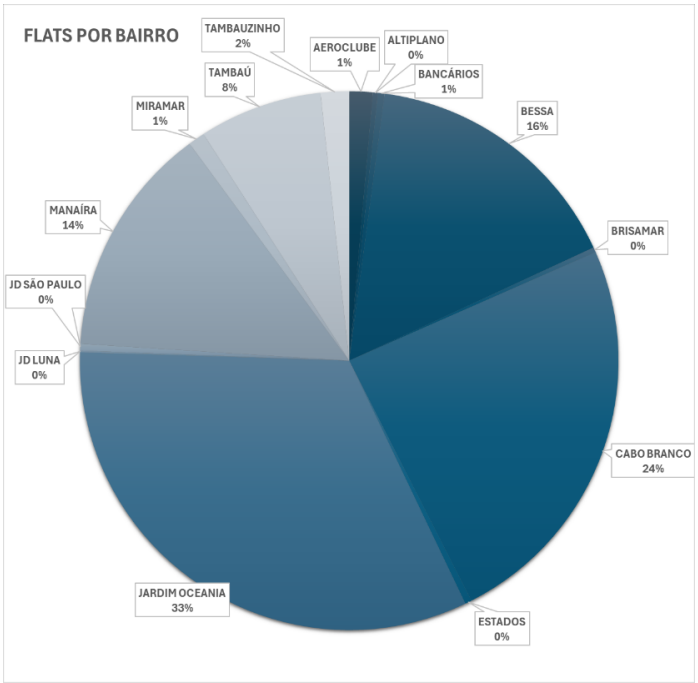
Tratando-se de empreendimento tipo *flat*, que sua estrutura é composta basicamente por uma área utilizada como quarto e cozinha, além de um banheiro, não faria sentido analisar a quantidade de quartos e banheiros do imóvel, variáveis estas comumente utilizadas na avaliação do bem imóvel. Assim, as variáveis selecionadas para a coleta de dados estão listadas no Quadro 2 abaixo.

Quadro 2: Variáveis do banco de dados coletado no site Viva Real.

Variável	Descrição
Código	Identificador do anúncio no *site*
Área	Área do imóvel em m²
Piscina	Possui ou não piscina no empreendimento
Espaço *Gourmet*	Possui ou não espaço *gourmet* no empreendimento
Lavanderia	Possui ou não lavanderia no empreendimento
Coworking	Possui ou não *coworking* no empreendimento
Academia	Possui ou não academia no empreendimento
Playground/ Salão de Jogos	Possui ou não *playground*/salão de jogos no empreendimento
Salão de Festas	Possui ou não salão de festas no empreendimento
Bairro	Bairro localizado o imóvel
Situação	Em construção ou obra entregue
Valor	Valor de venda do imóvel

Fonte: Elaborado pelo autor (2025).

Figura 12: Gráfico pizza dos flats por bairro em percentual.



Fonte: Elaborado pelo autor (2025).

A variável “código” foi utilizada apenas para identificar possíveis anúncios duplicados, não sendo utilizada nos modelos para a previsão de preços. Quanto aos bairros presentes na variável “bairro”, foram levantados 13 bairros da cidade de João Pessoa, sendo eles: Aeroclube, Altiplano, Bancários, Bessa, Brisamar, Cabo Branco, Bairro dos Estados, Jardim Oceania, Jardim Luna, Jardim São Paulo, Manaíra, Miramar, Tambaú e Tambauzinho. Na Figura 12 é possível observar a divisão dos flats por bairro em percentual.

A partir das variáveis selecionadas, dividiu-se as mesmas em variáveis categóricas e variáveis numéricas, conforme o Quadro 3 abaixo:

Quadro 3: Classificação das variáveis quanto ao tipo.

Variável	Tipo de Variável
Área	Contínua
Piscina	Categórica binária
Espaço *Gourmet*	Categórica binária
Lavanderia	Categórica binária
Coworking	Categórica binária
Academia	Categórica binária
Playground/Salão de Jogos	Categórica binária
Salão de Festas	Categórica binária
Bairro	Categórica de 1 a 13
Situação	Categórica binária
Valor	Contínua

Fonte: Elaborado pelo autor (2025).

Definidas as variáveis que serão consideradas nos modelos de previsão de preços, foram removidos os dados incompletos. Dos 1000 dados coletados a partir dos anúncios do *site* Viva Real, 63 não possuíam as informações completas, faltando os dados referentes a possuir ou não área *gourmet*, lavanderia, coworking, academia, *playground*/salão de jogos e salão de festas. Após a remoção dos dados faltantes do banco de dados, foi possível realizar a etapa de processamento destes.

3.3 Processamento dos Dados

Este trabalho foi realizado no *Google Colab* que é uma plataforma que funciona na nuvem e hospeda *notebooks* do *Jupyter.* Os dados foram estruturados em um *dataframe* por meio da biblioteca *Pandas*. Na Figura 13 apresenta-se o *dataframe* inicial, com os dados no formato que foram coletados.

Figura 13: Dataframe inicial.

	ÁREA	PISCINA	GOURMET	LAVANDERIA	COWORKING	ACADEMIA	PLAYGROUND	SALAO FESTAS	BAIRRO	SITUAÇÃO	VALOR
0	20	SIM	SIM	SIM	SIM	SIM	SIM	SIM	CABO BRANCO	FINALIZADO	475000
1	23	NaN	NaN	NaN	NaN	NaN	NaN	NaN	TAMBAÚ	EM CONSTRUÇÃO	368880
2	27	NaN	NaN	NaN	NaN	NaN	NaN	NaN	JARDIM OCEANIA	FINALIZADO	300000
3	30	SIM	SIM	SIM	SIM	SIM	SIM	SIM	CABO BRANCO	EM CONSTRUÇÃO	520000
4	22	SIM	SIM	SIM	SIM	SIM	SIM	NÃO	JARDIM OCEANIA	EM CONSTRUÇÃO	299800

Fonte: Elaborado pelo autor (2025).

3.3.1 Análise Descritiva

Após a remoção dos dados incompletos, 937 anúncios estavam disponíveis para análise. Dentre as variáveis escolhidas para processamento, apenas 2 eram variáveis contínuas, sendo elas: a área do imóvel em m² e o valor do imóvel, que era o alvo. Para compreender melhor estas variáveis, foi utilizada a função *describe* para obter dados como contagem (*count*), média (*mean*), desvio padrão (*std*), valor mínimo (*min*), primeiro, segundo e terceiro quartil (25%, 50% e 75%), e o valor máximo (*max*).

Figura 14: Análise descritiva das variáveis área e valor.

	ÁREA	VALOR
count	937.000000	937.000000
mean	25.693703	370062.309498
std	7.418870	101032.927350
min	12.000000	149990.000000
25%	20.000000	299900.000000
50%	24.000000	352000.000000
75%	30.000000	410000.000000
max	55.000000	865250.000000

Fonte: Elaborado pelo autor (2025).

A análise dos dados processados mostrou que o tamanho médio de um flat é de 26 m² e o valor médio é de R$ 370.062,00. O valor máximo foi de R$ 865.250,00 e o valor mínimo foi de R$ 149.990,00. Foi possível verificar ainda um desvio padrão na área de 7,4 m² e de R$ 101.032,00 no valor.

A análise descritiva dos dados é importante para verificar discrepâncias (*outliers*) e tomar decisões acerca dos dados que serão utilizados no algoritmo de previsão de preços. Para este trabalho, não foram excluídos valores extremos, pois

entendeu-se que era importante estes valores estarem presentes na análise, dado que o valor e a área são as principais variáveis para o estudo.

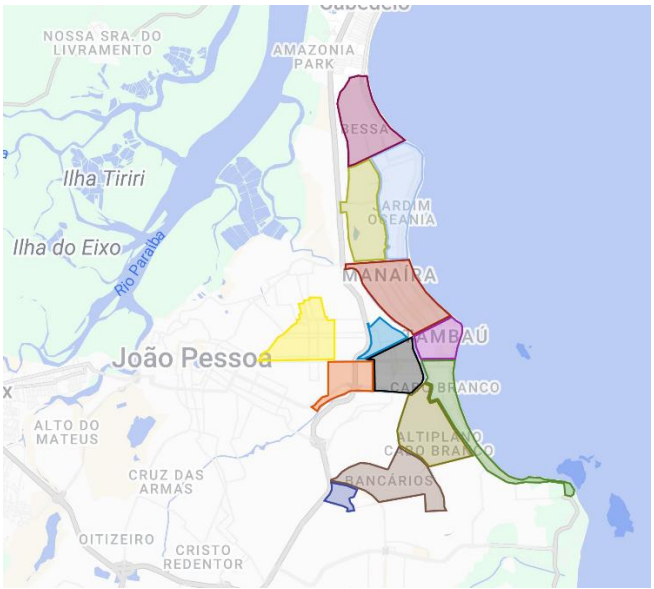
Quanto aos bairros, verificou-se que 86,6% dos *flats* estão localizados em Manaíra, Bessa, Cabo Branco e Jardim Oceania. Na Tabela 1 é possível verificar a concentração de *flats* por bairro em porcentagem, da menor para a maior concentração. Na Figura 15 tem-se o mapa de João Pessoa com os bairros presentes no banco de dados analisado.

Tabela 1: Flats por bairro em porcentagem.

Bairro	Flats/Bairro (%)
Jardim Luna	0,10%
Altiplano	0,30%
Brisamar	0,30%
Bancários	0,40%
Estados	0,40%
Jardim São Paulo	0,40%
Miramar	1,00%
Aeroclube	1,40%
Tambauzinho	1,70%
Tambaú	7,40%
Manaíra	13,90%
Bessa	16,00%
Cabo Branco	24,10%
Jardim Oceania	32,60%

Fonte: Elaborado pelo autor (2025).

Figura 15: Bairros com flats disponíveis no banco de dados.



Fonte: Elaborado pelo autor (2025).

3.3.2 Transformação de Variáveis Categóricas

Nos casos em que temos variáveis categóricas com apenas duas categorias, como, por exemplo, “sim” e “não”, foi necessário criar variáveis *dummy*, que transformam essas categorias em números. Por exemplo, para x_1 igual a “piscina”, em que a resposta é “sim” ou “não”, temos que:

$$x_1 = piscina = \begin{Bmatrix} 1, se\ possui\ piscina \\ 0, se\ não\ possui\ piscina \end{Bmatrix}$$

Essa nova variável, agora numérica, pode ser usada diretamente no modelo de regressão. Para essa transformação foi utilizada a biblioteca *Scikit-learn*, que dispõe de ferramentas para análise preditiva dos dados. Duas técnicas de transformação foram usadas: *One-hot Encoding* e o *Label Encoder*.

A transformação por *One-hot Encoding* transforma cada categoria em uma nova coluna binária, evitando, assim, que o modelo interprete hierarquia entre as variáveis. Enquanto isso, o *Label Encoder* transforma cada categoria em um número inteiro único. Por exemplo, no caso do bairro, cada um teve uma numeração atribuída. Essa numeração pode atribuir uma hierarquia que não existe entre as variáveis.

Na Figura 16 temos o *dataframe* após a transformação das variáveis categóricas por meio da ferramenta *One-hot Encoding*. Ao comparar com a Figura 13 (*dataframe* inicial) é possível verificar que o número de colunas passou de 11 para 32 colunas. Isso ocorre devido à criação de uma nova coluna para cada variável.

Figura 16: Dataframe após transformação por One-hot Encoding.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	20.0	475000.0
1	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	30.0	520000.0
2	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	22.0	299800.0
3	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	47.0	280000.0
4	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	26.0	319000.0

5 rows × 32 columns

Fonte: Elaborado pelo autor (2025).

Enquanto, na Figura 17, temos o *dataframe* após a transformação das variáveis categóricas por meio da ferramenta *Label Encoder*. Diferentemente da ferramenta *One-hot Encoding*, o número de colunas manteve-se igual, já que cada categoria recebeu um número inteiro. Como citado anteriormente, cada bairro teve uma

numeração atribuída, por exemplo, o bairro Cabo Branco foi representado pelo número 5, contudo, isto não quer dizer que ele representa o 5º no lugar na ordem dos bairros.

Figura 17: Dataframe após transformação por Label Encoder.

	0	1	2	3	4	5	6	7	8	9	10
0	20	1	1	1	1	1	1	1	5	1	475000
1	30	1	1	1	1	1	1	1	5	0	520000
2	22	1	1	1	1	1	1	0	7	0	299800
3	47	1	0	0	1	0	0	1	10	1	280000
4	26	1	1	1	1	1	1	1	10	1	319000

Fonte: Elaborado pelo autor (2025).

3.3.3 Padronização das Variáveis

Levando em consideração que o conjunto de dados deste trabalho possui variáveis com escalas de valores diferentes, como por exemplo, área em m² com valor máximo de 55 m² e o valor em reais com valor máximo de R$ 865.250,00, os algoritmos podem dar mais peso ao valor do que à área. A fim de evitar esta situação, foi necessário realizar a padronização das variáveis.

Para padronizar as variáveis, foi utilizada a ferramenta *StandardScaler* da biblioteca *Scikit-learn.* Por meio do *StandardScaler* todas as variáveis ficam com a mesma escala e importância, ajudando o modelo a compreender os dados da melhor maneira possível. Conforme o documento da biblioteca *Scikit-learn, o* escalonamento, que é a padronização através do *StandardScaler*, consiste em remover a média e ajustar a variância, a fórmula usada é:

$$z = \frac{x - \mu}{\sigma}$$

onde:

x = valor original da variável;

μ = média dos dados;

σ = desvio padrão dos dados.

Nas Figuras 18 e 19, encontram-se os *dataframes* escalonados com as variáveis categóricas transformadas com *One-hot Encoding* e *Label Encoder*, respectivamente.

Figura 18: Dataframe escalonado com variáveis categóricas transformadas com One-hot Encoding.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	-0.09848	0.09848	-0.229802	0.229802	-0.550635	0.550635	-0.859305	0.859305	-0.639549	0.639549	...	-0.032686	-0.065477	-0.390540	-0.103863	-0.135935	-0.284145	-1.349922	1.349922	-0.767872	1.039203
1	-0.09848	0.09848	-0.229802	0.229802	-0.550635	0.550635	-0.859305	0.859305	-0.639549	0.639549	...	-0.032686	-0.065477	-0.390540	-0.103863	-0.135935	-0.284145	0.740784	-0.740784	0.580762	1.484840
2	-0.09848	0.09848	-0.229802	0.229802	-0.550635	0.550635	-0.859305	0.859305	-0.639549	0.639549	...	-0.032686	-0.065477	-0.390540	-0.103863	-0.135935	-0.284145	0.740784	-0.740784	-0.498145	-0.695811
3	-0.09848	0.09848	4.351571	-4.351571	1.816085	-1.816085	-0.859305	0.859305	1.563603	-1.563603	...	-0.032686	-0.065477	2.560557	-0.103863	-0.135935	-0.284145	-1.349922	1.349922	2.873440	-0.891891
4	-0.09848	0.09848	-0.229802	0.229802	-0.550635	0.550635	-0.859305	0.859305	-0.639549	0.639549	...	-0.032686	-0.065477	2.560557	-0.103863	-0.135935	-0.284145	-1.349922	1.349922	0.041308	-0.505673

5 rows × 32 columns

Fonte: Elaborado pelo autor (2025).

Figura 19:Dataframe escalonado com variáveis categóricas transformadas com Label Encoder.

	0	1	2	3	4	5	6	7	8	9	10
0	-0.767872	0.09848	0.229802	0.550635	0.859305	0.639549	1.161194	1.303951	-0.567993	1.349922	1.039203
1	0.580762	0.09848	0.229802	0.550635	0.859305	0.639549	1.161194	1.303951	-0.567993	-0.740784	1.484840
2	-0.498145	0.09848	0.229802	0.550635	0.859305	0.639549	1.161194	-0.766900	0.100192	-0.740784	-0.695811
3	2.873440	0.09848	-4.351571	-1.816085	0.859305	-1.563603	-0.861183	1.303951	1.102469	1.349922	-0.891891
4	0.041308	0.09848	0.229802	0.550635	0.859305	0.639549	1.161194	1.303951	1.102469	1.349922	-0.505673

Fonte: Elaborado pelo autor (2025).

3.4 Implementação dos Algoritmos

Após o tratamento das variáveis, foi possível iniciar a etapa de implementação dos algoritmos. Foram testados os seguintes algoritmos de aprendizado de máquina: Regressão Linear Múltipla, Regressão por Vetores de Suporte, Árvore de Decisão, *Random Forest*, *XGBoost*, *LightGBM* e Redes Neurais. Os algoritmos foram implementados em *Python* por meio das bibliotecas *Scikit-learn*, *XGBoost* e *LightGBM*.

Inicialmente, dividiu-se o conjunto de dados em dois grupos, os dados de treino com 70% da amostra e os dados de teste com 30% da amostra. Essa divisão é realizada de forma aleatória, por meio da função *train_test_split* da biblioteca *Scikit-learn*. Os dados de treino são utilizados para que o modelo “aprenda” a relação entre as variáveis independentes e a variável dependente, enquanto os dados de teste avaliam se o modelo “aprendeu” bem e se consegue fazer previsões corretas com novos dados.

Primeiramente, foram processados os dados com todas as variáveis coletadas, porém, devido ao baixo coeficiente de determinação médio obtido, foram realizados

ajustes através de uma seleção de variáveis (*feature selection*). A seleção de variáveis tem como objetivo escolher quais variáveis devem ser utilizadas no modelo de aprendizado de máquina.

A seleção de variáveis iniciou-se por meio da seleção manual – uma técnica baseada no conhecimento. A primeira seleção manual resultou na escolha das variáveis: área, lavanderia, *coworking*, academia, bairro e valor. Estas variáveis foram escolhidas considerando a praticidade para o inquilino em ter estas facilidades no prédio. Manteve-se também o bairro do *flat* e a área em m².

Após verificar que não houve melhora no coeficiente de determinação médio, foi realizada uma segunda seleção manual, sendo estas: área, bairro, situação e valor. Estas variáveis foram escolhidas para eliminar as variáveis relacionada às facilidades do empreendimento, já que, comumente, não são levadas em consideração nos modelos de previsão de preços. Manteve-se, portanto, a área em m², o bairro do *flat* e a sua situação.

Ao processar os modelos com a segunda seleção de variáveis, verificou-se uma melhora nos coeficientes de determinação; porém, ainda insignificante. Em vista disso, uma outra técnica de seleção de variáveis foi implementada, a *Forward Selection*, que consiste em iniciar com nenhuma variável e ir adicionando as que mais importam. Para a *Forward Selection*¸ foram utilizadas as bibliotecas Scikit-learn, XGBoost e LightGBM, que forneceram as importâncias de cada variável para cada modelo executado, sendo selecionadas as variáveis com importância maior ou igual a 75%.

Como resultado da *Forward Selection*¸ obteve-se uma tabela com as variáveis importantes para cada modelo, em que "*true*" representa as variáveis importantes e "false" indica aquelas que não são importantes para o modelo (Figura 20). Assim, optou-se por processar o modelo com as variáveis importantes em pelo menos quatro dos seis modelos para análise final das métricas de desempenho. Essas variáveis foram: área, situação e os bairros Cabo Branco e Jardim Oceania. Após esta seleção, dos 1000 dados iniciais coletados, restaram 533 registros.

Figura 20: Análise da importância das variáveis através da forward selection.

	RFE_SVR	DecisionTree	RandomForest	XGBoost	LightGBM	MLP_Permutation	TotalSelected
ÁREA	True	True	True	True	True	True	6
BAIRRO_CABO BRANCO	True	True	True	True	False	True	5
BAIRRO_JARDIM OCEANIA	False	True	True	True	True	True	5
SITUAÇÃO	True	True	True	False	True	False	4
PLAYGROUND	False	True	True	False	True	False	3
BAIRRO_TAMBAÚ	True	True	False	True	False	False	3
SALAO FESTAS	False	False	True	False	True	False	2
BAIRRO_ESTADOS	False	False	False	True	False	False	1
COWORKING	False	False	False	False	True	False	1
LAVANDERIA	False	False	False	False	False	True	1
PISCINA	False	False	False	False	False	True	1
ACADEMIA	False	False	False	False	False	True	1
BAIRRO_ALTIPLANO	False	False	False	True	False	False	1
BAIRRO_BESSA	True	False	False	False	False	False	1
GOURMET	False	False	False	False	False	False	0
BAIRRO_BANCÁRIOS	False	False	False	False	False	False	0
BAIRRO_BRISAMAR	False	False	False	False	False	False	0
BAIRRO_JD LUNA	False	False	False	False	False	False	0
BAIRRO_JD SÃO PAULO	False	False	False	False	False	False	0
BAIRRO_MANAÍRA	False	False	False	False	False	False	0
BAIRRO_MIRAMAR	False	False	False	False	False	False	0
BAIRRO_TAMBAUZINHO	False	False	False	False	False	False	0

Fonte: Elaborado pelo autor (2025).

3.5 Parametrização dos Algoritmos

3.5.1 Construção do Modelo com Regressão Linear Múltipla

Conforme o documento da biblioteca Scikit-learn, a regressão linear ajusta um modelo com coeficientes $w = w_1, \ldots, w_p$, para minimizar a soma residual dos quadrados entre os alvos observados no conjunto de dados e os alvos previstos pela aproximação linear. O modelo foi criado utilizando a classe *LinearRegression*, que aplica o método dos mínimos quadrados ordinários para ajustar uma equação linear aos dados.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 2 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 2: Métricas de desempenho do modelo de Regressão Linear Múltipla.

Dataframe	**R²**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,4075	0,5990	0,7152	0,8457
1ª Seleção Manual	0,3997	0,6125	0,7180	0,8474
2ª Seleção Manual	0,3890	0,5840	0,6407	0,8004
Forward Selection	0,4112	0,5330	0,5312	0,7289

Fonte: Elaborado pelo autor (2025).

3.5.2 Construção do Modelo com SVR

A classe implementada para o modelo de Regressão por Vetores de Suporte foi a SVR que utiliza a biblioteca *sklearn*.SVM. A Regressão por Vetores de Suporte busca encontrar uma função que desvie no máximo ε (*episilon*) dos valores reais, ao mesmo tempo em que minimiza a complexidade do modelo (Scikit-learn, 2025). O modelo foi parametrizado com a função *kernel* tipo RBF (*Radial Basis Function*), que é adequado para capturar relações não lineares entre as variáveis e com margem de tolerância $\varepsilon = 0.1$.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 3 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 3:Métricas de desempenho do modelo de Regressão por Vetores de Suporte.

Dataframe	**R²**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,4220	0,5897	0,7626	0,8732
1ª Seleção Manual	0,4048	0,5966	0,7480	0,8649
2ª Seleção Manual	0,4238	0,5396	0,6193	0,7870
Forward Selection	0,4227	0,4943	0,4980	0,7057

Fonte: Elaborado pelo autor (2025).

3.5.3 Construção do Modelo com Árvore de Decisão

O modelo com Árvore de Decisão foi implementado por meio da classe *DecisionTreeRegressor* da biblioteca *Scikit-learn*. O modelo foi parametrizado com uma profundidade máxima da árvore (*max_depth*) de 5, com o objetivo de limitar o crescimento da árvore e evitar *overfitting*, o *random_state* de 10, que garante a reprodutibilidade dos resultados, e os demais parâmetros foram mantidos com os valores padrão da biblioteca (*default*).

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 4 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 4:Métricas de desempenho do modelo de Árvore de Decisão.

Dataframe	**R^2**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,3379	0,6456	0,8331	0,9127
1ª Seleção Manual	0,3042	0,6553	0,8509	0,9224
2ª Seleção Manual	0,3238	0,6345	0,8117	0,9009
Forward Selection	0,3756	0,5206	0,5544	0,7446

Fonte: Elaborado pelo autor (2025).

3.5.4 Construção do Modelo com Random Forest

Para a modelagem do algoritmo *Random Forest*, foi implementada a classe *RandomForestRegressor* da biblioteca *Scikit-learn*. Esse algoritmo é uma técnica de aprendizado de máquina baseada em conjuntos (*ensemble*), que combina múltiplas árvores de decisão treinadas sobre subconjuntos aleatórios dos dados, resultando em um modelo mais robusto, menos propenso a *overfitting* e capaz de capturar relações não lineares.

O modelo foi configurado com os seguintes hiperparâmetros:

- *n_estimators* = 100: define o número de árvores de decisão que compõem a floresta;
- *criterion = 'squared_error'*: utiliza o erro quadrático como função de perda para medir a qualidade das divisões em cada árvore;
- *max_depth* = 5: limita a profundidade máxima das árvores, controlando a complexidade do modelo e prevenindo overfitting;
- *random_state* = 10: garante a reprodutibilidade dos resultados.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 5 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 5:Métricas de desempenho do modelo de Random Forest.

Dataframe	**R²**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,4292	0,6006	0,7256	0,8577
1ª Seleção Manual	0,3930	0,6105	0,7461	0,8638
2ª Seleção Manual	0,3835	0,6027	0,7019	0,8378
Forward Selection	0,4260	0,5123	0,5198	0,7210

Fonte: Elaborado pelo autor (2025).

3.5.5 Construção do Modelo com *XGBoost*

O modelo com *XGBoost* foi implementado por meio da classe *XGBRegressor* da biblioteca *xgboost*. O *XGBoost* é um algoritmo baseado em gradiente *boosting*, que constrói sequencialmente um conjunto de árvores de decisão, onde cada nova árvore é treinada para corrigir os erros cometidos pelas anteriores. O modelo foi construído com os seguintes hiperparâmetros:

- *n_estimators* = 180: número total de árvores a serem construídas (iterações de *boosting*);
- *max_depth* = 3: profundidade máxima das árvores, controlando a complexidade de cada estimador;
- *learning_rate* = 0.05: taxa de aprendizado que regula a contribuição de cada árvore na predição final;
- *objective* = *'reg:squarederror'*: função objetivo usada para regressão com erro quadrático;
- *random_state* = 10: para garantir reprodutibilidade dos resultados.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 6 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 6:Métricas de desempenho do modelo XGBoost.

Dataframe	**R²**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,4118	0,5918	0,6954	0,8339
1ª Seleção Manual	0,3926	0,6162	0,7504	0,8662
2ª Seleção Manual	0,4027	0,5728	0,6631	0,8143
Forward Selection	0,4251	0,5120	0,5192	0,7206

Fonte: Elaborado pelo autor (2025).

3.5.6 Construção do Modelo com *Light* GBM

O modelo com Light GBM foi construído através da classe LGBMRegressor da biblioteca *lightgbm*. Esse algoritmo também é baseado no princípio de gradiente boosting, sendo altamente eficiente tanto em termos de tempo de execução quanto de consumo de memória. O modelo foi parametrizado com os seguintes hiperparâmetros:

- *num_leaves* = 50: número máximo de folhas por árvore, influenciando diretamente a complexidade do modelo;
- *max_depth* = 3: profundidade máxima das árvores, utilizada como controle adicional de complexidade;
- *learning_rate* = 0.1: taxa de aprendizado que regula a influência de cada árvore sobre o modelo final;
- *n_estimators* = 100: número total de árvores no modelo;
- *random_state* = 10: define a semente aleatória, garantindo a reprodutibilidade dos resultados.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 7 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 7:Métricas de desempenho do modelo LightGBM.

Dataframe	**R^2**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,3941	0,5730	0,6929	0,8324
1ª Seleção Manual	0,3788	0,6009	0,7239	0,8508
2ª Seleção Manual	0,3737	0,5798	0,6721	0,8198
Forward Selection	0,4189	0,5038	0,5354	0,7317

Fonte: Elaborado pelo autor (2025).

3.5.7 Construção do Modelo com Rede Neural

Para o modelo de Rede Neural Artificial Artificial do tipo *Multi-Layer Perceptron* (MLP), implementado por meio da classe MLPRegressor da biblioteca *Scikit-learn*. As redes neurais são particularmente eficazes para capturar padrões não lineares e interações complexas entre variáveis. O modelo foi parametrizado com os seguintes hiperparâmetros:

- *hidden_layer_sizes* = (100, 50): define uma rede com duas camadas ocultas, sendo a primeira com 100 neurônios e a segunda com 50;
- *activation* = '*relu*': função de ativação ReLU (*Rectified Linear Unit*), que ajuda a acelerar o treinamento e evita problemas de gradientes nulos;
- *solver* = *'adam'*: algoritmo de otimização eficiente baseado em gradiente estocástico adaptativo;
- *max_iter* = 1000: número máximo de iterações para o processo de treinamento;
- *random_state* = 10: define uma semente aleatória, garantindo reprodutibilidade dos resultados.

O modelo foi treinado com o conjunto de treino e, posteriormente, foram realizadas previsões sobre o conjunto de teste. Para avaliar o desempenho do modelo, foram utilizadas as seguintes métricas: Coeficiente de Determinação Médio (R^2), Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE). Na Tabela 8 é possível visualizar as métricas obtidas para cada *dataframe*.

Tabela 8:Métricas de desempenho do modelo com Rede Neural.

Dataframe	**R^2**	**MAE**	**MSE**	**RMSE**
Dados Iniciais	0,3714	0,5575	0,6088	0,7802
1ª Seleção Manual	0,4146	0,5804	0,6730	0,8203
2ª Seleção Manual	0,4302	0,5392	0,6006	0,7750
Forward Selection	0,4219	0,5183	0,5224	0,7228

Fonte: Elaborado pelo autor (2025).

4 RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados obtidos com a implementação dos algoritmos de aprendizado de máquina para cada conjunto de dados.

4.1 Dados Iniciais

A modelagem dos algoritmos iniciou-se com o banco de dados iniciais, retirados apenas os *missing values*. Para estes dados, foram treinados sete algoritmos diferentes para comparação dos resultados e análise dos que tiveram melhor desempenho.

Na Tabela 9, tem-se o valor de 0,4292 para o Coeficiente de Determinação Médio (R^2) de maior valor, referente ao modelo com *Random* Forest. Contudo, apresenta um Erro Médio Absoluto de 0,6006. Em seguida, o segundo melhor desempenho foi com o modelo de Regressão por Vetores de Suporte, com Coeficiente de Determinação Médio (R^2) igual a 0,4220 e Erro Médio Absoluto de 0,5897.

Tabela 9: Métricas de desempenho dos dados iniciais por algoritmo.

Algoritmo	R^2	MAE	MSE	RMSE
Regressão Linear Múltipla	0,4075	0,5990	0,7152	0,8457
SVR	0,4220	0,5897	0,7626	0,8732
Árvore de Decisão	0,3379	0,6456	0,8331	0,9127
Random Forest	0,4292	0,6006	0,7256	0,8577
XGBoost	0,4118	0,5918	0,6954	0,8339
Light GBM	0,3941	0,5730	0,6929	0,8324
Rede Neural	0,3714	0,5575	0,6088	0,7802

Fonte: Elaborado pelo autor (2025).

O Coeficiente de Determinação Médio, baixo em todos os modelos, sugere que os algoritmos estão capturando parcialmente a variabilidade dos preços, ou seja, os algoritmos estão acertando apenas entre 33% e 43% das previsões de preços dos *flats*.

4.2 Primeira Seleção Manual

Como estratégia para aumentar o desempenho dos modelos, reduziu-se os atributos considerados irrelevantes por meio de seleções manuais. Os atributos da primeira seleção manual foram: área, lavanderia, *coworking*, academia, bairro e valor. A amostra dos dados permaneceu a mesma, tendo sido retirados apenas os valores ausentes (*missing values*).

No geral, a primeira seleção manual das variáveis não beneficiou a maioria dos modelos, exceto o algoritmo de Rede Neural, no qual o Coeficiente de Determinação Médio (R^2) aumentou de 0,3714 para 0,4146, um aumento de 4,32%, ainda considerado baixo. Entretanto, todos os demais modelos tiveram redução do R^2 e aumento do MAE, sendo o *Random Forest* e a Árvore de Decisão os modelos que mais sofreram com a seleção.

Tabela 10: Métricas de desempenho da primeira seleção manual por algoritmo.

1ª Seleção Manual	R^2	MAE	MSE	RMSE
Regressão Linear Múltipla	0,3997	0,6125	0,7180	0,8474
SVR	0,4048	0,5966	0,7480	0,8649
Árvore de Decisão	0,3042	0,6553	0,8509	0,9224
Random Forest	0,3930	0,6105	0,7461	0,8638
XGBoost	0,3926	0,6162	0,7504	0,8662
Light GBM	0,3788	0,6009	0,7239	0,8508
Rede Neural	0,4146	0,5804	0,6730	0,8203

Fonte: Elaborado pelo autor (2025).

4.3 Segunda Seleção Manual

Mais uma seleção manual para melhoramento dos resultados foi realizada, sendo as variáveis escolhidas: área, bairro, situação e valor. O modelo de Rede Neural, novamente, obteve o melhor desempenho geral, com Coeficiente de Determinação Médio de 0,4302. Os erros médio absoluto e quadrático apresentaram os melhores resultados. Já o modelo de Árvore de Decisão manteve o pior desempenho, o que sugere uma limitação do modelo com as variáveis disponíveis.

Tabela 11: Métricas de desempenho da segunda seleção manual por algoritmo.

2ª Seleção Manual	R^2	MAE	MSE	RMSE
Regressão Linear Múltipla	0,3890	0,5840	0,6407	0,8004
SVR	0,4238	0,5396	0,6193	0,7870
Árvore de Decisão	0,3238	0,6345	0,8117	0,9009
Random Forest	0,3835	0,6027	0,7019	0,8378
XGBoost	0,4027	0,5728	0,6631	0,8143
Light GBM	0,3737	0,5798	0,6721	0,8198
Rede Neural	0,4302	0,5392	0,6006	0,7750

Fonte: Elaborado pelo autor (2025).

Dentre os três conjuntos de dados analisados até o momento, a segunda seleção manual teve o melhor desempenho geral, devido à redução de variáveis que não contribuem para a previsão dos preços dos *flats*. Entretanto, os modelos seguem acertando apenas entre 33% e 43% das previsões de preços dos *flats*. As seleções

manuais sugerem que a seleção baseada em critérios subjetivos não beneficia os modelos.

4.4 Forward Selection

A técnica de seleção de variáveis *Forward Selection* foi aplicada com o objetivo de aprimorar o desempenho dos modelos preditivos, por meio da escolha automática das variáveis mais relevantes para o problema de previsão de preços. Dessa forma, critérios subjetivos foram excluídos da seleção dos atributos. A *Forward Selection* consiste em adicionar gradualmente cada variável ao modelo, considerando aquelas que mais contribuem para a melhoria das métricas de desempenho.

Conforme citado anteriormente, na seção de Implementação dos Algoritmos, foram escolhidas as variáveis consideradas importantes em pelo menos quatro dos seis modelos. As variáveis foram: área, situação, bairros Cabo Branco e Jardim Oceania, e valor. Devido a redução dos bairros em relação aos demais conjuntos de dados, a nova amostra possui 533 registros para serem analisados.

Tabela 12: Métricas de desempenho da seleção foward por algoritmo.

Forward Selection	R^2	MAE	MSE	RMSE
Regressão Linear Múltipla	0,4112	0,5330	0,5312	0,7289
SVR	0,4227	0,4943	0,4980	0,7057
Árvore de Decisão	0,3756	0,5206	0,5544	0,7446
Random Forest	0,4260	0,5123	0,5198	0,7210
XGBoost	0,4251	0,5120	0,5192	0,7206
Light GBM	0,4189	0,5038	0,5354	0,7317
Rede Neural	0,4219	0,5183	0,5224	0,7228

Fonte: Elaborado pelo autor (2025).

Com a aplicação da *Forward Selection*, observou-se uma melhora no desempenho dos modelos. O algoritmo Random Forest apresentou o maior Coeficiente de Determinação Médio, demonstrando maior capacidade de explicar a variância dos preços com as variáveis selecionadas. Por outro lado, o modelo SVR se destacou com os menores valores de erro absoluto e quadrático (MAE = 0,4943; MSE = 0,4980; RMSE = 0,7057), sendo o mais preciso nas previsões pontuais.

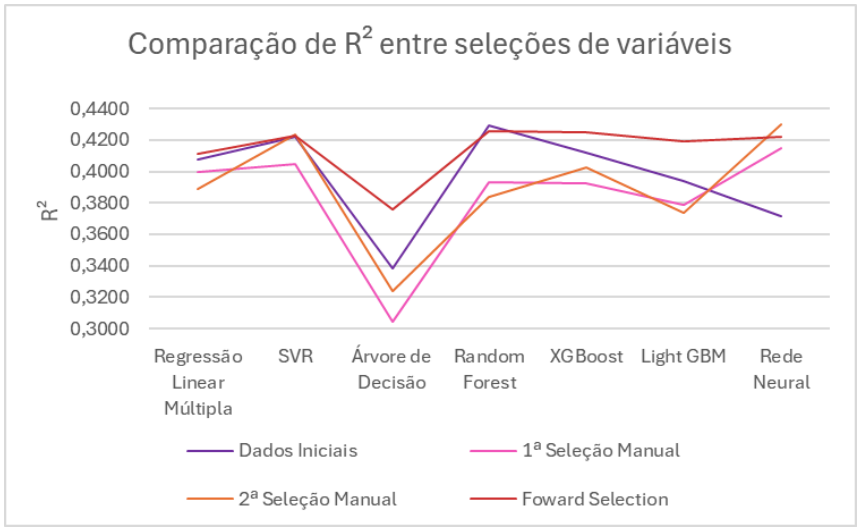
4.5 Comparação Entre Seleções de Variáveis

Com os resultados obtidos a partir dos modelos criados para cada seleção de variáveis, elaborou-se um gráfico de comparação dos coeficientes de determinação (Figura 22) e um gráfico de comparação das raízes do erro quadrático médio (Figura

23). O gráfico de R² mostra que, embora nenhum algoritmo tenha ultrapassado o valor de 0,45, o uso da *Forward Selection* melhorou significativamente o desempenho dos modelos.

O modelo de Rede Neural apresentou o maior valor para o Coeficiente de Determinação Médio (0,4302) na segunda seleção manual e manteve o bom desempenho com a *Forward Selection* (0,4219). Os modelos *Random Forest*, *XGBoost* e SVR também apresentaram melhoras com a seleção *Forward*, o que reforça a importância da escolha correta dos atributos para modelos de aprendizado de máquina.

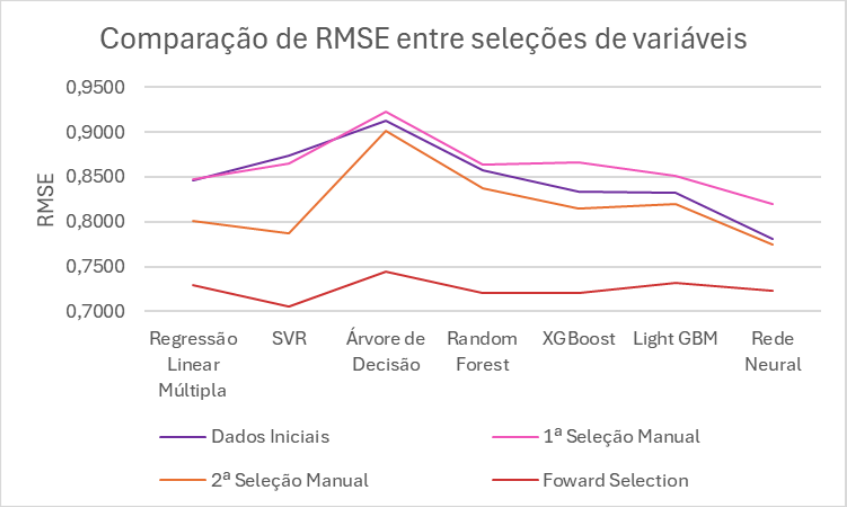
Figura 21: Gráfico de comparação de R² entre seleções de variáveis.



Fonte: Elaborado pelo autor (2025).

Quanto ao gráfico de RMSE, a seleção *Forward* destaca-se com os menores valores da Raiz do Erro Quadrático Médio, com destaque para o modelo SVR (0,7057) e Rede Neural (0,7228). A Regressão Linear também apresentou uma redução significativa no RMSE com a *Forward Selection* (0,7289). O algoritmo de Árvore de Decisão mostrou mais uma vez um baixo desempenho, teve consistentemente os maiores erros.

Figura 22: Gráfico de comparação de RMSE entre seleções de variáveis.



Fonte: Elaborado pelo autor (2025).

Portanto, a *Forward Selection* foi a estratégia que gerou os melhores resultados, ainda assim indicando previsões com limitações, o que sugere a necessidade de melhorias. É possível que esses resultados sejam decorrentes da ausência de fatores importantes nos dados, como, por exemplo, informações mais detalhadas da localização, o nível de sofisticação do empreendimento, o andar em que a unidade está localizada, entre outros fatores.

Dentro do bairro Cabo Branco, por exemplo, é possível que o *flat* esteja localizado na Avenida Cabo Branco, sendo, portanto, um empreendimento “pé na areia”, pois está situado de frente para a praia, ou esteja localizado em uma rua que não tenha acesso direto à praia. Da mesma forma, no mesmo empreendimento, é possível que um *flat* tenha vista para a praia e outro não. Logo, dois imóveis com a mesma área, mesmas facilidades (lavanderia, *coworking*, academia etc.) e no mesmo prédio, sendo o primeiro com vista para o mar e o segundo com vista para outros empreendimentos, terão preços diferentes, com o primeiro sendo mais caro. Outro fator que ainda pode influenciar a precificação do imóvel, ainda dentro do mesmo empreendimento, é o andar que se encontra; imóveis em andares mais altos tendem a ser mais caros devido à vista privilegiada.

A sofisticação do empreendimento é um fator interessante, porém subjetivo. Com o crescimento da construção civil na cidade de João Pessoa, as construtoras também cresceram e muitas tornaram-se referência no Estado. Muitas construtoras, além de venderem o empreendimento em si, também vendem o nome da marca. Investimentos em arquitetos renomados para liderar os projetos e prédios que têm

como conceito arquitetônico carros de luxo, são exemplos de como agregar valor ao empreendimento.

Quanto às variáveis como número de quartos, banheiros, vagas de estacionamento e suítes, geralmente analisadas em estudos semelhantes, não têm significância para a previsão de preços de venda de *flats*, uma vez que a maior parte dos empreendimentos tipo *flat* possuem um ambiente integrado de sala, cozinha e quarto, e apenas um banheiro. Quanto às garagens, os novos empreendimentos têm adotado o sistema de vaga rotativa para os condôminos.

Dessa forma, embora os ajustes e métodos empregados tenham trazido ganhos pontuais de desempenho, ainda há amplo espaço para aprimoramento, especialmente se o objetivo for construir um sistema confiável e preciso para apoio à tomada de decisão no mercado imobiliário.

5 CONCLUSÕES

Este trabalho teve como objetivo principal implementar e avaliar diferentes algoritmos de aprendizado de máquina para a previsão de preços de venda de flats na cidade de João Pessoa–PB, a partir de dados coletados do site Viva Real. A análise abrangeu desde a coleta e o pré-processamento dos dados até a construção e validação de modelos preditivos, com o intuito de identificar o algoritmo com melhor desempenho na estimativa de preços. Os modelos avaliados incluíram Regressão Linear Múltipla, Regressão por Vetores de Suporte (SVR), Árvore de Decisão, Random Forest, XGBoost, LightGBM e Redes Neurais.

Embora todos os modelos tenham sido capazes de capturar parcialmente a relação entre as variáveis e o preço dos flats, os coeficientes de determinação médios (R^2) obtidos foram considerados baixos, o que indica uma capacidade limitada dos modelos em explicar a variabilidade dos preços com os dados disponíveis. Dessa forma, os resultados não atingiram plenamente as expectativas, evidenciando um desempenho insatisfatório na predição precisa dos valores de venda dos imóveis analisados.

Foram identificadas algumas limitações importantes ao longo do estudo. A primeira ocorreu na etapa de coleta dos dados, realizada de forma manual para garantir o maior detalhamento possível das informações. No entanto, a falta de padronização nos anúncios do site Viva Real dificultou a uniformização das variáveis, além de resultar em diversas lacunas de dados. Outra limitação relevante foi a disponibilidade restrita de variáveis: por se tratar de flats, atributos como número de quartos, vagas de garagem ou área externa — frequentemente utilizados em modelos imobiliários — não estavam presentes, o que pode ter impactado a capacidade explicativa dos modelos.

Adicionalmente, fatores como o número limitado de variáveis relevantes, a ausência de dados detalhados sobre a localização precisa, o andar do imóvel e o nível de sofisticação do empreendimento podem ter comprometido a acurácia das previsões. A subjetividade de alguns desses aspectos, como o padrão de acabamento ou o prestígio da construtora, representa um desafio adicional à modelagem preditiva, que depende de dados estruturados e mensuráveis.

Quadro 4: Limitações e sugestões para trabalhos futuros.

Limitação	Sugestão para Trabalhos Futuros
Coleta manual e limitada de dados	Automatizar a coleta por meio de *web scraping* ou APIs, coletando dados de diferentes fontes.
Baixo desempenho dos modelos (R^2)	Incluir novas variáveis explicativas (localização via coordenadas, andar, proximidade de serviços).
Dados homogêneos (flats com poucas diferenças estruturais)	Ampliar a amostra para incluir diferentes tipos de imóveis (casas, apartamentos, terrenos etc.).
Falta de integração com dados externos	Integrar com bases públicas variáveis externas como renda média, criminalidade ou turismo.

Fonte: Elaborado pelo autor (2025).

Apesar dessas limitações, os resultados obtidos mostram que os algoritmos de aprendizado de máquina oferecem uma alternativa promissora e inovadora para a previsão de preços de imóveis, tornando as estimativas mais objetivas e fundamentadas em dados. O estudo evidencia que, mesmo com restrições, essas técnicas possuem grande potencial de aplicação no mercado imobiliário, especialmente quando associadas a bases de dados mais completas e variáveis mais representativas.

REFERÊNCIAS

ALPAYDIN, Ethem. **Introduction to Machine Learning**. MIT Press, 2014.

ARAÚJO, Raquel. ***O Perceptron – A Estrutura Base de Redes Neurais e Deep Learning*.** Hashtag Treinamentos, 2023. Disponível em: https://www.hashtagtreinamentos.com/o-perceptron-ciencia-de-dados. Acesso em: 10 abr. 2025.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14653-2**: Avaliação de bens - Parte 2: Imóveis urbanos. Rio de Janeiro: ABNT, 2011.

BANERJEE, Prashant. ***LightGBM Classifier in Python*.** Kaggle, 2021. Disponível em: https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python. Acesso em: 10 abr. 2025.

BREIMAN, Leo. **Random Forests.** Kluwer Academic Publishers. Machine Learning, 45, 5–32, 2001.

CBIC – Câmara Brasileira da Indústria da Construção. **Construção Civil cresce 4,3% em 2024 e impulsiona economia nacional.** 7 mar. 2025. Disponível em: https://cbic.org.br/construcao-civil-cresce-43-em-2024-e-impulsiona-economia-nacional/. Acesso em: 09 abr. 2025.

CECCON, Denny. **Os Tipos de Redes Neurais.** IA Expert Academy, 2020. Disponível em: https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/. Acesso em: 10 abr. 2025.

CHEN, Tianqi; GUESTRIN, Carlos. **XGBoost: A Scalable Tree Boosting System.** Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. p. 785–794.

DEEP LEARNING BOOK. **As 10 Principais Arquiteturas de Redes Neurais.** Deep Learning Book, 2022. Disponível em: https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/. Acesso em: 10 abr. 2025.

DIAS, Willamy. **Machine learning e a previsão de preços de terrenos em Brasília.** 2023. 69 f. Dissertação (Mestrado Profissional em Economia) – Instituto Brasileiro de Ensino, Desenvolvimento e Pesquisa, Programa de Pós-Graduação em Economia, Políticas Públicas e Desenvolvimento, Brasília, 2023.

FERNEDA, Edberto. **Redes Neurais e sua Aplicação em Sistemas de Recuperação de Informação.** Ciência da Informação, Brasília, 2006, p. 25-30

FRANCISCO, António Muinga. **Aplicação de Modelos de Machine Learning para Previsão do Housing Price em Singapura.** 2024. Dissertação (Mestrado em Economia Monetária e Financeira) – ISCTE – Instituto Universitário de Lisboa, Lisboa, 2024.

FUNDAÇÃO INSTITUTO DE PESQUISAS ECONÔMICAS (FIPE). **Índice FipeZAP: Venda Residencial – Informe de Dezembro de 2024**. São Paulo: Fipe, 2024.

Disponível em: https://www.datazap.com.br/wp-content/uploads/2025/01/fipezap-202412-residencial-venda.pdf. Acesso em: 10 abr. 2025.

KREMER, Joelma. **Mercado Imobiliário.** UNIASSELVI, 2008.

LANTZ, Brett. **Machine Learning With R.** 3. ed. Birmingham: Packt Publishing, 2019.

LAUTH, Artur Henrique. **Modelo preditivo para preço de aluguel de apartamento em Blumenau.** 2023. 61 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Santa Catarina, Centro Tecnológico, de Ciências Exatas e Educação, Departamento de Engenharia de Controle, Automação e Computação, Blumenau, 2023.

LEMES, Nelson Henrique Teixeira. **Neurônio de McCulloch-Pitts.** Universidade Federal de Alfenas, 2020. Disponível em: https://pessoas.unifal-mg.edu.br/nelsonlemes/neuronio-de-mcculloch-pitts/. Acesso em: 10 abr. 2025.

LIN, Hong-Lu; CHEN, Kuentai. **Predicting Price of Taiwan Peal Estates by Neural Networks and Support Vector Regression.** Proceedings of the 15th WSEAS international conference on Systems, p. 220–225, 14 jul. 2011.

MATOS, Débora; BARTKIW, Paula Izabela Nogueira. **Introdução ao Mercado Imobiliário.** Instituto Federal do Paraná, 2013.

MEDEIROS, José. **Bancos de Dados Geográficos e Redes Neurais Artificias: Tecnologias de Apoio à Gestão de Território.** Tese de Doutoramento em Geografia Física, Universidade de São Paulo, 1999.

MICROSOFT. ***LightGBM Documentation*.** Disponível em: https://lightgbm.readthedocs.io/en/stable/. Acesso em: 10 abr. 2025.

MIRANDA, Matheus. **Aplicação de aprendizado de máquina na previsão de preços de aluguel de imóveis.** 2023. Artigo de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Municipal de Franca (Uni-FACEF), 2023.

NEPTUNE.AI. **XGBoost: Everything you Need to Know.** Neptune.ai, 2023. Disponível em: https://neptune.ai/blog/xgboost-everything-you-need-to-know. Acesso em: 10 abr. 2025.

NETTO, Amilcar; MACIEL, Francisco. **Python para Data Science e Machine Learning Descomplicado.** Rio de Janeiro: Editora Alta Books, 2021. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9786555203172/. Acesso em: 11 abr. 2025.

OLIVEIRA FILHO, Luiz Carlos de. **Determinação de modelo de aprendizagem de máquina para precificação de imóveis.** 2023. 28 f. Trabalho de Conclusão de Curso (Graduação em Ciências Econômicas) – Universidade Federal do Ceará, Faculdade de Economia, Administração, Atuária e Contabilidade, Fortaleza, 2023.

POTRICH, Yuri Balczareki. **Precificação de imóveis em Florianópolis utilizando técnicas de aprendizado de máquina.** 2024. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção Mecânica) – Universidade Federal de Santa Catarina, Centro Tecnológico, Curso de Engenharia de Produção Mecânica, Florianópolis, 2024.

SCIKIT-LEARN. ***Scikit-learn: Machine Learning in Python***. Disponível em: https://scikit-learn.org. Acesso em: 21 abr. 2025.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. ***Understanding Machine Learning: From Theory to Algorithms*.** Cambridge: Cambridge University Press, 2014.

SICSÚ, Abraham L.; SAMARTINI, André; BARTH, Nelson L**. Técnicas de machine learning.** São Paulo: Editora Blucher, 2023. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9786555063974/. Acesso em: 10 abr. 2025.

SILVA, Gustavo Henrique Pinheiro da. **Modelos de aprendizagem de máquina para precificação de imóveis na cidade de Fortaleza.** 2019. 86 f. Monografia (Graduação em Engenharia Civil) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Estrutural e Construção Civil, Fortaleza, 2019.

SOUSA, Maiara Medeiros de. **Modelo Random Forest aplicado à precificação de imóveis à venda em Aracaju, SE.** 2023. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Estatística) – Universidade Federal de Sergipe, Centro de Ciências Exatas e Tecnologia, Departamento de Estatística e Ciências Atuariais, São Cristóvão, 2023.

SOUZA, Luan. **Mercado imobiliário de João Pessoa dá sinais de crescimento.** CRECI-PB, 2024. Disponível em: https://creci-pb.gov.br/mercado-imobiliario-joao-pessoa-da-sinais-de-crescimento-por-luan-pereira-de-souza/. Acesso em: 10 abr. 2025.

WILLMOTT, Cort J.; MATSUURA, Kenji**. Advantages of the Mean Absolute Error (MAE) Over the Root Nean Square Error (RMSE) in Assessing Average Model Performance.** *Climate Research*, Oldendorf/Luhe, v. 30, p. 79–82, 2005. Disponível em: https://www.int-res.com/articles/cr2005/30/c030p079.pdf. Acesso em: 11 abr. 2025.

XGBOOST Developers. ***XGBoost Python Package — Model tutorial***. XGBoost Documentation. Disponível em: https://xgboost.readthedocs.io/en/stable/tutorials/model.html. Acesso em: 10 abr. 2025.