INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Talles Eduardo Camargo dos Santos

ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL

Trabalho de Graduação
2024

Curso de Engenharia Civil-Aeronáutica

CDU 624:681.3.02

Talles Eduardo Camargo dos Santos

ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL

Orientadora

Profa. Dra. Maryangela Geimba Lima (ITA)

Coorientador

Cap. Eng. Paulo de Tarso Machado Leite Soares (CEPE)

ENGENHARIA CIVIL-AERONÁUTICA

São José dos Campos
Instituto Tecnológico de Aeronáutica

2024

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

dos Santos, Talles Eduardo Camargo
ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL / Talles Eduardo Camargo dos Santos.
São José dos Campos, 2024.
81f.

Trabalho de Graduação – Curso de Engenharia Civil-Aeronáutica– Instituto Tecnológico de Aeronáutica, 2024. Orientadora: Profa. Dra. Maryangela Geimba Lima. Coorientador: Cap. Eng. Paulo de Tarso Machado Leite Soares .

1. Construção civil. 2. Estimativa de custos. 3. Aprendizagem (inteligência artificial). 4. Controle preditivo. 5. Programas de computadores. 6. Algoritmos. 7. Engenharia civil. 8. Computação. I. Instituto Tecnológico de Aeronáutica. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

DOS SANTOS, Talles Eduardo Camargo. **ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL**. 2024. 81f. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSÃO DE DIREITOS

NOME DO AUTOR: Talles Eduardo Camargo dos Santos
TITULO DO TRABALHO: ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL.
TIPO DO TRABALHO/ANO: Trabalho de Conclusão de Curso (Graduação) / 2024

Talles Eduardo Camargo dos Santos
Rua H8B, Ap. 232
12.228-461 – São José dos Campos–SP

ANÁLISE DE MÉTODOS E PROPOSTA DE FERRAMENTA COMPUTACIONAL PARA ESTIMATIVAS PARAMÉTRICAS DE OBRAS DE ENGENHARIA CIVIL

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação

Talles E. C. dos Santos

Talles Eduardo Camargo dos Santos

Autor

Maryangela Geimba Lima (ITA)

Orientadora

Paulo de Tarso Machado Leite Soares (CEPE)

Coorientador

Evandro José da Silva

Coordenador do Curso de Engenharia Civil-Aeronáutica

São José dos Campos, 27 de NOVEMBRO de 2024.

Dedico este trabalho à minha família, pelo amor e apoio incondicionais, e aos meus amigos e mentores, pela inspiração e orientação ao longo desta jornada.

Agradecimentos

Como traduzir em palavras toda a gratidão que sinto por aqueles que caminharam ao meu lado nesta jornada? Refletindo sobre isso, inúmeros nomes surgem em minha mente, pessoas que, de diversas formas, contribuíram para o meu crescimento. Aos que não foram mencionados, peço que considerem o que carrego no coração: os valores que compartilhamos reciprocamente, que possuem um alcance que transcende este breve texto. Esta é apenas uma humilde homenagem para honrar todos que me ajudaram a trilhar o meu caminho.

Em primeiro lugar, dirijo meus sinceros agradecimentos à minha família, especialmente aos meus pais e irmãos, cujo apoio e motivação foram fundamentais para que eu prosseguisse em frente.

Manifesto minha gratidão aos professores e técnicos que contribuíram diretamente para a construção do meu conhecimento, não apenas em Engenharia Civil, mas também em lições de vida, ensinando-me a ser, acima de tudo, humano, valorizando a humildade, o carisma, o espírito crítico e a dedicação.

Agradeço à Prof.ª Dra. Maryangela, ao professor Evandro e à equipe do CEPE pelo inestimável apoio em todas as etapas deste trabalho, desde a coleta de dados até as valiosas orientações fornecidas.

Sou grato à minha turma de Engenharia Civil, composta por pessoas especiais que promoveram um ambiente harmonioso e uma constante troca de conhecimentos. Lucas e Samuel foram fundamentais, formando o triângulo trismegisto crucial para esta conquista.

Agradeço aos amigos que cultivei ao longo da minha trajetória, desde o CMB, Pódion, Farias Brito, Olimpo, até chegar ao ITA. Aos amigos de longa data com quem mantenho contato até hoje (Domingol, Shirley, Convento dos Perdidos e os "basqueteiros" de Brasília) e aos amigos que obtive no H8, minha sincera gratidão.

Por fim, estendo minha apreciação ao ITA e à FAB por fornecer os recursos e instalações necessários para a condução da minha pesquisa.

"Só quem tem coragem pro início
Sabe o peso que é parar
E só na hora que a canoa vira
Que você lembra que sabe nadar."
— Djonga

Resumo

Este trabalho aborda o desenvolvimento de métodos inovadores para estimativa de custos no setor de construção civil, com foco na aplicação de técnicas paramétricas e aprendizado de máquina. Reconhecendo as limitações dos métodos tradicionais, como os baseados no Custo Unitário Básico (CUB), propõe-se a criação de uma ferramenta computacional para prever custos com maior precisão, mesmo em estágios iniciais dos projetos, quando os detalhes são escassos. A metodologia integra dados históricos de 204 obras, disponibilizados pelo Centro de Estudos e Pesquisa de Engenharia da Força Aérea Brasileira (CEPE), com algoritmos avançados de regressão, como Elastic Net, Support Vector Regressor e técnicas baseadas em árvores de decisão (Random Forest e XGBoost). Os melhores algoritmos se apresentaram como ajustes moderados para o contexto, atingindo coeficientes de determinação R^2 próximos de 0,5. Os resultados destacam a evolução de modelos baseados em aprendizado de máquina na captura de interações complexas entre variáveis do projeto, contribuindo para menor risco financeiro e maior eficiência no gerenciamento de recursos. Potenciais melhorias e limitações foram identificadas e discutidas para que a futura implementação possa ser otimizada.

Abstract

The cost forecasting of civil engineering projects is of great importance, especially in tenders conducted annually by the Brazilian Air Force (FAB). Due to the variability of projects in aspects such as location, function, scale, complexity, and typology, estimating costs in the early stages of a project is a challenging task. Parametrization, based on historical series with the characteristics of each project, enables a predictive analysis of the total cost, serving as an important indicator to assess the financial feasibility of the project. Moreover, it contributes to establishing a consistent investment plan for projects, allowing for the evaluation of each project's priority and ensuring that expenditures align with expectations. This study aims to compare machine learning methods for cost estimation in civil engineering projects. Several regression models, such as Ridge Regression, Lasso, Random Forest, CatBoost, among others, were applied to the database provided by the Center for Engineering Studies and Projects (CEPE - São Paulo). The comparison of results seeks to identify the method that delivers the best performance in cost estimation, contributing to the improvement of the planning and management process of investments in civil engineering projects.

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

1.1 Motivação

A estimativa de custos é uma etapa crítica no planejamento de obras de construção civil, sendo essencial para viabilizar projetos dentro de restrições orçamentárias e cronogramas rigorosos já nas fases iniciais. Métodos tradicionais de estimativa, como aqueles baseados em Custo Unitário Básico (CUB), apresentam grandes limitações devido à generalização de dados e à falta de detalhamento nas variáveis de entrada, o que pode levar a desvios significativos entre os custos estimados e os efetivamente realizados.

Para avaliar a viabilidade de um empreendimento, é indispensável apresentar uma estimativa de custos que atenda às expectativas do gestor quanto à precisão requerida. Entretanto, os estudos de viabilidade, fundamentais para a comercialização do empreendimento, são frequentemente realizados em uma etapa preliminar, antes da conclusão dos projetos complementares principais. Isso impossibilita a elaboração de um orçamento detalhado com alta precisão. (KATO *et al.*, 2022)

Dessa limitação surge a necessidade de métodos de estimativa de custos que sejam suficientemente precisos mesmo com informações limitadas sobre o projeto. Um desses métodos é a análise paramétrica de custos, que assume que fatores que influenciaram custos de projetos semelhantes no passado continuarão a atuar de forma semelhante em novos projetos. Esse tipo de estimativa se baseia em estabelecer relações estatísticas entre determinadas características do projeto e seus custos. (NASA, 2015)

A literatura evidencia que a precisão das estimativas depende diretamente do nível de detalhamento do projeto e das informações disponíveis. Projetos incompletos ou insuficientemente especificados frequentemente geram estimativas menos precisas, com margens de erro que podem variar de $\pm$ 5% a $\pm$100% , dependendo do tipo e da metodologia empregada. Além disso, os métodos tradicionais tendem a ser insuficientes para captar a complexidade das variáveis envolvidas, especialmente em projetos que incluem componentes inovadores ou requisitos específicos. (INTERNATIONAL, 1997)

Nesse contexto, a integração de métodos estatísticos e algoritmos avançados de Apren-

dizado de Máquina ou *Machine Learning* (ML) representa um avanço significativo para gerenciar essas dificuldades. Essas abordagens são capazes de lidar com grandes volumes de dados, modelar interações não lineares e incorporar incertezas ao processo de estimativa. Além disso, metodologias baseadas em ML têm se mostrado especialmente eficazes na construção de modelos preditivos robustos, mesmo em contextos onde os dados disponíveis são limitados ou incompletos. (LIBRELOTTO *et al.*, 2003) (ANALYSTS, 2008)

1.2 Objetivo

Este trabalho se propõe a explorar essas ferramentas modernas para desenvolver uma solução prática, escalável e precisa para estimativas de custo de valores de referência de licitações, integrando técnicas paramétricas baseadas em ML ao contexto da construção civil. Além da elaboração das técnicas, também faz parte do escopo deste estudo analisar os métodos utilizados segundo os desempenhos resultantes da aplicação de cada um à base de dados disponibilizada pelo Centro de Estudos e Pesquisa de Engenharia (CEPE), da Força Aérea Brasileira (FAB).

Isso permitirá que a estimativa de custos seja realizada já na fase do programa de necessidades, antes mesmo da elaboração do projeto básico. Ao utilizar dados históricos fornecidos pelo CEPE em conjunto com técnicas avançadas de modelagem, a ferramenta proposta poderá fornecer previsões mais precisas e confiáveis, permitindo a realização de estimativas de custos já na etapa de programa de necessidades. Essa antecipação proporciona uma visão mais detalhada dos custos previstos antes da elaboração do projeto básico, auxiliando na tomada de decisões estratégicas desde os estágios iniciais do planejamento.

Outrossim, os algoritmos avançados de ML empregados neste estudo, como árvores de decisão e Redes Neurais Artificiais, são capazes de modelar interações complexas entre variáveis e lidar com dados incompletos ou dispersos, superando as limitações dos orçamentos expedidos tradicionais. (RODRIGUES, 2020)

Por fim, a melhoria na precisão das estimativas iniciais contribui para a redução de retrabalhos e aditivos contratuais. Essa abordagem minimiza a necessidade de ajustes posteriores e mitiga os impactos financeiros relacionados a revisões orçamentárias, promovendo maior eficiência e controle no desenvolvimento dos projetos.

1.3 Estrutura do trabalho

Este trabalho foi estruturado para explorar métodos de estimativa de custos em obras de engenharia civil, com foco especial nas demandas específicas da FAB.

A primeira seção, Introdução, contextualiza o problema e destaca a importância do estudo para a área de engenharia civil, justificando a adoção de técnicas avançadas para a estimativa de custos diante das particularidades e desafios que a FAB enfrenta na execução de suas obras.

Em seguida, a seção de Embasamento Teórico apresenta os principais conceitos que sustentam a estimativa de custos em obras civis, definindo o objeto de estudo e explorando os dispositivos legais aplicáveis, processo licitatório e os atributos essenciais para a formulação de orçamentos confiáveis. Nessa seção, também se revisa a literatura acadêmica sobre métodos de previsão de custos, enfatizando o uso de técnicas de ML.

Na Metodologia, são descritos os procedimentos adotados para coleta e tratamento de dados, bem como a aplicação dos modelos de ML, incluindo regressões lineares, modelos baseados em árvores de decisão e redes neurais artificiais. Esta etapa envolve uma análise rigorosa para selecionar as metodologias mais adequadas aos dados utilizados.

Posteriormente, nos Resultados, são apresentados e discutidos os resultados obtidos com cada técnica, incluindo uma análise comparativa do desempenho dos modelos, com base em métricas de precisão e erro, que busca interpretar os fatores mais impactantes nas estimativas e identificar as abordagens mais promissoras para o contexto das obras analisadas.

Finalmente, a seção de Conclusão sintetiza as contribuições principais do trabalho, ressaltando os avanços proporcionados pela ferramenta computacional proposta e suas potenciais aplicações na gestão de obras da FAB. São também discutidas as limitações encontradas ao longo do estudo e sugeridas direções para futuras pesquisas.

Nas últimas páginas do trabalho, apresenta-se o Apêndice com o código utilizado para o tratamento dos dados e aplicação dos métodos.

O autor utilizou a versão 4 do *ChatGPT* para identificar erros nas rotinas de código, aprimorar a gramática e melhorar a legibilidade geral dos textos. Após o uso dessas ferramentas de IA, o autor revisou e editou o conteúdo conforme necessário para manter o controle total autoral sobre a substância do trabalho.

2 Embasamento teórico

2.1 A Estimativa de Custo

A estimativa de custo pode ser definida como uma avaliação preliminar realizada com base em dados históricos, índices, estudos de ordem de grandeza, correlações ou comparações com projetos semelhantes, conforme destacado pelo Instituto Brasileiro de Auditoria de Obras Públicas ((IBRAOP), 2016). Como explica (BARROS, 2019), a estimativa conceitual deve ser preparada a partir de uma quantidade mínima de informações, havendo poucos ou nenhum detalhe nesta fase.

Uma das formas mais usuais de obter estimativas rápidas é por meio de indicadores de custo médio, como o custo por metro quadrado em edificações ou o custo por quilômetro em obras rodoviárias. Para projetos de tipologia e porte semelhantes, o uso de estudos comparativos ou parâmetros pré-definidos pode ser eficaz para determinar custos aproximados, sendo essa abordagem conhecida como método paramétrico de estimativa. (MAUÉS *et al.*, 2022)

O método paramétrico utiliza variáveis que refletem características físicas do empreendimento, como dimensões, tipologia e tecnologias empregadas. Dados históricos de projetos anteriores semelhantes são frequentemente usados para criar relações paramétricas de custo, que permitem a estimativa de custos globais ou parciais da obra. (OTERO, 1998)

Além de economizarem tempo em processos de licitação e planejamento, também ajudam a evitar erros e omissões comuns em estimativas tradicionais, mostrando-se uma ferramenta eficaz em cenários com informações técnicas limitadas. (DYSERT, 2001) (MEYER; BURNS, 1999)

No contexto dos níveis de precisão de estimativas de custos e orçamentos, (INTERNATIONAL, 1997) apresenta diretrizes amplamente utilizadas na construção civil, conforme ilustrado na Tabela 2.1.

Estabelecendo uma analogia com os níveis de maturidade dos projetos apresentados na Tabela 2.1, pode-se associar a classe 5 ao programa de necessidades, a classe 4 ao estudo

TABELA 2.1 – Níveis de precisão segundo níveis de maturidade do projeto. Adaptação de (INTERNATIONAL, 1997)

Classe	Nível de Maturidade do Projeto	Metodologia	Margens de Erro	Níveis de Precisão
5	0 a 2%	Estocástica ou julgamento	±20 a ±100%	0 a 80%
4	1 a 15%	Principalmente estocástica	±15 a ±60%	40 a 85%
3	10 a 40%	Mista, mas principalmente estocástica	±10 a ±30%	70 a 90%
2	30 a 75%	Principalmente determinística	±5 a ±15%	85 a 95%
1	65 a 100%	Determinística	±5%	95%

de viabilidade técnico-econômica, a classe 3 ao anteprojeto, a classe 2 ao projeto básico e a classe 1 ao projeto executivo. A prática recomendada enfatiza o uso de metodologias estocásticas para as estimativas de custos das classes 4 e 5. Essas estimativas apresentam níveis de precisão que variam entre 40% e 85% para a classe 4 e de 0% a 80% para a classe 5. (BELTRÃO *et al.*, 2022)

De forma resumida, metodologias estocásticas baseiam-se na modelagem de relações entre custos e características do objeto analisado utilizando dados históricos de projetos similares realizados anteriormente. Entre os métodos estocásticos disponíveis, destaca-se, neste estudo, o uso de modelos de regressão, para estimar os valores de projetos licitados pela FAB.

2.2 As Licitações no Âmbito da FAB

A licitação de obras públicas no Brasil é o processo administrativo obrigatório para a contratação de serviços e execução de projetos pela Administração Pública. Regida pela Lei nº 14.133/2021, tem como objetivos assegurar a seleção da proposta mais vantajosa, promover a isonomia, eficiência, transparência e desenvolvimento sustentável. (BRASIL, 2021)

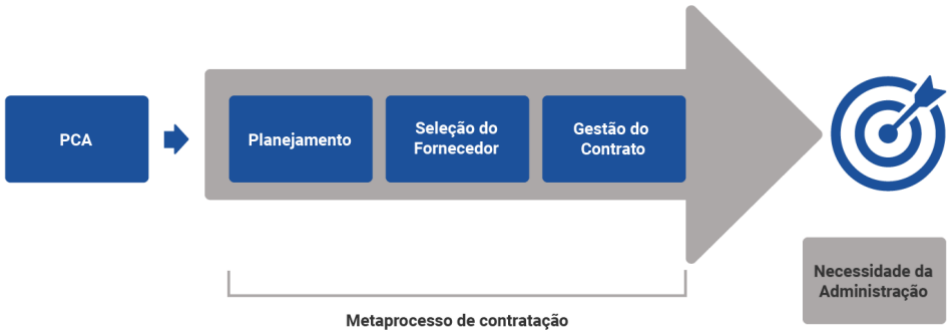

FIGURA 2.1 – Fluxograma das etapas de uma contratação. ((TCU), 2024)

O Plano de Contratações Anual (PCA) organiza e prioriza as contratações planejadas para o exercício subsequente, integrando o planejamento estratégico da organização

pública (Figura 2.1). ((TCU), 2024)

Seus principais elementos são:

- **Identificação de Necessidades:** Levantamento das demandas pelas unidades organizacionais.
- **Estimativa de Custos Preliminares:** Baseada em pesquisas de mercado e tabelas oficiais.
- **Justificativas:** Fundamentação considerando impacto social, econômico e ambiental.

No contexto da FAB, o PCA é parte do ciclo de planejamento institucional, conectando demandas operacionais às diretrizes estratégicas. A abordagem sistemática é especialmente relevante na FAB, na qual as restrições orçamentárias e a natureza estratégica das obras exigem rigor na previsão de custos e prazos.

Nesse fluxo licitatório, a estimativa de custos baseada em métodos de regressão desempenha um papel crítico, estando presente na fase inicial de planejamento da contratação e durante a elaboração dos estudos técnicos preliminares. Uma estimativa de custos precisa e fundamentada é essencial por várias razões. Primeiro, orienta a elaboração do orçamento e estabelece parâmetros para a análise das propostas, influenciando diretamente a viabilidade econômica do projeto. Segundo, contribui para a prevenção de sobrepreços e superfaturamentos, garantindo a integridade do processo licitatório e o uso eficiente dos recursos públicos. Terceiro, auxilia na tomada de decisão, permitindo uma avaliação realista da capacidade financeira da instituição para realizar a obra ou serviço.

A legislação brasileira reforça ainda a importância da estimativa de custos ao estabelecer que a administração pública deve adotar práticas e metodologias que garantam a precisão das estimativas, podendo utilizar sistemas de custos de referência, pesquisas de mercado, custos de obras similares e outras técnicas reconhecidas. (BRASIL, 2021)

2.3 Propriedades do Orçamento

O Manual para a elaboração de Planilhas Orçamentárias destaca que a construção de estimativas de custos envolve diversas propriedades e atributos que devem ser considerados para a análise da precisão e confiabilidade dos orçamentos. ((TCU), 2014)

2.3.1 Especificidade

Os custos de serviços e obras de engenharia são influenciados por fatores específicos de cada projeto, como o tipo de construção, métodos construtivos, materiais utilizados,

localização geográfica, condições climáticas e características do terreno. Essas particularidades tornam essencial que a estimativa reflita com precisão as características únicas do projeto.

2.3.2 Temporalidade

Os preços de insumos, mão de obra e serviços sofrem variações ao longo do tempo devido a fatores como inflação, oscilações econômicas, alterações nos custos de produção e mudanças nas políticas fiscais e tributárias. Por isso, é fundamental que as estimativas sejam baseadas em uma data de referência para os preços e incluam mecanismos de atualização que garantam sua precisão temporal.

2.3.3 Aproximação

Apesar de todos os esforços para alcançar a maior precisão possível, as estimativas de custo sempre representam uma aproximação da realidade. Eventos imprevistos, como mudanças no projeto, atrasos ou condições climáticas adversas, podem impactar os custos reais da obra. Dessa forma, é importante incorporar margens de contingência nas estimativas e realizar análises de sensibilidade que permitam avaliar o impacto de possíveis variações nos custos previstos.

2.3.4 Curva ABC de Serviços e Insumos

A Curva ABC (Figura 2.2) é uma ferramenta de gestão que classifica os itens de um orçamento de acordo com sua relevância financeira. Itens ”A” representam uma pequena quantidade de itens com alto valor agregado, ”B” correspondem a itens com importância intermediária, e ”C” são muitos itens com baixo valor individual. Essa classificação auxilia na priorização do controle e gestão dos recursos, se mostrando um instrumento importante para o esclarecimento da relevância de um determinado item no projeto. ((TCU), 2014)

Tipo	Descrição do Insumo	Unid.	Quantidade	Preço Parcial	%	% Acumulado
Equipamento	Caminhão Basculante : Mercedes Benz : 2423 K : 10 m3 - 15 t	H	177.486,08	23.758.464,36	43,87	43,87
Material	Cimento Asfáltico de Petróleo	Ton.	4.320,00	5.616.000,00	10,37	54,24
Mão de Obra	Servente	H	219.575,72	2.458.977,67	4,54	58,78
Material	Brita 3	M^3	12.247,08	1.751.332,42	3,23	62,01
Material	Brita 2	M^3	12.243,41	1.750.807,12	3,23	65,24
Material	Brita 1	M^3	12.243,41	1.750.807,12	3,23	68,47
Material	Dente de corte (W6/22) p/ recicladora	Unid.	22.321,15	1.427.660,88	2,64	71,11
Material	Asfalto diluído - CM-30	Ton.	598,38	1.244.630,40	2,30	73,41
Equipamento	Caminhão Basculante : Volvo BM : FM 12 6X4 : 20 t	H	4.308,13	976.370,03	1,80	75,21
Equipamento	Recicladora de Pavimento : Wirtgen : WR 2000 : a frio	H	1.701,31	918.966,90	1,70	76,91
Material	Óleo combustível 1A	Litro	576.000,00	913.536,00	1,69	78,60

FIGURA 2.2 – Exemplo de Curva ABC de insumos ((TCU), 2014).

2.4 Literatura Acadêmica

Diversos métodos têm sido desenvolvidos para aprimorar a precisão das estimativas, especialmente em estágios iniciais, nas quais informações detalhadas são limitadas. Nesta revisão, serão mostrados estudos que destacam a importância de estimativas precisas. Os trabalhos incluem modelos de regressão linear, redes neurais artificiais e outras abordagens inovadoras.

(GÜNAYDIN; DOĞAN, 2004) reforçam que o custo é um fator determinante na tomada de decisões iniciais em projetos de engenharia civil. Em um mercado global competitivo, a definição precisa de custos é essencial para estabelecer margens de lucro e assegurar a qualidade planejada. A regressão linear surge como uma ferramenta estatística robusta para modelar custos com base em variáveis acessíveis, sendo particularmente útil quando o detalhamento do projeto ainda é limitado.

(ANDRADE; SOUZA, 2003) enfatizam a importância da precisão nas estimativas iniciais, demonstrando que a regressão linear pode ser aplicada de forma eficaz em estruturas repetitivas, como tanques de armazenamento. (KATO *et al.*, 2022) reforçam essa aplicabilidade ao adaptarem a metodologia para reservatórios de água, evidenciando a versatilidade do método em diferentes contextos.

(LOWE *et al.*, 2006) compararam redes neurais a métodos tradicionais, concluindo que as redes neurais oferecem maior acurácia quando treinadas com dados robustos e relevantes. Ao lidar com múltiplas variáveis de forma interdependente, as redes neurais se ajustam automaticamente a padrões complexos presentes em bases de dados amplas, superando as limitações da regressão linear.

(HEGAZY; AYED, 1998) aplicaram redes neurais em uma base de dados de projetos

rodoviários, identificando os principais fatores de custo. Ao empregar o método de otimização simplex e o algoritmo genético, obtiveram resultados mais precisos, evidenciando uma redução significativa no erro de previsão. Esse estudo se tornou referência na aplicação de redes neurais em projetos de infraestrutura.

(SONMEZ, 2008) combinaram a técnica de regressão com o método bootstrap, obtendo resultados positivos ao aplicar essa combinação em 20 projetos de construção na Turquia. Os métodos híbridos oferecem maior flexibilidade e precisão em amostras menores, evidenciando seu potencial em diferentes contextos.

(KIM *et al.*, 2004) avaliaram a eficácia de diferentes técnicas, incluindo redes neurais, regressão múltipla e o Case-Based Reasoning (CBR), para estimar custos de projetos residenciais. As redes neurais demonstraram maior precisão, embora exigissem mais tempo para aquisição de dados e modelagem. Esse estudo evidencia a necessidade de avaliar o custo-benefício e o tempo disponível para decidir a técnica mais apropriada para cada projeto.

(BELTRÃO *et al.*, 2022) desenvolveram modelos de estimativa de custos utilizando regressão linear aplicada a obras penitenciárias. O estudo, baseado em dados de 27 projetos, demonstrou que a utilização de modelos estatísticos pode fornecer estimativas confiáveis mesmo em setores específicos da construção civil.

(RODRIGUES, 2020) explorou o uso de redes neurais artificiais para estimar custos de obras, utilizando informações de 49 projetos da Comissão de Obras do Departamento de Ciência e Tecnologia da Aeronáutica. Os resultados indicaram que as redes neurais podem capturar padrões complexos nos dados, proporcionando estimativas mais precisas em comparação com métodos tradicionais.

Esses trabalhos evidenciam a importância de utilizar técnicas avançadas de análise de dados e modelagem estatística para aprimorar as estimativas de custos, reduzindo incertezas e contribuindo para a tomada de decisões mais embasadas. A integração de técnicas estatísticas, inteligência artificial e simulações estocásticas proporciona ferramentas poderosas para engenheiros e gestores, permitindo estimativas mais confiáveis e redução de riscos financeiros.

2.5 Aprendizado de Máquina no Escopo da Estimativa de Custo

O uso de ML na estimativa de custo tem sido amplamente explorado como uma alternativa mais robusta e precisa em relação aos métodos tradicionais, como o custo unitário básico (CUB) e as análises lineares simplificadas. Estudos como os de (TIBSHIRANI, 1996)

com o Lasso e de (FRIEDMAN, 2001) com o Gradient Boosting destacam a importância de incorporar regularização e técnicas de otimização iterativa para capturar relações complexas em dados de engenharia civil.

Entre essas técnicas, a Regressão Linear permanece como um ponto de partida fundamental, estabelecendo uma relação direta entre as variáveis independentes e o custo. (HASTIE *et al.*, 2009)

No contexto deste trabalho, serão explorados os seguintes métodos de aprendizado de máquina:

- Regressão Linear
- Regressão Ridge
- Regressão Lasso
- Elastic Net
- Multilayer Perceptron (MLP)
- Support Vector Regressor (SVR)
- Random Forest
- XGBoost
- Gradient Boost
- CatBoost

A Regressão Ridge introduz um mecanismo de regularização para lidar com problemas de multicolinearidade, penalizando os coeficientes e evitando ajustes excessivos aos dados. A Regressão Lasso, por sua vez, além da regularização, realiza a seleção de variáveis, eliminando aquelas com menor influência no modelo. O Elastic Net combina as propriedades das duas regressões anteriores, oferecendo um balanceamento entre penalização e seleção de características. (TIBSHIRANI, 1996)

As MLPs representam um avanço significativo em modelagem não linear, sendo capazes de capturar interações complexas que métodos lineares não conseguem modelar. O trabalho de (GÉRON, 2019) detalha como essas redes podem ser configuradas para lidar com dados ruidosos e incompletos, comuns em projetos de engenharia civil.

Métodos baseados em árvores de decisão, como o Random Forest, utilizam múltiplas árvores para melhorar a precisão e reduzir o *overfitting*. Já os algoritmos de boosting, como

o XGBoost, Gradient Boost e CatBoost, aprimoram iterativamente o modelo, focando nos erros das previsões anteriores para aumentar a acurácia. (CHEN; GUESTRIN, 2016)

O SVR é outra técnica poderosa, especialmente útil para conjuntos de dados de alta dimensionalidade, que busca encontrar hiperplanos ótimos que separem as classes ou valores de interesse. Cada um desses métodos oferece vantagens específicas na modelagem de dados complexos e na melhoria das estimativas de custos. (HASTIE *et al.*, 2009)

A integração dessas técnicas no escopo da engenharia civil possibilita a captura de interações sutis entre variáveis de projeto, reduzindo riscos financeiros associados a subestimações ou superestimações nos custos. Estudos como os de (ADAMS, 2006) e (FONSECA, 2013) destacam como a análise probabilística combinada com aprendizado de máquina pode melhorar a previsibilidade e a confiabilidade das estimativas.

A análise dessas técnicas buscará destacar suas aplicações práticas no escopo da estimativa de custos, sem, por ora, aprofundar-se em formulações matemáticas complexas, as quais serão pinceladas a seguir. A compreensão dessas metodologias é essencial para avançar na implementação de modelos preditivos eficazes e alinhados com as necessidades atuais da engenharia civil.

3 Metodologia

3.1 Primeiros Passos

3.1.1 Coleta dos Dados

A coleta de dados envolveu duas abordagens principais: a utilização de um formulário Google para obtenção de informações padronizadas e a disponibilização de arquivos complementares pelo CEPE. O formulário foi distribuído com o objetivo de compilar informações detalhadas sobre obras realizadas pela FAB, especificamente em Organizações Militares (OMs), enquanto o CEPE disponibilizou arquivos de 85 projetos adicionais, contendo documentos como arquivos PDF (*Portable Document Format*) e CAD (*Computer-Aided Design*) de plantas, planilhas de orçamento e outros materiais relevantes sobre as obras.

O formulário foi estruturado com campos específicos que capturavam dados sobre as obras, tais como:

- **Identificação do Responsável**: Campos para o preenchimento do posto seguido do nome de guerra e o e-mail institucional (@fab.mil.br) do responsável pela submissão dos dados.
- **Informações sobre a OM e Localização da Obra**: A OM responsável pela execução da obra e a Unidade Federativa (UF) em que a obra foi realizada foram solicitados.
- **Descrição do Projeto**: Foram requisitadas informações como o título do projeto, tipo de obra ou serviço de engenharia (ex.: reforma, adequação, ampliação), e a tipologia principal da benfeitoria (ex.: telhado, cobertura, prédio administrativo).
- **Dados Quantitativos e Financeiros**: Os responsáveis foram orientados a preencher o quantitativo total do projeto, utilizando a unidade de medida específica (ex.: metros quadrados ou cúbicos), e o custo total, incluindo elementos como BDI (Benefícios e Despesas Indiretas) e canteiro de obras.

- **Data de Referência do Orçamento**: A data de referência do orçamento foi indicada como o primeiro dia do mês em que os valores orçamentários foram considerados, seguindo parâmetros específicos de obras, por exemplo os fornecidos pelo Sistema Nacional de Pesquisa de Custos e Índices da Construção Civil (SINAPI), para edificações, e pelo Sistema de Custos Referenciais de Obras (SICRO) para infraestrutura. Essa data permite a padronização dos custos informados, diferenciando-se da data da coleta, que é registrada automaticamente como *timestamp* no momento do envio do formulário.

- **Escopo e Observações**: Um campo para o breve descritivo do escopo permitiu aos respondentes detalhar as principais ações e objetivos da obra, enquanto um campo para observações foi reservado para comentários adicionais, facilitando uma compreensão mais abrangente do contexto de cada projeto.

A pesquisa abrangeu o período de 26 de março de 2024 a 9 de setembro de 2024, durante o qual o formulário foi distribuído e as respostas foram coletadas de diferentes OMs. Esse intervalo de tempo permitiu uma coleta ampla e detalhada, garantindo a captação de dados atualizados e específicos sobre o panorama das obras realizadas pela Aeronáutica. Deve-se considerar a possibilidade de erros em alguns *inputs* de dados ou mesmo inconsistências nos dados disponibilizados e inseridos, que podem impactar a interpretação e análise dos resultados.

As obras incluídas na pesquisa possuem datas de referência de orçamento que variam de janeiro de 2015 a junho de 2024. Esse período contém projetos planejados ao longo de quase uma década, oferecendo uma visão histórica das obras realizadas pela Aeronáutica nesse intervalo de tempo.

3.1.2 Parametrização

Os direcionadores de custo, nesse caso, são definidos como variáveis independentes que têm forte influência sobre a variável dependente, geralmente o custo. A estimativa é construída ao combinar os valores históricos dessas variáveis em um plano cartesiano, permitindo que, por meio de análises estatísticas, seja obtida uma equação matemática que represente a relação entre elas. (NASA, 2015)

Para que a transferência dos dados das obras pudesse ser feita de maneira eficiente, buscou-se definir esses direcionadores com intuito de torná-la menos complexa possível. Essa padronização mitiga possíveis erros humanos na inserção das informações no formulário.

Diferentes tipos de informações sobre um projeto afetam a precisão das estimativas de

maneiras distintas. Quanto mais abrangentes forem os dados disponíveis, maior será sua contribuição para a qualidade da estimativa de custos. (KATO *et al.*, 2022)

Com isso em mente, a escolha das variáveis categóricas para análise foi realizada atribuindo a cada obra características intrínsecas a elas, de tal forma que fosse possível caracterizar benfeitorias futuras com base em variáveis já consagradas e praticamente imutáveis, como a unidade federativa, o tipo da obra e a tipologia. Isto significa que, para a predição fazer sentido, uma unidade declarada como prédio residencial, por exemplo, será sempre tratada como um prédio residencial.

Também foi de interesse atribuir a cada obra a unidade de referência utilizada para o quantitativo. Assim é possível analisar o comportamento dos modelos diante de obras que, apesar de possuírem tipologias distintas, são mensuradas por uma mesma quantidade. Além disso, é esperado que essa quantidade pertença à categoria "A" da Curva ABC do orçamento referente ao projeto em questão, devido à relevância financeira do quantitativo. A Tabela 3.1 abaixo detalha as tipologias e as medidas associadas a cada uma delas.

TABELA 3.1 – Unidades de Referência por Tipologia

Tipologia	**Unidade de Referência**
Prédio Administrativo (escritório, auditório, sala de aula, biblioteca, prédio do comando, corpo da guarda, capela)	m^2 de área construída, considerando todos os pavimentos da edificação
Prédio Operacional ou Técnico (oficina, laboratório, seção contra incêndio de aeródromo, seção de material bélico)	m^2 de área construída, considerando todos os pavimentos da edificação
Hangar de Aeronaves	m^2 de projeção da cobertura do hangar
Cobertura sem Fechamento Lateral (hangaretes, área de lavagem de aeronaves, garagem de veículos)	m^2 de projeção da cobertura
Galpão com Fechamento Lateral (terminal de passageiro ou carga, galpão logístico, depósito, ginásio)	m^2 de projeção da cobertura do galpão
Paiol	m^2 de área de projeção
Prédio Hospitalar (hospital, clínicas, grupamento de saúde, laboratório, farmácia)	m^2 de área construída, considerando todos os pavimentos da edificação
Rancho (cozinha industrial, despensa, câmaras frias, refeitório)	m^2 de área construída

Continua na próxima página...

Tipologia	**Unidade de Referência**
Posto de Abastecimento de Veículos	m^3 de combustível armazenado
Prédio Residencial (próprio nacional residencial, hotel)	m^2 de área construída, considerando todos os pavimentos da edificação
Vestiário e Alojamento	m^2 de área construída
Muros e Cercas	m linear de muro ou cerca
Rede Elétrica de Baixa Tensão	m linear de rede
Rede Elétrica de Média Tensão	m linear de rede
Subestação Elétrica	kVA
Usina Fotovoltaica	kWp
Rede de Abastecimento de Água	m linear de rede
Rede de Coleta de Esgoto ou Drenagem	m linear de rede
Estação de Tratamento de Água ou Esgoto	m^3/dia de volume tratado
Reservatório Enterrado (cisterna)	m^3 de volume armazenado
Reservatório Elevado (castelo d'água)	m^3 de volume armazenado
Pavimento Flexível (asfáltico)	m^2 de asfalto
Pavimento Rígido (concreto)	m^2 de concreto
Iluminação de Pátio de Aeronaves	m linear da linha de postes de iluminação
Muro de Contenção	m^2 de área de frente do muro
Sinalização Horizontal	m^2 de área de pátio e/ou pista
Toldo ou Cobertura	m^2 de área de projeção
Portão de Hangar ou Galpão	m^2 de área de portão

Em conjunto com essas categorizações, a variável numérica escolhida para refletir as especifidades das obras foi o quantitativo associado. Preteriu-se o uso do custo total como *output*, pois isso evita que obras de custo total muito elevado engendrem viés na predição, dada a variância atrelada a esse indicador.

A parametrização resultante, portanto, possui as seguintes variáveis:

- **Unidade Federativa:** unidade federativa onde a obra foi executada.
- **Tipo da Obra:** reforma ou de construção.
- **Tipologia da Benfeitoria:** classificação da obra quanto à finalidade.
- **Unidade de Referência:** medida na qual se calcula o quantitativo.
- **Quantitativo Total:** quantificação da obra, na unidade de referência correspondente.

3.1.3 Descrição da Base de Dados

A base de dados analisada contém um conjunto de 204 registros que representam diversos projetos de obras, distribuídos entre diferentes tipos de benfeitorias e regiões do Brasil. As informações foram organizadas em oito colunas que detalham o projeto, a localização, o tipo e a natureza da obra, além de dados quantitativos e financeiros associados a cada projeto.

Os projetos são identificados pela coluna `Titulo_Projeto`, que traz o nome específico de cada empreendimento, compondo uma base de dados abrangente e diversificada. Esses projetos estão distribuídos por 17 unidades federativas, sendo o estado do Rio de Janeiro (RJ) o mais representado, com 41 registros.

	Titulo_Projeto	UF	Tipo_Obra	Tipologia_Benfeitoria	Quantitativo_Total	Custo_Unitario	Unidade_Referencia
0	Reforma do telhado, do Sistema de Proteção Con...	SP	Reforma	TELHADO OU COBERTURA	6173.85	339.79	R$ / m² de área de projeção
1	Adequação das instalações do CELOG	SP	Reforma	PRÉDIO ADMINISTRATIVO	52.65	1876.43	R$ / m² de área construída, considerando todos...
2	Adaptações na nova sede do Destacamento de Eng...	SP	Reforma	PRÉDIO ADMINISTRATIVO	210.00	1024.50	R$ / m² de área construída, considerando todos...
3	Reparo da Cobertura nas Áreas de Ambulatórios ...	SP	Reforma	TELHADO OU COBERTURA	4600.00	187.21	R$ / m² de área de projeção
4	Adequação das Instalações da Subdivisão de Apo...	SP	Reforma	PRÉDIO ADMINISTRATIVO	88.00	1128.23	R$ / m² de área construída, considerando todos...
...	...	...	...	...	...	...	...
199	RECUPERAÇÃO EMERGENCIAL DA TWY C, TRECHO ENTRE...	RS	Reforma	PAVIMENTO FLEXÍVEL	581.40	2291.95	R$ / m² de asfalto
200	IMPLANTAÇÃO DO SISTEMA DE ILUMINAÇÃO DE PÁTIOS...	RS	Construção	ILUMINAÇÃO DE PÁTIO DE AERONAVES	1318.00	5976.68	R$ / m linear da linha de postes de iluminação
201	PROJETO DE INSTALAÇÃO DE CERCA OPERACIONAL NO ...	RS	Construção	MUROS E CERCAS	6000.00	1146.36	R$ / m linear de muro ou cerca
202	ADEQUAÇÃO DA ENTRADA DE ENERGIA DA SEÇÃO DE SU...	RS	Reforma	REDE ELÉTRICA DE MÉDIA TENSÃO	30.00	7755.20	R$ / m linear de rede
203	DCTA Hangar PAMA SJ	SP	Construção	HANGAR DE AERONAVES	10200.00	5556.54	R$ / m² de projeção da cobertura do hangar

204 rows × 7 columns

FIGURA 3.1 – Primeiras linhas da base de dados utilizada no estudo.

A coluna `Tipo_Obra` indica a natureza das obras, distinguindo-as entre *Reforma* e *Construção*.

Na tipologia das benfeitorias, representada pela coluna `Tipologia_Benfeitoria`, há 28 categorias diferentes. A categoria *Prédio Administrativo* destaca-se como a mais frequente, presente em 33 registros.

3.1.4 Análise Quantitativa e Financeira

A análise quantitativa e financeira revela uma grande variação nos valores registrados. A coluna `Quantitativo_Total`, que se refere ao total de unidades da métrica utilizada em cada projeto (como metros quadrados ou quantidade de itens), apresenta uma média de $6.499,08$ unidades, com uma alta dispersão dos valores, evidenciada por um desvio padrão de $31.863,22$. O valor mínimo registrado é de $4,16$ unidades, enquanto o máximo atinge 349.000.

Em relação ao custo, a coluna `Custo_Total_Atualizado` traz o valor atualizado do investimento de cada projeto, isto é, o montante ajustado para refletir a variação dos custos na construção civil ao longo do tempo. Esse índice, calculado mensalmente pela Fundação Getúlio Vargas (FGV), mede a inflação específica no setor de construção e é

amplamente utilizado para atualizar valores de contratos e orçamentos relacionados a obras e serviços de engenharia civil.

A média dos custos totais atualizados é de R$ 7.043.004, com um desvio padrão considerável de R$ 13.912.640. Esse elevado desvio reflete a natureza desigual dos custos dos projetos, que variam desde R$ 9.278, 54 até valores expressivos de R$ 92.958.520.

A coluna `Custo_Unitario` representa o custo por unidade de referência, fornecendo um indicador detalhado do investimento relativo à métrica empregada. O custo unitário médio é de R$ 4.924, 43, com uma alta variação entre os registros, apresentando um desvio padrão de R$9.624,90, e os valores variam de R$ 1, 62 até R$ 114.904, 09.

TABELA 3.2 – Exemplo de parâmetros estatísticos para a coluna de `Custo_Unitario`.

Estatísticas de Quantis		**Estatísticas Descritivas**	
Mínimo	1.62	Desvio padrão	9624.9036
5º percentil	204.1145	CV*	1.9545232
Q1	1159.41	Curtose	86.003914
Mediana	2903.535	Média	4924.4255
Q3	5828.4575	MAD**	1909.18
95º percentil	13817.553	Assimetria	8.151433
Máximo	114904.09	Soma	1004582.8
Amplitude	114902.47	Variância	92638770
IQR***	4669.0475	Monotonicidade	Não monotônico

* CV: Coeficiente de Variação.

** MAD: Desvio Absoluto Mediano (*Median Absolute Deviation*).

*** IQR: Intervalo Interquartil (*Interquartile Range*).

A coluna final, `Unidade_Referencia`, descreve as unidades utilizadas para o cálculo do custo de cada projeto, como metros quadrados de área construída ou de área de projeção. Ao todo, há 20 variações nas unidades de referência, sendo a mais comum *R$ / m^2 de área construída*, presente em 73 registros.

3.2 Tratamento dos Dados

Em concordância com as orientações de (GÉRON, 2019), o tratamento de dados feito neste trabalho envolveu etapas como limpeza e transformação, análise exploratória, remoção de *outliers*, transformação de variáveis, divisão dos dados para validação e seleção e avaliação de modelos preditivos. O trecho de código contendo o tratamento mencionado é exibido na listagem *Listing* 3.1 e 3.2 abaixo. O código completo pode ser consultado ao

final do estudo, no Anexo A.

Identificaram-se *outliers* na variável alvo `Custo_Unitario`. Logo, para reduzir seu impacto, foram removidos valores acima do percentil de 95%, filtrando os dados com `Custo_Unitario` até esse limite. O mesmo foi feito para a variável `Quantitativo_Total`, mas para limite de 99,8%, por se tratarem de dados diretamente relacionados ao porte da obra. Isso é importante para minimizar a influência de extremos sem comprometer a representatividade de projetos excepcionais na amostra.

Durante o desenvolvimento deste trabalho, utilizou-se a linguagem Python por meio do ambiente *Google Colab* para realizar o tratamento dos dados e a construção dos modelos preditivos. O processo de tratamento foi minuciosamente conduzido para assegurar a qualidade e a integridade dos dados utilizados nas análises subsequentes. O resultado foi a construção de modelos robustos e a obtenção de insights relevantes sobre os fatores que impactam o custo unitário nas obras analisadas.

Listing 3.1 – Preprocessing and Outlier Removal for Housing Data

```
# Copia do DataFrame Original
df = housing.copy()

# Transformar colunas categoricas em variaveis dummy
df = pd.get_dummies(df, columns=['UF', 'Tipo_Obra', 'Tipologia_Benfeitoria', '
    Unidade_Referencia'], drop_first=True)

# Calcular limites
upper_limit_custo = df['Custo_Unitario'].quantile(0.95)
upper_limit_quantitativo = df['Quantitativo_Total'].quantile(0.998)

# Identificar e combinar outliers
outliers_custo = df[df['Custo_Unitario'] > upper_limit_custo]
outliers_quantitativo = df[df['Quantitativo_Total'] > upper_limit_quantitativo]
outliers_total = pd.concat([outliers_custo]).drop_duplicates()

# Remover os outliers do DataFrame
outlier_indices = outliers_total.index
df_cleaned = df.drop(index=outlier_indices)

# Separar variaveis dependentes e independentes usando o DataFrame limpo
X = df_cleaned.drop(columns=['Custo_Unitario', 'Titulo_Projeto', '
    Custo_Total_Atualizado'])
y = df_cleaned['Custo_Unitario']

# Dividir os dados em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state
    =42)
```

Listing 3.2 – Preprocessing and Outlier Removal for Housing Data

```
# Identificar colunas numericas e categoricas
num_cols = X_train.select_dtypes(include=['float64', 'int']).columns.tolist()
cat_cols = X_train.select_dtypes(include=['object']).columns.tolist()
colunas_train = X_train.columns
index_train = X_train.index
colunas_test = X_test.columns
index_test = X_test.index

# Transformacao de normalizacao das variaveis numericas
scaler = StandardScaler()
```

Inicialmente, importaram-se as bibliotecas essenciais para a manipulação e análise de dados, como `NumPy`, `Pandas` e `Matplotlib`.

A tabela consolidada com as informações das obras foi importada para o *Google Drive* no formato *.csv*, o qual foi acessado diretamente no ambiente *Colab*.

Durante o carregamento dos dados, detectou-se que algumas colunas numéricas eram interpretadas como *strings* devido ao uso de vírgulas como separador decimal. Para resolver isso, substituíram-se as vírgulas por pontos e converteu-se as colunas para o tipo *float*, assegurando a precisão em operações matemáticas subsequentes. Ademais, a coluna `Custo_Total_Atualizado` foi removida por ser desnecessária ou redundante para a análise.

Para entender melhor a estrutura e características dos dados, foi gerado um relatório exploratório com a biblioteca `ydata-profiling`, revelando a distribuição das variáveis, valores ausentes, correlações e possíveis *outliers*.

Diante da presença de variáveis categóricas, como `UF`, `Tipo_Obra`, `Tipologia_Benfeitoria` e `Unidade_Referencia`, foi necessário transformá-las em variáveis numéricas para inclusão nos modelos de regressão. Utilizou-se a técnica de *one-hot encoding* com a função `pd.get_dummies()`, que converte categorias em colunas binárias.

Após o pré-processamento, os dados foram divididos em 85% para treinamento e 15% para teste com a função `train_test_split` do `scikit-learn`.

Na modelagem, treinou-se inicialmente um modelo de Regressão Linear com a variável alvo `Custo_Unitario`. O desempenho foi avaliado com o Erro Quadrático Médio ou *Mean Squared Error* (MSE) e o Coeficiente de Determinação (R^2). Um gráfico de dispersão entre valores reais e previstos foi gerado para visualizar a qualidade do ajuste e identificar possíveis tendências.

Para melhorar o desempenho dos modelos, testaram-se transformações Raiz Quadrada e Yeo-Johnson na variável alvo visando normalizar a distribuição e reduzir a influência de

assimetrias ou *outliers*, assim reduzindo potenciais vieses. (DOCUMENTATION, 2024)

Os modelos foram re-treinados e as previsões revertidas à escala original para avaliação. Em seguida, explorou-se os algoritmos de regressão supracitados para identificar qual modelo apresentava melhor desempenho com os dados disponíveis. Cada modelo foi treinado e avaliado com e sem transformações na variável alvo, comparando a eficácia de cada abordagem com métricas consistentes (MSE e R^2). Gráficos de dispersão também foram gerados para visualizar o desempenho e identificar padrões ou discrepâncias.

Com o modelo Random Forest treinado, foram feitas previsões no conjunto de teste, e os valores reais e previstos de `Custo_Unitario` foram adicionados ao *DataFrame* original para facilitar a análise comparativa. Calculou-se a diferença absoluta entre os valores reais e previstos para identificar observações com maiores discrepâncias.

Por fim, realizou-se uma análise de importância das variáveis com o modelo Random Forest. As importâncias das *features* foram extraídas, destacando as dez variáveis mais relevantes. Um gráfico de barras foi gerado para visualizar as importâncias, oferecendo insights sobre os fatores que mais influenciam o `Custo_Unitario`.

3.3 Aplicação dos Métodos

A parametrização pode ser realizada de diversas formas, com a execução de uma simulação em que as características da obra servem como *inputs* e o valor como *output*. Neste trabalho, utilizou-se a Regressão Linear Simples para conduzir a simulação.

Para ampliar a análise, foram aplicados os modelos supervisionados de ML listados abaixo. Nessa abordagem, os algoritmos são treinados utilizando um conjunto de dados que já possui rótulos conhecidos, ou seja, cada exemplo de entrada está associado a um resultado esperado. O objetivo é ensinar os modelos a estabelecer uma relação entre os dados de entrada e suas respectivas saídas, permitindo que eles façam previsões precisas para novos dados. Essas técnicas são amplamente utilizadas para regressão (como prever o valor de imóveis com base em suas características). (MURPHY, 2012)

A escolha dos hiperparâmetros baseou-se nos valores padrão das bibliotecas utilizadas. Esses parâmetros foram definidos pelos desenvolvedores após extensivos testes em diferentes cenários e conjuntos de dados, garantindo um desempenho sólido em uma ampla variedade de aplicações. Segundo a documentação da biblioteca *scikit-learn*, os valores padrão são cuidadosamente ajustados para fornecer um equilíbrio entre desempenho e simplicidade, servindo como um ponto de partida confiável para experimentos iniciais (SCIKIT-LEARN TEAM, 2021). Essa abordagem também é destacada por Brownlee (2023), que menciona que utilizar configurações *default* pode simplificar o processo de modelagem

inicial e reduzir a complexidade de ajustes prematuros (BROWNLEE, 2023). Além disso, ferramentas complementares como `NumPy` e `Pandas` desempenham um papel importante no gerenciamento eficiente de dados e na aplicação prática dos algoritmos (NumPy Community, 2024; Pandas Development Team, 2024), permitindo que as configurações padrão sejam exploradas com eficiência e flexibilidade.

Parâmetros como `alpha=1.0` na Ridge Regression e `n_estimators=100` no Random Forest oferecem um bom compromisso entre precisão e eficiência computacional. Essa abordagem permite concentrar-se inicialmente na compreensão dos dados e na seleção do modelo adequado, sabendo que os hiperparâmetros escolhidos fornecerão resultados confiáveis. Posteriormente, caso seja necessário, ajustes finos podem ser realizados para otimizar ainda mais o desempenho com base em métricas de validação.

- **Ridge Regression**: `Ridge(alpha=1.0)`
- **Lasso Regression**: `Lasso(alpha=0.1)`
- **ElasticNet**: `ElasticNet(alpha=0.1, l1_ratio=0.5)`
- **MLP Regressor**: `MLPRegressor` com:
  - Camadas ocultas: (10, 5) para evitar *overfitting*
  - Função de ativação: `tanh`
  - Otimizador: `adam`
  - Regularização L2 (`alpha=0.01`) para evitar *overfitting*
  - Taxa de aprendizado adaptativa (`learning_rate='adaptive'`)
  - Máximo de 500 iterações com parada antecipada (`early_stopping=True`)
  - `random_state=42`
- **Support Vector Regressor**: `SVR(kernel='rbf', C=1.0, epsilon=0.1)`
- **Random Forest**: `RandomForestRegressor(n_estimators=100, random_state=42)`
- **XGBoost Regressor**: `XGBRegressor(n_estimators=100, random_state=42)`
- **Gradient Boosting**: `GradientBoostingRegressor(n_estimators=100, random_state=42)`
- **CatBoost Regressor**: `CatBoostRegressor` com:
  - 1.000 iterações para ajuste fino
  - Taxa de aprendizado reduzida (`learning_rate=0.01`)

- Profundidade das árvores aumentada (`depth=10`) para capturar relações complexas
- Regularização (`l2_leaf_reg=3`) para evitar *overfitting*
- Função de perda `RMSE` e métrica de avaliação `RMSE`
- `random_seed=42`
- Parada precoce por iteração (`od_type='Iter'`) após 100 iterações sem melhora
- Progresso exibido a cada 5.000 iterações

É importante considerar a qualidade da base de dados utilizada para aprimorar a previsão. Quanto maior o número de casos na amostra, mais precisa será a previsão. O tratamento adequado dos dados antes da modelagem é essencial para evitar resultados enviesados por *outliers* ou valores extremos.

3.3.1 Regressão Ridge

Regressão Ridge é uma extensão da Regressão Linear que inclui um termo de penalização L2. Esse método é especialmente útil quando há multicolinearidade nas variáveis independentes (previsores), que pode tornar os coeficientes da regressão instáveis e amplificar o *overfitting* do modelo. A Regressão Ridge reduz essa variabilidade adicionando uma penalidade aos coeficientes, o que leva a uma melhor generalização do modelo. (GÉRON, 2019)

Dado um conjunto de dados de treinamento com m amostras e n características, onde:

- y_i representa o valor da variável dependente para a i-ésima amostra,
- $x_i \in \mathbb{R}^n$ é o vetor de características da i-ésima amostra,
- $\beta \in \mathbb{R}^n$ são os coeficientes que queremos estimar.

A Regressão Ridge busca minimizar a seguinte função de custo (GÉRON, 2019) p.194:

$$J(\beta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \alpha\sum_{j=1}^{n}\beta_j^2$$

onde:

- O termo $\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$ representa o Erro Quadrático Médio,
- $\hat{y}_i = \beta_0 + \sum_{j=1}^{n}\beta_j x_{ij}$ é a previsão para a i-ésima amostra,

- $\alpha \geq 0$ é o parâmetro de regularização, que controla a penalização aplicada aos coeficientes β_j.

O termo de penalização, $\alpha \sum_{j=1}^{n} \beta_j^2$, é conhecido como norma L2, e sua inclusão no modelo restringe os valores dos coeficientes, reduzindo o impacto de variáveis colineares. Quando $\alpha = 0$, a Regressão Ridge é equivalente a uma regressão linear simples; para valores de α maiores, os coeficientes são penalizados de forma mais intensa, o que diminui a variância do modelo.

O parâmetro de regularização α desempenha um papel fundamental na Regressão Ridge. A sua escolha afeta diretamente o *trade-off* entre viés e variância no modelo:

- **Quando α é pequeno**: A penalização sobre os coeficientes é baixa, o que permite que os coeficientes assumam valores grandes, aumentando o risco de *overfitting*.
- **Quando α é grande**: A penalização sobre os coeficientes é alta, levando-os a valores próximos de zero, o que pode causar *underfitting*.

A escolha ideal de α é geralmente feita através de validação cruzada, onde diferentes valores de α são testados, e aquele que minimiza o erro de validação é escolhido.

3.3.2 Regressão Lasso

A Regressão Lasso (*Least Absolute Shrinkage and Selection Operator*) é uma variação da regressão linear que inclui uma penalização L1, promovendo esparsidade nos coeficientes. Em outras palavras, a regressão Lasso pode definir alguns coeficientes como zero, o que efetivamente elimina variáveis irrelevantes do modelo. Esse recurso faz da Lasso uma técnica útil para seleção de variáveis, especialmente em conjuntos de dados com muitas variáveis. (TIBSHIRANI, 1996)

Dado um conjunto de dados de treinamento com m amostras e n características, onde:

- y_i representa o valor da variável dependente para a i-ésima amostra,
- $x_i \in \mathbb{R}^n$ é o vetor de características da i-ésima amostra,
- $\beta \in \mathbb{R}^n$ são os coeficientes do modelo que queremos estimar.

A Regressão Lasso minimiza a seguinte função de custo (TIBSHIRANI, 1996):

$$J(\beta) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} |\beta_j|,$$

onde:

- O termo $\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$ representa o Erro Quadrático Médio,
- $\hat{y}_i = \beta_0 + \sum_{j=1}^{n}\beta_j x_{ij}$ é a previsão para a i-ésima amostra,
- $\alpha \geq 0$ é o parâmetro de regularização, que controla a penalização aplicada aos coeficientes β_j.

O termo de penalização $\alpha\sum_{j=1}^{n}|\beta_j|$ representa a norma L1 dos coeficientes. Essa penalização força alguns coeficientes a serem exatamente zero, o que resulta em um modelo mais esparso e facilita a interpretação ao identificar as variáveis mais importantes.

3.3.3 Elastic Net

Elastic Net é um modelo de regressão linear que combina as penalizações L1 e L2 da Regressão Lasso e da Regressão Ridge, respectivamente. Esse modelo é particularmente útil em situações onde há uma grande quantidade de variáveis preditoras correlacionadas, pois ele herda a propriedade de seleção de variáveis da Regressão Lasso e a estabilidade da Regressão Ridge. Elastic Net é amplamente utilizado para modelagem preditiva em que se deseja um modelo esparso (com alguns coeficientes zero) e, ao mesmo tempo, evitar problemas de multicolinearidade. (HASTIE *et al.*, 2009)

Dado um conjunto de dados de treinamento com m amostras e n características, onde:

- y_i representa o valor da variável dependente para a i-ésima amostra,
- $x_i \in \mathbb{R}^n$ é o vetor de características da i-ésima amostra,
- $\beta \in \mathbb{R}^n$ são os coeficientes que queremos estimar.

A Elastic Net minimiza a seguinte função de custo (GÉRON, 2019):

$$J(\beta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + r\alpha\sum_{i=1}^{n}|\beta_i| + \frac{1-r}{2}\alpha\sum_{i=1}^{n}\beta_i^2$$

onde:

- O termo $\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$ representa o Erro Quadrático Médio,
- O termo $r\alpha\sum_{i=1}^{n}|\beta_i|$ vem do Lasso e impõe uma penalidade no valor absoluto dos coeficientes (L_1-norma),
- O termo $\frac{1-r}{2}\alpha\sum_{i=1}^{n}\beta_i^2$ vem do Ridge e penaliza o quadrado dos coeficientes (L_2-norma).

O termo $r\alpha \sum_{i=1}^{n} |\beta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \beta_i^2$ representa a combinação das penalizações L1 e L2. Quando $(r = 1)$, o modelo se torna equivalente à Regressão Lasso, e quando $(r = 0)$, ele se torna equivalente à Regressão Ridge.

3.3.4 MLP

O MLP (*Multi-Layer Perceptron* ou Rede Neural Multicamadas) é uma rede neural de múltiplas camadas usada para problemas de regressão, permitindo capturar padrões complexos nos dados. O MLP consiste em uma série de camadas de neurônios organizadas em uma arquitetura *feed-forward* (Figura 3.2), onde a informação flui das camadas de entrada para a camada de saída passando por uma ou mais camadas ocultas. Esse modelo é amplamente utilizado em problemas de aprendizado supervisionado que envolvem dados não lineares, devido à sua capacidade de modelar relações complexas. (BISHOP, 2006)

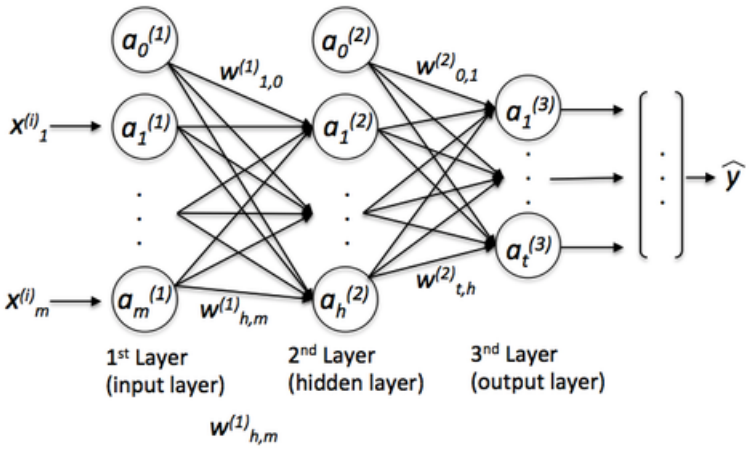

FIGURA 3.2 – Esquematização modelo MLP. (RASCHKA, 2018)

O MLP é composto por três tipos de camadas principais:

- **Camada de Entrada**: Recebe as variáveis preditoras $(x_1, x_2, \ldots, x_n)$, onde cada neurônio da camada de entrada representa uma característica dos dados.
- **Camadas Ocultas**: Responsáveis pelo aprendizado das representações complexas dos dados. Cada neurônio nas camadas ocultas aplica uma transformação linear seguida por uma função de ativação não linear.
- **Camada de Saída**: Contém um único neurônio em problemas de regressão, fornecendo a previsão final do modelo.

Para uma rede neural com uma camada oculta, o cálculo da saída para uma entrada x ocorre da seguinte forma:

1. **Primeira Camada Oculta**:

$$z^{(1)} = W^{(1)}x + b^{(1)}$$

$$a^{(1)} = g(z^{(1)})$$

onde:

- $W^{(1)}$ e $b^{(1)}$ são os pesos e o *bias* da camada,
- $g(\cdot)$ é a função de ativação, como a `tanh` ou `ReLU`.

2. **Camada de Saída**:

$$\hat{y} = W^{(2)}a^{(1)} + b^{(2)}$$

onde:

- $W^{(2)}$ e $b^{(2)}$ são os pesos e o *bias* da camada de saída.

Cada camada oculta aplica uma transformação linear seguida por uma função de ativação, permitindo que o modelo capture relações não lineares entre as variáveis de entrada e a saída.

As funções de ativação são essenciais para introduzir não linearidade no MLP e permitir que o modelo aprenda representações mais complexas. (GOODFELLOW *et al.*, 2016)

As funções de ativação mais comuns em redes MLP são:

- **ReLU (*Rectified Linear Unit*)**: $g(z) = \max(0, z)$. Essa função é popular devido à sua eficiência computacional e ao fato de mitigar o problema de gradientes desaparecendo.
- **Tanh (Tangente Hiperbólica)**: $g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. A `tanh` limita a saída no intervalo $[-1, 1]$, sendo útil para normalizar as ativações.

A escolha da função de ativação pode impactar o desempenho e a convergência do modelo.

Em problemas de regressão, o MLP usa geralmente o Erro Quadrático Médio (MSE) como função de custo (BISHOP, 2006):

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

onde:

- y_i é o valor real da variável dependente para a i-ésima amostra,
- $\hat{y}_i$ é a previsão do modelo para a i-ésima amostra,
- m é o número total de amostras.

O MLP utiliza o algoritmo de *backpropagation* em conjunto com métodos de otimização, como o *Adam* ou *Stochastic Gradient Descent* (SGD), para minimizar o erro e ajustar os pesos W e *bias* b da rede. A atualização dos pesos é feita de acordo com a direção negativa do gradiente da função de custo em relação aos pesos, com uma taxa de aprendizado η. (GÉRON, 2019)

O desempenho do MLP depende de vários hiperparâmetros, os quais precisam ser ajustados cuidadosamente:

- **hidden_layer_sizes**: Define o número de camadas ocultas e o número de neurônios em cada camada. Exemplo: `(10, 5)` cria uma rede com duas camadas ocultas, com 10 neurônios na primeira e 5 na segunda.
- **activation**: A função de ativação para as camadas ocultas, como `tanh` ou `relu`.
- **solver**: O algoritmo de otimização, como *adam* ou SGD.
- **alpha**: O parâmetro de regularização L2, que ajuda a evitar *overfitting*.
- **learning_rate**: Controla a taxa de aprendizado do modelo. Com o valor *adaptive*, a taxa de aprendizado ajusta-se automaticamente conforme o modelo converge.
- **max_iter**: O número máximo de iterações para o otimizador.
- **early_stopping**: Se ativado, interrompe o treinamento se o erro de validação não melhorar após um número de iterações.

3.3.5 SVR

O SVR (*Support Vector Regressor*) é uma técnica de aprendizado supervisionado que se baseia no conceito de *Support Vector Machines* (SVM), originalmente desenvolvido para classificação, mas adaptado para tarefas de regressão. No caso do modelo utilizado, `SVR(kernel='rbf', C=1.0, epsilon=0.1)`, a regressão se dá utilizando o Kernel Radial Base ou *Radial Basis Function* (RBF), ideal para capturar padrões não lineares presentes nos dados. O objetivo do SVR é encontrar uma função que se ajuste aos dados de treinamento, enquanto mantém um erro máximo (ϵ) entre a previsão e os valores reais, minimizando a complexidade do modelo e garantindo boa generalização.

A principal característica do SVR é o uso de uma função de perda ϵ-*insensitive*, que desconsidera erros menores que ϵ, tornando o modelo mais robusto e menos propenso a capturar ruídos nos dados. No caso do modelo em questão (GÉRON, 2019):

- O kernel RBF permite mapear os dados para um espaço de maior dimensionalidade, onde relações não lineares se tornam lineares.
- O parâmetro $C = 1.0$ controla o *trade-off* entre ajustar os dados de treinamento e a regularização. Valores médios, como $C = 1.0$, oferecem um equilíbrio entre precisão e generalização.
- O parâmetro $\epsilon = 0.1$ define a largura do "corredor de tolerância" em torno da função ajustada. Previsões que caem dentro dessa margem não são penalizadas, garantindo maior robustez a pequenas variações.

Em termos matemáticos, o SVR busca minimizar a seguinte função de custo (GÉRON, 2019):

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

Sujeito às restrições:

$$\begin{cases} y_i - (w^T\phi(x_i) + b) \leq \epsilon + \xi_i, \\ (w^T\phi(x_i) + b) - y_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases}$$

Onde:

- $\phi(x_i)$ é a transformação para o espaço de características, realizada pelo kernel RBF,
- $\|w\|^2$ representa a complexidade do modelo,
- C é o parâmetro de regularização,
- ξ_i e ξ_i^* são variáveis de folga que capturam desvios fora da margem de tolerância ϵ.

A função de perda ϵ-*insensitive* é definida como:

$$L(y, \hat{y}) = \begin{cases} 0 & \text{se } |y - \hat{y}| \leq \epsilon, \\ |y - \hat{y}| - \epsilon & \text{caso contrário.} \end{cases}$$

Essa configuração é ideal para tarefas onde há relações não lineares entre as variáveis de entrada e a saída, e se beneficia do poder do kernel RBF para capturar tais padrões. O kernel RBF é uma escolha padrão robusta para muitos problemas devido à sua capacidade de modelar relações complexas e generalizar bem em cenários com dados ruidosos ou com padrões não triviais. (GÉRON, 2019)

3.3.6 Random Forest

O Random Forest é um modelo de ML que combina múltiplas árvores de decisão para fazer previsões robustas e reduzir o risco de *overfitting*. Esse modelo é amplamente utilizado em problemas de regressão devido à sua capacidade de capturar relações complexas entre as variáveis preditoras e a variável resposta. A principal ideia do Random Forest é construir várias árvores de decisão em subconjuntos aleatórios dos dados de treinamento e, em seguida, combinar as previsões de todas as árvores para obter uma predição final mais precisa (esquema ilustrado na Figura 3.3). (BISHOP, 2006)

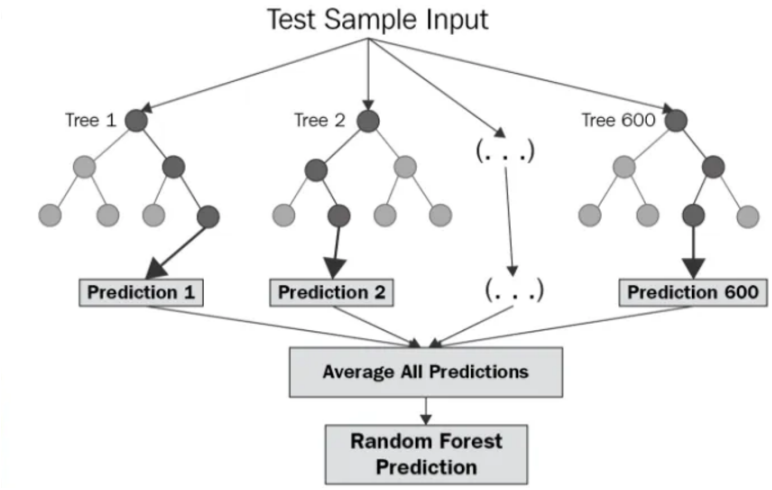

FIGURA 3.3 – Previsão no Random Forest: cada entrada é avaliada por múltiplas árvores de decisão, gerando previsões individuais. Essas previsões são combinadas por média, para regressão, para produzir a previsão final, garantindo maior robustez e precisão. (CHAYA, 2020)

No diagrama, observa-se que cada árvore é treinada com um subconjunto aleatório dos dados de treinamento e das características disponíveis. Durante a fase de predição, cada árvore individual fornece sua previsão, e a decisão final é obtida pela média das previsões.

O Random Forest funciona criando um conjunto de T árvores de decisão, onde cada árvore é treinada em um subconjunto diferente dos dados de treinamento, gerado por amostragem aleatória com reposição (*bootstrap*). Além disso, em cada nó de uma árvore, uma seleção aleatória de variáveis é considerada para dividir os dados, o que aumenta a diversidade entre as árvores. (BREIMAN, 2001)

Para uma nova amostra x, a predição do Random Forest é obtida pela média das previsões individuais das árvores (BISHOP, 2006):

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \text{tree}_t(x)$$

onde:

- T é o número total de árvores na floresta,
- $\text{tree}_t(x)$ é a previsão da t-ésima árvore para a entrada x.

O Random Forest utiliza duas técnicas principais de aleatoriedade para aumentar a diversidade entre as árvores e reduzir o *overfitting*:

- **Amostragem de dados (*bootstrap*)**: Cada árvore é treinada em um subconjunto de dados gerado por amostragem com reposição. Isso significa que, para cada árvore, algumas amostras do conjunto de dados original podem ser repetidas, enquanto outras podem não ser incluídas.
- **Amostragem de variáveis**: Em cada nó de uma árvore, apenas um subconjunto aleatório de variáveis é considerado para a divisão. Essa seleção aleatória de variáveis contribui para tornar as árvores menos correlacionadas entre si, o que melhora a robustez do modelo final.

O desempenho do Random Forest pode ser ajustado através de diversos hiperparâmetros. Alguns dos mais importantes são:

- **n_estimators**: O número de árvores na floresta. Em geral, quanto maior o número de árvores, melhor a precisão do modelo, mas o custo computacional também aumenta.
- **max_depth**: A profundidade máxima de cada árvore. Um valor mais alto permite que cada árvore se ajuste mais aos dados de treinamento, mas aumenta o risco de *overfitting*. Um valor menor torna o modelo mais simples e reduz o *overfitting*.
- **max_features**: O número máximo de variáveis a serem consideradas para divisão em cada nó. Um valor mais baixo aumenta a diversidade entre as árvores, enquanto um valor mais alto permite que cada árvore capture mais detalhes dos dados.
- **min_samples_split** e **min_samples_leaf**: Controlam o número mínimo de amostras necessárias para dividir um nó e para formar uma folha, respectivamente. Esses parâmetros ajudam a controlar a complexidade de cada árvore.

Uma característica interessante do Random Forest é sua capacidade de fornecer uma medida de importância das variáveis. A importância de uma variável pode ser calculada com base na redução da impureza em todos os nós onde essa variável é usada como critério de divisão. Em problemas de regressão, a impureza é medida pelo Erro Quadrático Médio (MSE). A importância de uma variável x_j é então dada por:

$$\text{Importância}(x_j) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\text{nó em tree}_t} \Delta\text{MSE}_{\text{nó},x_j}$$

onde $\Delta\text{MSE}_{\text{nó},x_j}$ representa a redução de MSE em cada nó dividido pela variável x_j.

3.3.7 XGBoost Regressor

O XGBoost Regressor é um modelo de aprendizado de máquina baseado em *boosting* de árvores de decisão, projetado para ser altamente eficiente e escalável. A principal diferença entre Random Forest e XGBoost está na forma como as árvores são construídas e utilizadas no processo de aprendizado. Enquanto a Random Forest utiliza uma abordagem de *bagging* (paralelismo e independência entre árvores), o XGBoost utiliza *boosting* (sequencial e interdependente entre árvores). (CHEN; GUESTRIN, 2016)

O XGBoost, ou *Extreme Gradient Boosting*, é uma implementação aprimorada do Gradient Boosting que utiliza uma série de otimizações para melhorar o desempenho computacional e a precisão do modelo. Ele é amplamente utilizado em tarefas de regressão e classificação devido à sua capacidade de lidar com dados grandes e complexos, além de reduzir o risco de *overfitting*.

O XGBoost Regressor constrói o modelo de forma iterativa, adicionando uma nova árvore de decisão em cada etapa para corrigir os erros das previsões anteriores. Em vez de construir uma única árvore grande, o XGBoost cria um conjunto de árvores mais simples (também chamadas de "árvores fracas") que, combinadas, formam um modelo poderoso.

Para uma nova amostra x, a previsão do XGBoost é a soma das previsões de todas as árvores construídas até aquele ponto:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i)$$

onde:

- T é o número total de árvores no modelo,
- f_t representa a t-ésima árvore,

- $\hat{y}_i$ é a previsão final para a amostra i.

Cada árvore f_t é treinada para minimizar o erro residual das previsões acumuladas até a iteração anterior.

A função de custo do XGBoost é composta por dois termos principais: um termo de perda que mede o erro entre as previsões e os valores reais, e um termo de regularização que penaliza a complexidade do modelo. Para uma amostra y_i com previsão $\hat{y}_i$, a função de custo J é dada por (CHEN; GUESTRIN, 2016):

$$J = \sum_{i=1}^{m} \mathcal{L}(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

onde:

- $\mathcal{L}(y_i, \hat{y}_i)$ é a função de perda, geralmente o erro quadrático $(y_i - \hat{y}_i)^2$ para regressão,
- $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{n} w_j^2$ é o termo de regularização, onde γ controla o número de nós folha em cada árvore e λ controla a penalização sobre os pesos w_j das folhas.

A regularização é uma característica chave do XGBoost, pois evita o *overfitting*, tornando o modelo mais robusto.

Os hiperparâmetros do XGBoost controlam vários aspectos do treinamento e da regularização do modelo. Alguns dos mais importantes são:

- **n_estimators**: O número total de árvores no modelo. Um valor mais alto pode aumentar a precisão, mas também o custo computacional.
- **max_depth**: A profundidade máxima das árvores. Um valor maior permite que cada árvore capture padrões mais complexos, mas aumenta o risco de *overfitting*.
- **learning_rate**: Controla a contribuição de cada árvore para o modelo final. Um valor menor de `learning_rate` requer mais árvores para obter o mesmo ajuste.
- **gamma** (γ): Controla a penalização para adicionar uma nova divisão em uma árvore. Valores mais altos tornam o modelo mais conservador.
- **lambda** (λ) e **alpha** (α): Parâmetros de regularização L2 e L1, respectivamente, que ajudam a reduzir o *overfitting*.

3.3.8 Gradient Boosting

O Gradient Boosting é um modelo de ML que utiliza a técnica de *boosting* para criar uma combinação sequencial de modelos de árvore de decisão, de forma que cada nova

árvore corrige os erros cometidos pelas anteriores. Embora considerado uma versão simplificada do método XGBoost, esse modelo é importante pois, para conjuntos de dados muito pequenos, o ganho de performance do XGBoost pode não ser significativo e o Gradient Boosting clássico pode ser suficiente, trazendo também economias computacionais, já que requer menos recursos e tempo de processamento.

O Gradient Boosting é construído de maneira aditiva, ajustando uma nova árvore f_t em cada etapa t para corrigir o erro residual das previsões acumuladas até aquele ponto. Para uma entrada x, a predição final do Gradient Boosting é a soma das previsões de todas as árvores construídas (FRIEDMAN, 2001):

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i)$$

onde:

- T é o número total de árvores no modelo,
- $f_t(x_i)$ representa a previsão da t-ésima árvore para a entrada x_i,
- $\hat{y}_i$ é a previsão final para a i-ésima amostra.

Cada nova árvore é ajustada para minimizar o erro residual das árvores anteriores, o que permite ao modelo se adaptar e melhorar continuamente a precisão.

O Gradient Boosting minimiza uma função de custo que mede a diferença entre os valores reais y_i e as previsões $\hat{y}_i$. Para problemas de regressão, a função de custo é geralmente o MSE (FRIEDMAN, 2001):

$$J = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

onde:

- m é o número de amostras,
- y_i é o valor real da i-ésima amostra,
- $\hat{y}_i$ é a previsão do modelo para a i-ésima amostra.

O Gradient Boosting utiliza uma abordagem baseada em gradiente para ajustar o modelo, adicionando uma nova árvore em cada etapa para minimizar o erro residual. Em cada iteração t, o modelo ajusta uma árvore f_t para prever o gradiente negativo da função de custo, que representa a direção do erro residual.

A previsão acumulada após a iteração t é dada por:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

onde:

- η é a taxa de aprendizado, que controla a contribuição de cada árvore para o modelo final,
- $f_t(x_i)$ é a previsão da t-ésima árvore para a amostra x_i,
- $\hat{y}_i^{(t-1)}$ é a previsão acumulada até a iteração anterior.

A taxa de aprendizado η é um hiperparâmetro importante, pois uma taxa muito alta pode fazer o modelo convergir rapidamente, mas também aumentar o risco de *overfitting*, enquanto uma taxa muito baixa pode requerer muitas árvores para obter um bom ajuste.

O Gradient Boosting possui vários hiperparâmetros importantes que afetam seu desempenho e devem ser ajustados cuidadosamente:

- **n_estimators**: O número de árvores a serem criadas no modelo. Valores maiores aumentam a precisão, mas também o custo computacional.
- **learning_rate** (η): Controla a contribuição de cada árvore para o modelo final. Uma taxa de aprendizado menor geralmente melhora o desempenho, mas requer um número maior de árvores.
- **max_depth**: Define a profundidade máxima de cada árvore. Árvores mais profundas podem capturar padrões mais complexos, mas aumentam o risco de *overfitting*.
- **min_samples_split** e **min_samples_leaf**: Controlam o número mínimo de amostras necessárias para dividir um nó e para formar uma folha, respectivamente, o que ajuda a regular a complexidade de cada árvore.

3.3.9 CatBoost

O CatBoost é um modelo também baseado em *boosting* de árvores de decisão, tal qual Gradient Boost e XGBoost, desenvolvido especificamente para lidar de forma eficiente com variáveis categóricas e para otimizar o desempenho de *boosting*. CatBoost, abreviação de *Categorical Boosting*, utiliza um algoritmo que preserva a ordem dos dados durante o treinamento, evitando o *overfitting* e proporcionando previsões mais precisas. CatBoost é amplamente utilizado em problemas de regressão e classificação, especialmente em cenários

onde há uma combinação de variáveis categóricas e numéricas. (PROKHORENKOVA *et al.*, 2019)

O CatBoost constrói um modelo de *boosting* adicionando árvores de decisão sequencialmente. Cada nova árvore é ajustada para corrigir os erros cometidos pelas árvores anteriores, de forma que o modelo final seja uma combinação das previsões de todas as árvores. Para uma nova amostra x, a predição final do CatBoost é a soma das previsões de todas as árvores ajustadas:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i)$$

onde:

- T é o número total de árvores no modelo,
- $f_t(x_i)$ representa a previsão da t-ésima árvore para a entrada x_i,
- $\hat{y}_i$ é a previsão final para a i-ésima amostra.

O CatBoost utiliza uma técnica de codificação para variáveis categóricas que permite o uso eficiente dessas variáveis, convertendo-as em representações numéricas de maneira que preserva a ordem dos dados e evita vazamento de informação.

Para problemas de regressão, o CatBoost Regressor geralmente utiliza o Erro Quadrático Médio (MSE) como função de custo:

$$J = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

onde:

- m é o número de amostras,
- y_i é o valor real da i-ésima amostra,
- $\hat{y}_i$ é a previsão do modelo para a i-ésima amostra.

O CatBoost otimiza essa função de custo utilizando uma abordagem de *gradient boosting* e um método de codificação de variáveis que evita o *overfitting* causado pelo vazamento de informação.

Uma das principais características do CatBoost é sua capacidade de lidar diretamente com variáveis categóricas sem necessidade de pré-processamento, como *one-hot encoding*. Em vez disso, o CatBoost utiliza uma técnica de codificação de ordem que calcula a

média de valores alvo de cada categoria ao longo do conjunto de dados, garantindo que apenas informações disponíveis até aquele ponto do treinamento sejam usadas. Isso evita vazamentos de informação e melhora a robustez do modelo.

Os hiperparâmetros do CatBoost permitem controlar vários aspectos do treinamento e do desempenho do modelo. Alguns dos mais importantes são:

- **iterations**: O número de árvores (iterações) a serem criadas no modelo. Um valor maior pode melhorar a precisão, mas aumenta o tempo de treinamento.
- **learning_rate**: Controla a contribuição de cada árvore para o modelo final. Um valor menor pode aumentar a precisão, mas requer um número maior de iterações.
- **depth**: Define a profundidade máxima das árvores. Profundidades maiores permitem capturar padrões complexos, mas aumentam o risco de *overfitting*.
- **l2_leaf_reg**: O parâmetro de regularização L2, que ajuda a reduzir o *overfitting*.
- **od_type** e **od_wait**: Parâmetros de *early stopping*, que interrompem o treinamento se o erro de validação não melhorar após um número específico de iterações.

3.4 Avaliação dos Modelos

Para avaliar o desempenho dos modelos de predição, foram utilizados dois principais parâmetros estatísticos: o RMSE (*Root Mean Squared Error* ou Raiz do Erro Quadrático Médio) e o Coeficiente de Determinação, R^2.

O RMSE mede a diferença média entre os valores preditos pelo modelo e os valores reais, expressando essa diferença em unidades da variável predita. Quanto menor o valor de RMSE, melhor é o desempenho do modelo, pois indica que as previsões estão mais próximas dos valores observados. Esse parâmetro é especialmente útil em problemas de regressão, pois penaliza erros maiores, sendo sensível a discrepâncias que podem ocorrer em previsões de custo mais elevadas. O R^2, por sua vez, mede a proporção da variabilidade dos dados que é explicada pelo modelo. Ele varia de 0 a 1, sendo que valores mais próximos de 1 indicam que o modelo explica grande parte da variação nos dados. Dessa forma, um R^2 alto sugere que o modelo possui uma boa capacidade de ajuste aos dados, capturando as tendências gerais do conjunto de observações. (AGRAWAL, 2024)

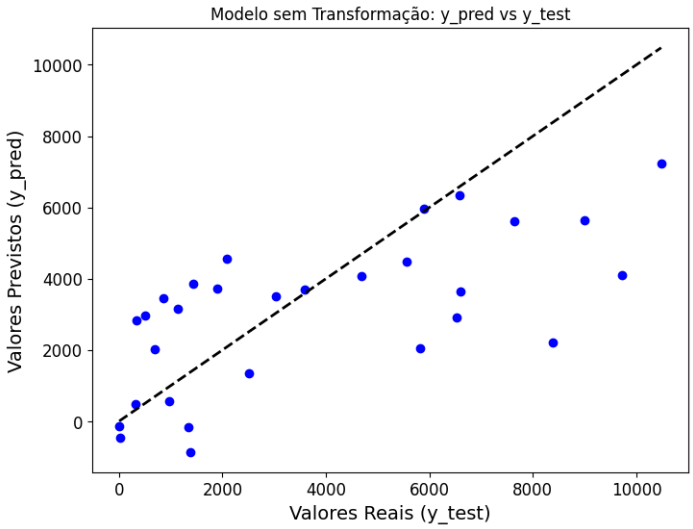

FIGURA 3.4 – Exemplo gráfico de Regressão Linear Simples gerado.

Além de avaliar o desempenho dos modelos com base nesses parâmetros, foi realizada uma comparação visual das previsões, por meio de gráficos de regressão (exemplo na Figura 3.4 que exibem a relação entre os valores preditos e os valores reais para cada modelo. Em cada gráfico, os valores de RMSE e R^2 foram apresentados como métricas de desempenho, permitindo uma análise visual clara das previsões em relação aos valores reais, além de uma comparação quantitativa entre os modelos. Essa abordagem facilita a identificação do modelo com maior precisão na estimativa dos custos das obras, possibilitando escolhas mais embasadas para aplicações práticas.

4 Resultados

4.1 Correlações entre as Variáveis

A matriz de correlação (Tabela 4.1) mede o grau de relação linear entre variáveis relacionadas à estimativa de custo nas obras da FAB analisadas. Os valores variam de -1 a 1, indicando a força e a direção das correlações, sendo que valores próximos a 1 indicam uma correlação positiva forte, valores próximos a -1 indicam uma correlação negativa forte, e valores próximos a 0 indicam ausência de correlação linear. Para se ter uma visualização rápida das intensidades das correlações, na Figura 4.1 é exibido um mapa de calor correspondente às correlações encontradas.

TABELA 4.1 – Matriz de Correlação entre as Variáveis

	Custo Unitario	**Quantitativo Total**	**Tipo Obra**	**Tipologia Benf.**	**UF**	**Unidade Referencia**
Custo_Unitario	1.000	-0.433	0.000	0.267	0.000	0.271
Quantitativo_Total	-0.433	1.000	0.000	0.389	0.000	0.432
Tipo_Obra	0.000	0.000	1.000	0.382	0.327	0.205
Tipologia_Benfeitoria	0.267	0.389	0.382	1.000	0.049	0.978
UF	0.000	0.000	0.327	0.049	1.000	0.000
Unidade_Referencia	0.271	0.432	0.205	0.978	0.000	1.000

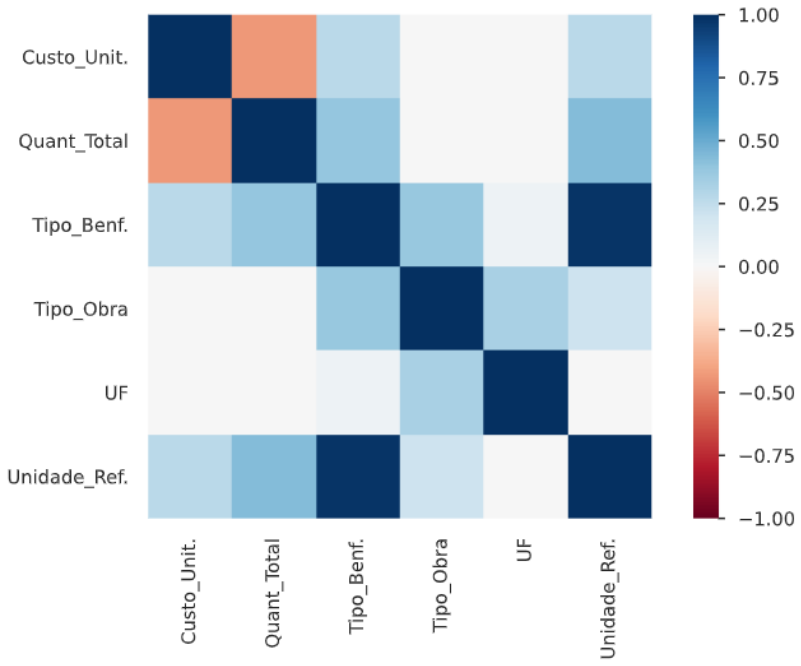

FIGURA 4.1 – Mapa de calor das correlações entre as variáveis.

A variável `Custo_Unitario` apresenta uma correlação negativa com a variável `Quantitativo_Total` (-0,433), sugerindo que, em geral, quando o quantitativo total aumenta, o custo unitário tende a diminuir. Esse comportamento pode estar associado a economias de escala, nas quais maiores volumes resultam em custos reduzidos por unidade. Além disso, `Custo_Unitario` apresenta uma correlação positiva moderada com a `Tipologia_Benfeitoria` (0,267), indicando que a mudança da espécie de benfeitoria pode gerar alterações singelas no custo unitário das obras. Por outro lado, as correlações de `Custo_Unitario` com as demais variáveis, como `Tipo_Obra` e `UF`, são muito próximas de zero, indicando que essas variáveis têm pouca influência direta sobre o custo unitário.

A variável `Quantitativo_Total`, por sua vez, possui uma correlação positiva moderada com `Tipologia_Benfeitoria` (0,389), sugerindo que diferentes tipos de benfeitorias demandam quantitativos distintos. Há também uma correlação positiva entre `Quantitativo_Total` e `Unidade_Referencia` (0,432), o que indica que variáveis padronizadas, como as unidades de referência, impactam significativamente o quantitativo total das obras.

Isso tem sentido ao se observar os valores de área contemplada pelas obras de ”SINALIZAÇÃO HORIZONTAL”, os quais são substancialmente maiores, na maior parte das observações, quando comparados a valores de área de ”VESTIÁRIO E ALOJAMENTO”, por exemplo. No primeiro caso, por se tratarem obras de pistas de pouso e decolagem ou *taxiways*, é esperado que a área correspondente seja maior que as áreas envolvidas em reforma ou construção de alojamento. Além disso, a tipologia de ”VESTIÁRIO E ALO-

JAMENTO" se refere a edificações térreas, ou seja, não aumenta devido à quantidade de pavimentos como em outras tipologias de edificações que alcançam quantitativos mais pronunciados.

`Tipo_Obra` demonstra correlação positiva moderada com a `Tipologia_Benfeitoria` (0,382). Isso pode ser explicado pelo número de obras de determinada benfeitoria estar muito mais presente em casos de reforma do que de construção, e vice-versa. Um exemplo disso é encontrado na tipologia "PRÉDIO ADMINISTRATIVO", a qual possui 20 obras de reforma contra apenas 5 de construção. Essa comparação é realista, pois realizam-se mais reformas em prédios administrativos da FAB que construção de novos.

A variável `Tipologia_Benfeitoria` apresenta a correlação mais forte da matriz, sendo altamente relacionada com `Unidade_Referencia` (0,978). Essa relação ocorre pois a unidade de referência é totalmente dependente da tipologia da obra, sendo esta última quem determina a unidade de referência que será utilizada. Embora reflitam características muito similares, optou-se por manter as duas variáveis na base de dados pois a unidade de referência reflete uma abordagem de classificação menos segmentada que a tipologia de benfeitoria, isto é, tipologias diferentes podem ser mensuradas pela mesma unidade de medida.

Por fim, `UF` apresenta uma correlação moderada com o `Tipo_Obra` (0,327), evidenciando que o tipo de obra está relacionado com características específicas do local onde será executada, indicando predominância ou de reforma ou de construção em uma dada região.

4.2 Resultados da Aplicação dos Modelos

4.2.1 Regressão Linear

O modelo Regressão Linear, conforme ilustrado na Figura 4.2, apresentou desempenho limitado. Sem transformações, o R^2 foi de 0,39 e o MSE de 6.299.879,34, evidenciando uma capacidade preditiva restrita. A aplicação da transformação Raiz Quadrada resultou em leve redução de desempenho, com R^2 de 0,37 e MSE de 6.530.956,23. A transformação Yeo-Johnson trouxe os piores resultados, com R^2 de 0,35 e MSE de 6.733.962,82. Os resultados indicam que o modelo não consegue capturar adequadamente padrões não lineares presentes nos dados, mesmo com transformações aplicadas.

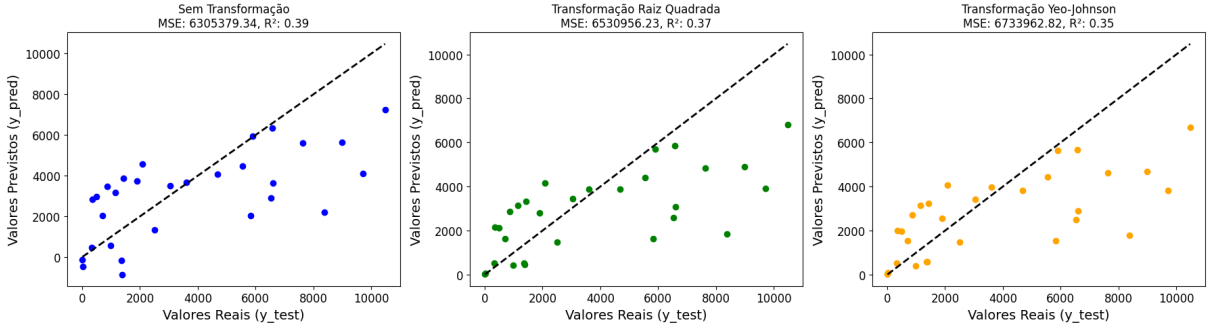

FIGURA 4.2 – Modelo de Regressão Linear. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.2 Regressão Ridge

O modelo Ridge, conforme evidenciado na Figura 4.3, apresentou desempenho inferior aos modelos não lineares. Sem transformações, o modelo obteve um R^2 de 0,40 e um MSE de 6.176.956,73. A transformação Raiz Quadrada reduziu o desempenho, com R^2 de 0,35 e MSE de 6.781.694,02. A transformação Yeo-Johnson também não trouxe ganhos, resultando em R^2 de 0,32 e MSE de 7.053.648,76. Esses resultados indicam que, embora a regularização L_2 melhore ligeiramente a estabilidade do modelo em relação à Regressão Linear Simples, o Ridge ainda não é adequado para capturar padrões não lineares.

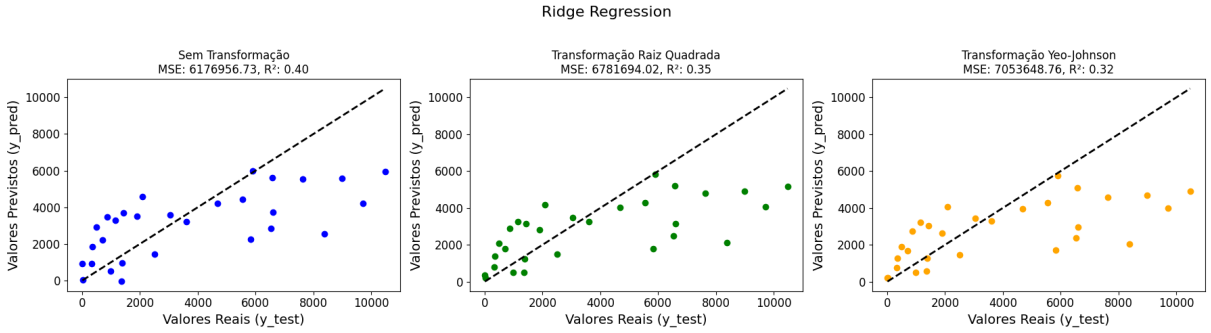

FIGURA 4.3 – Modelo de Regressão Ridge. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.3 Regressão Lasso

O modelo Lasso, ilustrado na Figura 4.4, apresentou um desempenho ainda mais limitado devido à regularização L_1, que forçou muitos coeficientes a zero. Sem transformações, o R^2 foi de 0,39 e o MSE de 6.292.824,64, valores semelhantes aos da Regressão Linear Simples. A transformação Raiz Quadrada resultou em R^2 de 0,34 e MSE de 6.865.649,92. A transformação Yeo-Johnson trouxe os piores resultados, com R^2 negativo (-0,01) e MSE de 10.478.016,04, demonstrando que a forte regularização prejudicou severamente a capacidade preditiva do modelo.

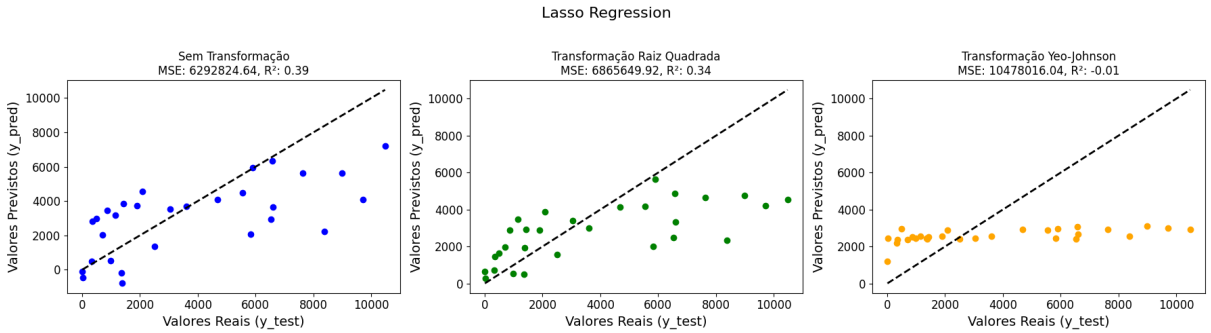

FIGURA 4.4 – Modelo de Regressão Lasso. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.4 Elastic Net

O modelo Elastic Net, conforme ilustrado na Figura 4.5, combinou regularizações L_1 e L_2, mas ainda apresentou desempenho insuficiente. Sem transformações, o R^2 foi de 0,34 e o MSE de 6.873.636,21. A transformação Raiz Quadrada piorou o ajuste, resultando em R^2 de 0,26 e MSE de 7.689.408,69. A transformação Yeo-Johnson foi ainda menos eficaz, com R^2 de 0,12 e MSE de 9.136.141,69. Esses resultados reforçam que o Elastic Net não é capaz de lidar com a complexidade dos dados avaliados.

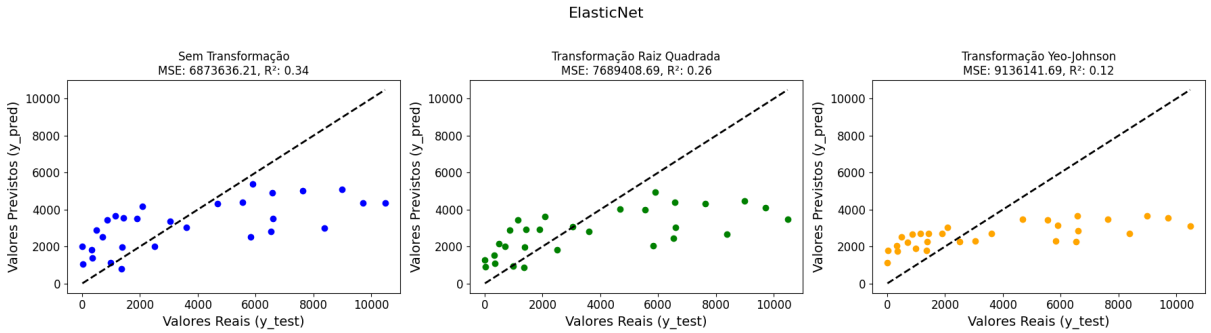

FIGURA 4.5 – Modelo de Elastic Net. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.5 MLP

O MLP, representado na Figura 4.6, apresentou o pior desempenho entre os modelos analisados. Em todas as condições, os valores de R^2 foram negativos, variando de -1,36 a -0,10, e os MSEs foram extremamente elevados, ultrapassando 24.500.000 na ausência de transformações. Este resultado reflete limitações na arquitetura da rede, que não foi capaz de capturar a complexidade dos dados, mesmo após as transformações aplicadas.

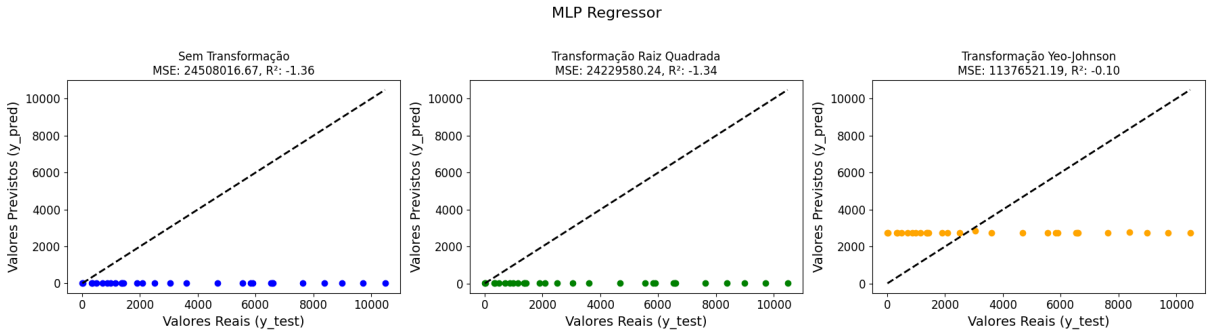

FIGURA 4.6 – Modelo de MLP (*Multilayer Perceptron*). À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.6 Support Vector Regressor

O SVR, cujo resultado é mostrado na Figura 4.7, também teve desempenho insatisfatório, com R^2 negativo em todas as condições, variando de -0,10 a -0,12, e MSEs superiores a 11.000.000. O modelo demonstrou incapacidade de capturar os padrões nos dados, mesmo após a aplicação das transformações Raiz Quadrada e Yeo-Johnson, indicando que não é adequado para este conjunto de dados.

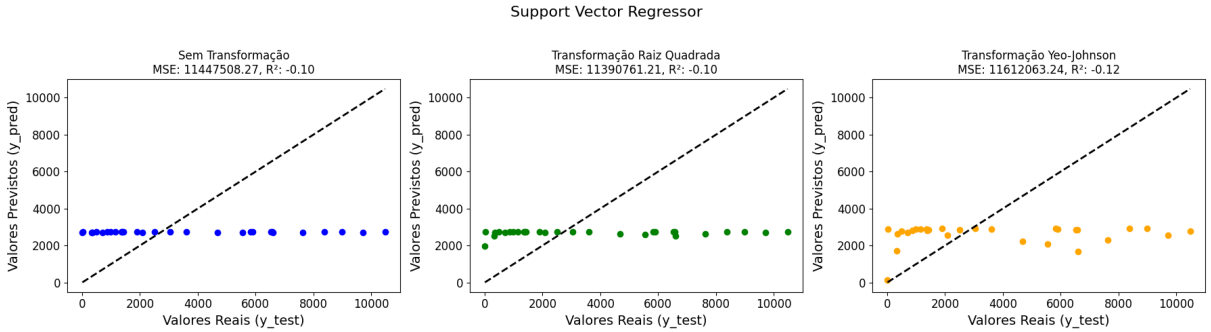

FIGURA 4.7 – Modelo de SVR (Support Vector Regressor). À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.7 Random Forest

O modelo Random Forest, conforme evidenciado na Figura 4.8, apresentou o melhor desempenho entre todos os métodos analisados. Sem transformações nos dados, o modelo obteve um R^2 de 0,52 e um MSE de 4.945.194,50, os melhores resultados registrados. A dispersão dos pontos ao longo da linha de identidade demonstra a capacidade do modelo de capturar padrões complexos presentes nos dados. Com a transformação Raiz Quadrada, o desempenho foi ligeiramente reduzido, com R^2 de 0,47 e MSE de 5.526.189,32, indicando que a transformação não trouxe melhorias significativas. De forma semelhante, a transformação Yeo-Johnson resultou em R^2 de 0,46 e MSE de 5.558.441,51, reforçando a

robustez do modelo mesmo sem ajustes nos dados. Portanto, o Random Forest destaca-se como o modelo mais eficiente e robusto para este problema.

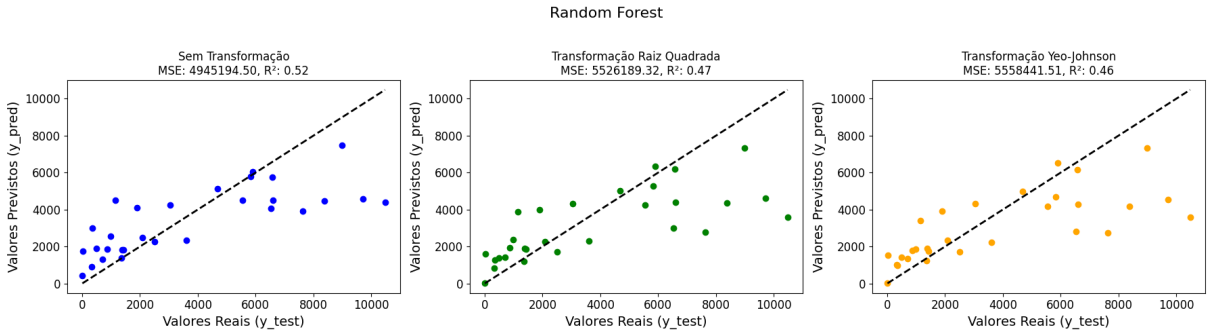

FIGURA 4.8 – Modelo de Random Forest. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.8 XGBoost

O modelo XGBoost, exibido na Figura 4.9, apresentou resultados consistentes, mas inferiores aos do Random Forest. Sem transformações, o modelo alcançou R^2 de 0,33 e MSE de 6.924.171,03. As transformações trouxeram melhorias discretas, com R^2 subindo para 0,40 e MSE reduzido para 6.230.866,01 na transformação Yeo-Johnson. Embora competitivo, o XGBoost não conseguiu superar a performance do Random Forest.

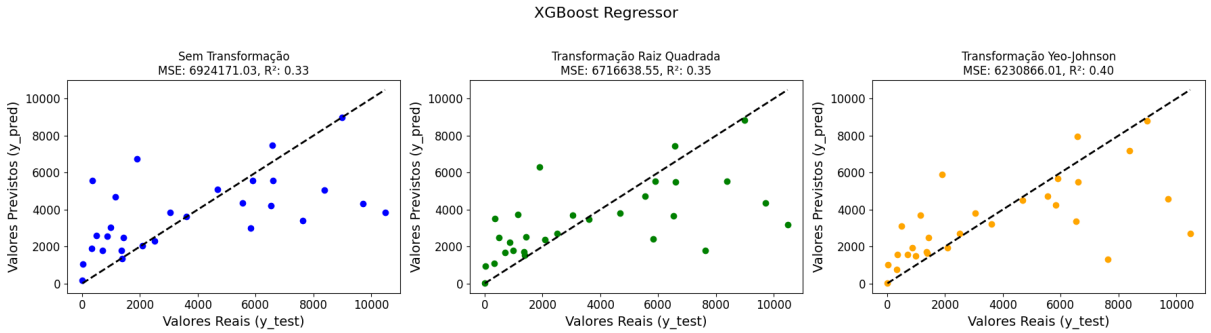

FIGURA 4.9 – Modelo de XGBoost. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.9 Gradient Boosting

O Gradient Boosting, exibido na Figura 4.10, apresentou resultados sólidos, embora também inferiores ao Random Forest. Na ausência de transformações, o modelo alcançou R^2 de 0,48 e MSE de 5.406.720,22, valores próximos aos do Random Forest, mas ligeiramente inferiores. As transformações dos dados reduziram seu desempenho, com R^2 de 0,41 e 0,43 e MSEs de 6.070.381,19 e 5.950.649,08 para as transformações Raiz Quadrada

e Yeo-Johnson, respectivamente. Embora eficiente, o Gradient Boosting mostrou menor robustez às transformações em comparação ao Random Forest.

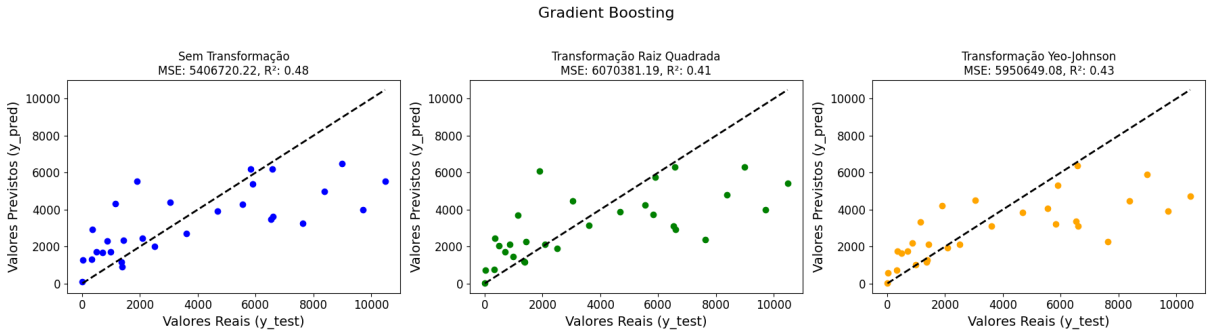

FIGURA 4.10 – Modelo de Gradient Boost. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.2.10 CatBoost

O modelo CatBoost, ilustrado na Figura 4.11, apresentou um desempenho competitivo, mas inferior ao Random Forest. Sem transformações, o CatBoost alcançou R^2 de 0,42 e MSE de 5.991.541,86, valores ligeiramente piores do que os do Random Forest. A aplicação das transformações Raiz Quadrada e Yeo-Johnson não trouxe ganhos significativos, resultando em R^2 de 0,41 e 0,42, respectivamente, e MSEs acima de 6.000.000 em ambas as condições. Apesar de ser um modelo avançado, sua eficiência não superou a do Random Forest neste conjunto de dados.

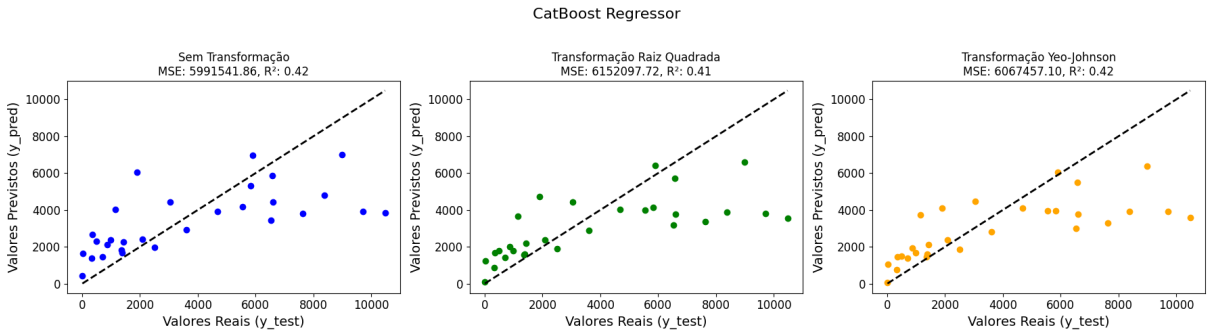

FIGURA 4.11 – Modelo de CatBoost. À esquerda sem transformação. No centro a com transformação Raiz Quadrada e à direita com transformação Yeo-Johnson dos dados.

4.3 Desempenho dos Modelos

Os resultados obtidos evidenciam diferenças significativas no desempenho entre os modelos analisados, destacando a superioridade dos métodos baseados em árvores em relação aos modelos lineares e redes neurais. A Tabela 4.2 apresenta o desempenho dos

modelos com e sem a aplicação de transformações, convertendo o MSE encontrado para RMSE. De maneira geral, os modelos baseados em árvores apresentaram maior robustez e eficácia na modelagem de dados complexos e não lineares, enquanto os modelos lineares demonstraram limitações claras e os modelos baseados em redes neurais exigiram ajustes mais refinados para atingir um desempenho satisfatório.

TABELA 4.2 – Desempenho dos modelos analisados com R^2 e RMSE em diferentes condições de transformação.

Modelo	**Sem Transf.**		**Raiz Quad.**		**Yeo-Johnson**	
	R^2	RMSE	R^2	RMSE	R^2	RMSE
Regressão Linear	0,39	2.509,96	0,37	2.555,57	0,35	2.594,99
Regressão Ridge	0,40	2.485,35	0,35	2.604,17	0,32	2.655,87
Regressão Lasso	0,39	2.508,55	0,34	2.620,24	-0,01	3.236,99
ElasticNet	0,34	2.621,76	0,26	2.773,00	0,12	3.022,59
MLP	-1,36	4.950,56	-1,34	4.922,35	-0,10	3.372,91
SVR	-0,10	3.383,42	-0,10	3.375,01	-0,12	3.407,65
Random Forest	0,52	2.223,78	0,47	2.350,78	0,46	2.357,64
XGBoost	0,33	2.631,38	0,35	2.591,65	0,40	2.496,17
Gradient Boosting	0,48	2.325,24	0,41	2.463,81	0,43	2.439,39
CatBoost	0,42	2.447,76	0,41	2.480,34	0,42	2.463,22

O Random Forest destacou-se como o mais robusto e eficaz, atingindo os maiores valores de R^2 (0,52 sem transformações) e os menores MSEs (4.945.194 sem transformações). Este desempenho reflete a capacidade do modelo em capturar padrões complexos sem a necessidade de ajustes adicionais nos dados. O Gradient Boosting teve resultados similares, mas ligeiramente inferiores, seguido pelo CatBoost e XGBoost, que também mostraram consistência. Em geral, esses modelos apresentaram boa estabilidade mesmo após transformações nos dados, indicando que sua robustez permite lidar com os padrões na forma original dos dados.

Por outro lado, os modelos lineares, como Regressão Linear, Ridge, Lasso e Elastic Net, apresentaram desempenho limitado. A Regressão Linear atingiu um R^2 de 0,39 e um MSE de 6.299.879 sem transformações, resultados que permaneceram insuficientes mesmo após transformações. O Ridge, que utiliza regularização L_2, mostrou desempenho ligeiramente melhor, mas ainda abaixo dos modelos não lineares. O Lasso, devido à regularização L_1, apresentou pior desempenho em algumas condições, como R^2 negativo (-0,01) após a transformação Yeo-Johnson, demonstrando que a regularização forçou coeficientes importantes a zero, prejudicando o ajuste. O Elastic Net, que combina regularizações L_1 e L_2, apresentou os piores resultados entre os modelos lineares, com R^2 de 0,12 e MSE de 9.136.142 na transformação Yeo-Johnson, indicando sua incapacidade de lidar com a

complexidade dos dados.

O desempenho do MLP foi o menos satisfatório entre os métodos analisados, com valores de R^2 negativos em todas as condições e MSE extremamente elevados, como 24.508.017 sem transformações. Isso sugere que a arquitetura da rede neural não foi configurada adequadamente para capturar os padrões nos dados. Redes neurais, em geral, requerem ajustes cuidadosos na arquitetura, como o número de camadas, neurônios e funções de ativação, além de otimização dos hiperparâmetros, para obter um desempenho competitivo.

O Support Vector Regressor (SVR) também apresentou resultados insatisfatórios, com R^2 negativos entre -0,10 e -0,12, e MSE acima de 11.000.000 em todas as condições testadas. Isso reflete a incapacidade do modelo de capturar os padrões nos dados, mesmo após a aplicação de transformações como Raiz Quadrada e Yeo-Johnson.

Por fim, os resultados mostram que as transformações aplicadas aos dados tiveram impacto limitado no desempenho dos modelos mais robustos, como Random Forest e Gradient Boosting, que já capturam padrões complexos sem ajustes nos dados. Para os modelos lineares, as transformações trouxeram pequenas melhorias, mas ainda insuficientes para torná-los competitivos. Modelos como o MLP e o SVR mostraram-se inadequados para este conjunto de dados, independentemente das transformações.

Em resumo, o Random Forest destacou-se como o modelo mais eficaz e robusto para este problema, sendo o mais indicado para capturar padrões complexos nos dados. Os modelos lineares e de redes neurais exigem ajustes significativos para melhorar seu desempenho, enquanto os modelos avançados, como Gradient Boosting e CatBoost, ofereceram alternativas consistentes, mas ligeiramente inferiores ao Random Forest.

4.4 Impacto das Transformações nos Dados

A análise evidenciou que o impacto dessas transformações variou consideravelmente entre os modelos testados. Para os modelos mais robustos, como Random Forest e CatBoost, o impacto das transformações foi limitado. Esses modelos já possuem mecanismos internos que permitem lidar com distribuições não uniformes, como divisões em *nodes* baseadas em percentis. Os resultados mostraram que ambos os modelos mantiveram seu desempenho robusto mesmo sem ajustes nos dados, com pequenas variações no R^2 e MSE após a aplicação das transformações. Por exemplo, o Random Forest apresentou R^2 de 0,52 sem transformações, reduzindo apenas para 0,47 com a transformação Raiz Quadrada e para 0,46 com a transformação Yeo-Johnson.

Modelos como Gradient Boosting e XGBoost seguiram um padrão semelhante, com de-

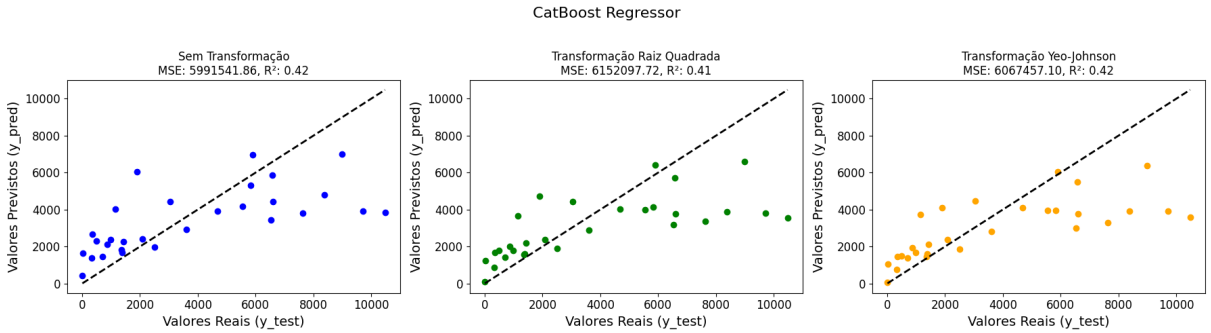

FIGURA 4.12 – Desempenho do CatBoost com Diferentes Transformações

sempenho levemente afetado pelas transformações. A transformação Yeo-Johnson trouxe pequenos ganhos no R^2 do XGBoost, que aumentou de 0,33 para 0,40. No entanto, a maior parte dos ganhos foi observada em conjuntos sem transformações, indicando que esses algoritmos também são relativamente robustos às distribuições originais.

Por outro lado, para os modelos lineares (Regressão Linear, Ridge, Lasso e Elastic Net) as transformações tiveram um papel mais significativo. Esses métodos, devido à sua natureza restrita à linearidade, dependem de dados bem comportados para alcançar um desempenho competitivo. A transformação Raiz Quadrada melhorou marginalmente o desempenho em alguns casos, mas ainda insuficiente para torná-los comparáveis aos modelos não lineares. Por exemplo, o Ridge viu uma redução no R^2 de 0,40 sem transformações para 0,35 após a aplicação da Raiz Quadrada, enquanto a transformação Yeo-Johnson resultou em um R^2 ainda menor de 0,32, mostrando que essas alterações nem sempre são benéficas.

O impacto das transformações foi mais evidente no MLP e no SVR, que apresentaram R^2 negativos em todas as condições. Esses modelos não conseguiram se beneficiar das transformações, possivelmente devido à inadequação de seus ajustes de hiperparâmetros e arquiteturas para este conjunto de dados. Por exemplo, o SVR manteve R^2 negativo em torno de -0,10 a -0,12, independentemente da transformação aplicada.

Uma dificuldade notada foi a escolha da transformação ideal para cada modelo. Enquanto a Raiz Quadrada apresentou resultados mais consistentes para variáveis com assimetria moderada, a transformação Yeo-Johnson teve dificuldades em cenários com distribuições mais complexas, como aquelas bimodais. Isso reforça a necessidade de uma análise cuidadosa da distribuição dos dados antes de aplicar transformações, especialmente para modelos que dependem fortemente de uma representação adequada das variáveis.

4.5 Análise de Importância das Variáveis

A análise de importância das variáveis (Figura 4.13) revelou que variáveis como `Quantitativo_Total`, `Tipologia_Benfeitoria` e `UF` possuem forte impacto no custo unitário. Estas variáveis estão associadas às características físicas e geográficas das obras, que influenciam diretamente os custos de materiais e logística.

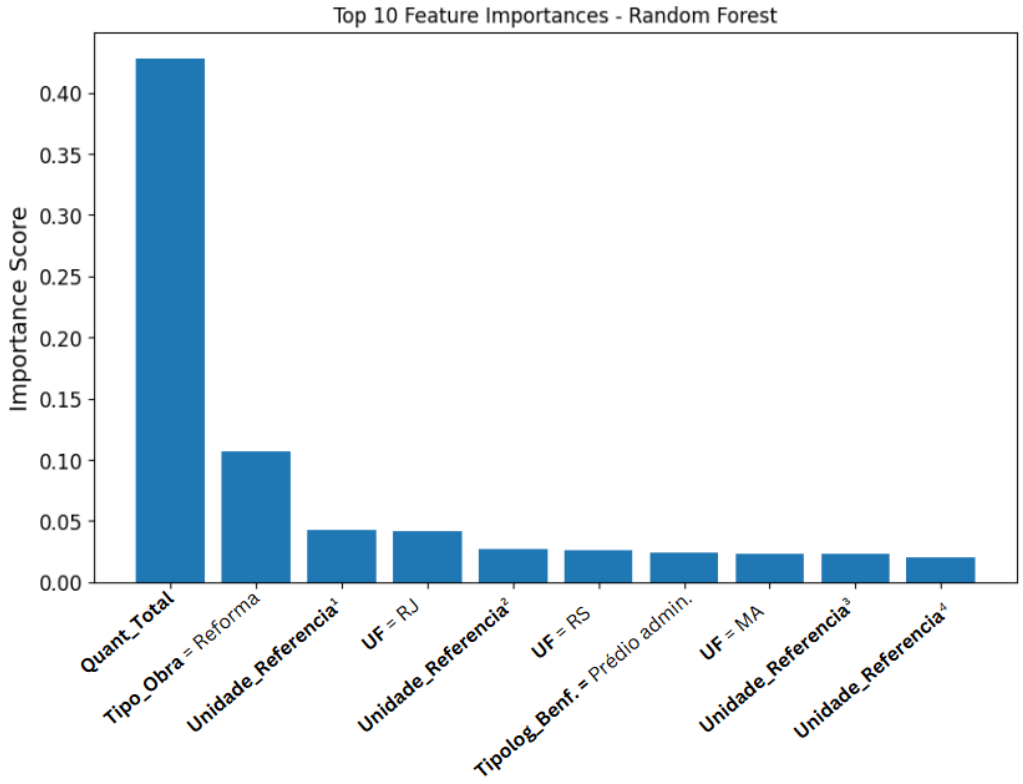

FIGURA 4.13 – Importância das Variáveis no Modelo Random Forest

A partir do gráfico, observa-se que a variável mais importante no modelo Random Forest é a `Quantitativo_Total`, com uma contribuição dominante superior a 40% no escore de importância. Essa relação é esperada, pois o quantitativo reflete diretamente a escala da obra, influenciando diretamente os recursos necessários, como materiais e mão de obra.

A segunda variável mais relevante é o `Tipo_Obra`. Embora sua importância seja bem menor que a de `Quantitativo_Total`, ela desempenha um papel significativo na previsão, já que reformas e construções envolvem processos, escopos e custos bastante distintos. Essa diferença entre os tipos de obras influencia diretamente os parâmetros financeiros de cada projeto.

[1] Unidade_Referencia_R$ / m² de área construída, considerando todos os pavimentos da edificação
[2] Unidade_Referencia_R$ / m³ de asfalto
[3] Unidade_Referencia_R$ / m linear da linha de postes de iluminação
[4] Unidade_Referencia_R$ / m linear de rede

Variáveis como `Unidade_Referencia` apresentaram desafios na padronização, devido à heterogeneidade entre as unidades cadastradas no banco de dados.

Unidades federativas como RJ, RS e MA também aparecem no gráfico, sugerindo que os fatores regionais têm impacto considerável nos custos. Diferenças nos preços de mão de obra, insumos e logística entre regiões explicam por que essas variáveis influenciam o modelo. Essas condições locais frequentemente refletem características econômicas e estruturais únicas de cada estado.

Por fim, a "`Tipologia_Benfeitoria` = Prédio Administrativo" aparece como um fator moderadamente importante, evidenciando que o tipo de edificação também contribui para a estimativa, embora com menor peso em comparação a variáveis como `Quantitativo_Total` ou `Tipo_Obra`. Isso demonstra que, embora a especificidade de cada projeto seja relevante, ela é secundária frente às variáveis gerais que definem escala e contexto.

Em resumo, o gráfico mostra que o modelo Random Forest dá maior peso a variáveis relacionadas à escala da obra e características gerais do projeto, como tipo e localização. Essas escolhas refletem fatores fundamentais na engenharia civil, enquanto variáveis mais específicas têm impacto menor, mas ainda contribuem para aumentar a precisão do modelo.

4.6 Resultados do Melhor Modelo: Random Forest

Os resultados das dez primeiras e dez últimas linhas das previsões resultantes do Random Forest são apresentados nas Figuras 4.14 e 4.15, respectivamente. A tabela destaca valores reais, previstos e as diferenças calculadas para diversos projetos contidos na amostra de teste.

	Titulo_Projeto	UF	Tipo_Obra	Tipologia_Benfeitoria	Quantitativo_Total	Custo_Total_Atualizado	Custo_Unitario	Unidade_Referencia	Valor Real	Valor Previsto	Erro_Percentual
168	INSTALAÇÃO DE TOMADAS PARA EQUIPAMENTOS	DF	Reforma	REDE ELÉTRICA DE BAIXA TENSÃO	400.00	9278.54	23.20	R$ / m linear de rede	23.2	1731.8816	7365.01%
140	REVITALIZAÇÃO DA SINALIZAÇÃO HORIZONTAL DO AER...	RS	Reforma	SINALIZAÇÃO HORIZONTAL	64000.00	362293.98	5.66	R$ / m² de área de pátio e/ou pista	5.66	414.8312	7229.17%
18	PROJETO DE RECUPERAÇÃO DA REDE DE ESGOTO DA VI...	MA	Reforma	REDE DE COLETA DE ESGOTO OU DRENAGEM	3322.92	1135839.29	341.82	R$ / m linear de rede	341.82	2998.8628	777.32%
114	ADEQUAÇÃO ESTRUTURAL E DA COBERTURA DO PNR E-0...	RS	Reforma	PRÉDIO RESIDENCIAL	157.16	178956.55	1138.69	R$ / m² de área construída, considerando todos...	1138.69	4502.3507	295.40%
150	CONSTRUÇÃO DO NOVO RESERVATÓRIO D'ÁGUA PARA SU...	RN	Construção	REDE DE ABASTECIMENTO DE ÁGUA	1552.00	779717.60	502.40	R$ / m linear de rede	502.4	1908.0058	279.78%
118	RECUPERAÇÃO DAS LINHAS DE HANGARETES A, B E D,...	RN	Reforma	COBERTURA SEM FECHAMENTO LATERAL	11489.91	3685953.84	320.80	R$ / m² de projeção da cobertura	320.8	916.405	185.66%
121	MANUTENÇÃO DO TELHADO DO DEPÓSITO DE INFRAESTR...	AM	Reforma	TELHADO OU COBERTURA	150.99	147003.73	973.60	R$ / m² de área de projeção	973.6	2568.3603	163.80%
67	COMGAP REFORMA DO 5º ANDAR	SP	Reforma	PRÉDIO ADMINISTRATIVO	1348.52	1168798.82	866.73	R$ / m² de área construída, considerando todos...	866.73	1869.3228	115.68%
152	PROJETO DO SISTEMA DE PROTEÇÃO CONTRA DESCARGA...	MT	Reforma	PRÉDIO ADMINISTRATIVO	128.00	243803.68	1904.72	R$ / m² de área construída, considerando todos...	1904.72	4087.3951	114.59%
162	PROJETO DE REFORMA DO HANGAR H-002 - GALC - BAAF	RJ	Reforma	GALPÃO COM FECHAMENTO LATERAL	2328.00	1630117.53	700.22	R$ / m² de projeção da cobertura do galpão	700.22	1306.3012	86.56%

FIGURA 4.14 – Resultados encontrados para o melhor modelo (Random Forest) - 10 primeiras linhas.

Esses resultados mostram uma análise interessante sobre a precisão das estimativas de custos, com distinção clara entre os casos de maiores e menores erros percentuais. Para os casos com maiores erros, observa-se que o modelo enfrentou dificuldades significati-

vas em obras mais específicas ou atípicas. Por exemplo, projetos como ”INSTALAÇÃO DE TOMADAS PARA EQUIPAMENTOS” (de rede elétrica) e ”REVITALIZAÇÃO DA SINALIZAÇÃO HORIZONTAL DO AEROPORTO” (de sinalização horizontal) apresentaram erros percentuais acima de 7000%.

Algumas tipologias, como ”REDE ELÉTRICA DE BAIXA TENSÃO”, possuem pouquíssimos registros na base de dados (apenas 1 observação). Isso impacta diretamente a capacidade do modelo Random Forest de generalizar para esses casos, já que ele depende de uma quantidade representativa de dados para aprender padrões robustos.

	Titulo_Projeto	UF	Tipo_Obra	Tipologia_Benfeitoria	Quantitativo_Total	Custo_Total_Atualizado	Custo_Unitario	Unidade_Referencia	Valor Real	Valor Previsto	Erro_Percentual
116	ADEQUAÇÃO DO ESQUADRÃO HÓSPEDE AO GOP, VISANDO...	RN	Reforma	PRÉDIO ADMINISTRATIVO	772.65	1110377.28	1437.10	R$ / m² de área construída, considerando todos...	1437.1	1819.8148	26.63%
68	DCTA Ampliação do Pátio	SP	Construção	PAVIMENTO RÍGIDO	4037.20	8392105.32	2078.69	R$ / m² de concreto	2078.69	2489.8905	19.78%
80	PANT 2 Blocos de PNR Graduados CHAS	RN	Construção	PRÉDIO RESIDENCIAL	8321.41	46207382.63	5552.83	R$ / m² de área construída, considerando todos...	5552.83	4505.3721	18.86%
100	CONSTRUÇÃO DA SALA B1 DE ESTUDO DO ITA	SP	Construção	PRÉDIO ADMINISTRATIVO	82.19	739626.27	8998.98	R$ / m² de área construída, considerando todos...	8998.98	7467.091	17.02%
9	CONSTRUÇÃO DA UNIDADE DE TERAPIA INTENSIVA DO ...	AM	Construção	PRÉDIO HOSPITALAR	602.25	3961472.82	6577.79	R$ / m² de área construída, considerando todos...	6577.79	5768.5463	12.30%
78	MUSAL Reforma SE	RJ	Reforma	SUBESTAÇÃO ELÉTRICA	600.00	1509177.44	2515.30	R$ / kVA	2515.3	2250.198	10.54%
70	DCTA PNR Graduados	SP	Construção	PRÉDIO RESIDENCIAL	6913.71	32362085.80	4680.86	R$ / m² de área construída, considerando todos...	4680.86	5121.047	9.40%
16	Construção de arruamento entre os blocos A, B,...	DF	Construção	PAVIMENTO FLEXÍVEL	300.00	1768813.93	5896.05	R$ / m² de asfalto	5896.05	6051.8902	2.64%
142	Recuperação da rede de esgoto do Rancho da BANT	RN	Reforma	REDE DE COLETA DE ESGOTO OU DRENAGEM	365.00	491274.77	1345.96	R$ / m linear de rede	1345.96	1375.7683	2.21%
159	Projeto de Adequação da Antiga Estação de Oxig...	RJ	Reforma	GALPÃO COM FECHAMENTO LATERAL	69.00	402109.17	5827.67	R$ / m² de projeção da cobertura do galpão	5827.67	5787.8206	0.68%

FIGURA 4.15 – Resultados encontrados para o melhor modelo (Random Forest) - 10 últimas linhas.

Por outro lado, tipologias com maior frequência, como ”PRÉDIO ADMINISTRATIVO” (33 observações) e ”PRÉDIO RESIDENCIAL” (18 observações), apresentaram previsões mais precisas, com menores erros percentuais. Isso ocorre porque o modelo tem mais exemplos para identificar relações consistentes entre as variáveis independentes (como quantitativo e tipologia) e a variável dependente (custo unitário).

Além disso, o desequilíbrio na distribuição das observações por tipologia também reflete a dificuldade do modelo em lidar com dados raros. Tipologias sub-representadas, como ”PORTÃO DE HANGAR OU GALPÃO” e ”ESTAÇÃO DE TRATAMENTO DE ÁGUA OU ESGOTO”, têm pouca ou nenhuma oportunidade de contribuir significativamente para o treinamento do modelo, resultando em previsões com maior probabilidade de erro.

Essa análise evidencia os pontos fortes do Random Forest em capturar padrões complexos em dados abundantes e diversos, aproveitando sua estrutura de múltiplas árvores para reduzir o *overfitting* e melhorar a robustez das previsões. O modelo é particularmente eficaz em situações onde há uma representação equilibrada das diferentes tipologias de obras, permitindo que ele aprenda as nuances e variações dentro dos dados.

Por outro lado, a principal fraqueza do Random Forest neste contexto é sua dependência de um conjunto de dados representativo para todas as classes de interesse. A presença de tipologias com poucas observações cria um desafio, pois o modelo não possui informações suficientes para aprender padrões confiáveis nessas categorias. Além disso, como o Random Forest não extrapola bem para além dos dados observados, sua capacidade de

prever custos para obras atípicas ou inovadoras é limitada.

Em suma, o Random Forest demonstra excelente desempenho em prever custos unitários para obras comuns e bem representadas no conjunto de dados, aproveitando sua capacidade de modelar relações não lineares e interações entre variáveis. No entanto, sua eficácia diminui em projetos específicos ou raros, destacando a necessidade de abordar o desequilíbrio e a representatividade dos dados para melhorar as previsões nesses casos.

4.7 Próximos Passos

Com base nas limitações identificadas e nas possibilidades de melhoria, sugerem-se os seguintes próximos passos para o desenvolvimento e aprimoramento das práticas e ferramentas de estimativa de custos:

1. **Ampliação e Enriquecimento da Base de Dados**
   É fundamental ampliar a base de dados utilizada, incorporando mais registros de obras e contemplando variáveis adicionais que possam influenciar significativamente as estimativas de custos. A inclusão de informações que diferenciem cenários antes e depois da pandemia, por exemplo, pode ajudar a capturar os efeitos de eventos externos no comportamento dos custos.

2. **Implementação de Interface Online com API**
   Propõe-se a criação de um serviço online acessível por meio de uma API (*Application Programming*) Interface integrada a uma interface intuitiva, permitindo que engenheiros insiram dados dos projetos e obtenham estimativas em tempo real. Essa solução agilizaria o processo de planejamento, eliminando etapas manuais e garantindo maior acessibilidade. Além disso, a ferramenta poderia evoluir continuamente com base em novos dados e modelos, aumentando sua precisão e adaptabilidade às necessidades específicas da FAB.

3. **Tratamento Avançado de Dados**
   Considerando a heterogeneidade dos dados, sugere-se um tratamento mais refinado, com estratégias específicas para lidar com inconsistências, *outliers* e categorias subrepresentadas. Isso inclui a padronização de variáveis e a análise cuidadosa de *outliers* para melhorar a qualidade e a representatividade dos dados.

4. **Segmentação dos Métodos por Tipologia de Obra**
   Dado que diferentes tipologias de obras apresentam padrões de custo distintos, é recomendável segmentar os métodos preditivos de acordo com a tipologia. Isso pode melhorar a precisão ao permitir que os modelos sejam ajustados às características específicas de cada tipo de obra.

5. **Exploração de Abordagens Híbridas**
   Recomenda-se a exploração de técnicas híbridas que combinem a robustez de métodos estatísticos tradicionais com a flexibilidade dos algoritmos de aprendizado de máquina. Por exemplo, a utilização de Random Forest aliado a modelos especializados para cenários raros pode aumentar a robustez das estimativas, reduzindo desvios em casos extremos.

6. **Incorporação de Variáveis Contextuais**
   Para reduzir desvios nas estimativas, sugere-se a inclusão de variáveis que capturem melhor as nuances dos projetos, como indicadores econômicos regionais, sazonalidade, políticas públicas vigentes e fatores climáticos. Esses dados podem ajudar a ajustar as previsões às condições específicas de cada obra.

7. **Simplicidade como Diretriz**
   Apesar das melhorias sugeridas, é importante manter o foco na simplicidade das estimativas, garantindo que o modelo seja prático e acessível para os gestores de projetos. A simplicidade é crucial para a adoção efetiva da ferramenta em ambientes de planejamento e execução.

8. **Atenção quanto a Possíveis Erros em Cenários Futuros**
   Em um cenário de produção da solução, é importante considerar e mitigar potenciais erros ou vieses que possam surgir na utilização da ferramenta. Por exemplo, projetos de reforma podem introduzir variabilidade nos dados devido à dificuldade em padronizar as condições prévias das obras. Além disso, obras em regiões litorâneas podem trazer custos adicionais não previstos devido a fatores ambientais específicos, como corrosão e logística diferenciada. Outro ponto crítico é a atenção à unidade de referência utilizada, uma vez que erros nesse aspecto podem comprometer a precisão das estimativas. A aplicação deve incluir alertas ou instruções claras para minimizar tais problemas, garantindo maior confiabilidade e eficiência.

Com esses próximos passos, espera-se avançar ainda mais no desenvolvimento de uma ferramenta computacional concisa e eficiente, contribuindo para a evolução das práticas de estimativa de custos na engenharia civil e para a gestão eficiente de projetos de infraestrutura.

5 Conclusão

Este trabalho destacou a relevância e a aplicação prática de diferentes técnicas de Aprendizado de Máquina para a estimativa de custos em obras de engenharia civil, com especial atenção às necessidades da Força Aérea Brasileira. A análise comparativa revelou que modelos como Random Forest e CatBoost apresentaram desempenho superior, evidenciando sua capacidade de capturar relações complexas e fornecer estimativas mais precisas.

Além dos avanços apresentados, o estudo aponta para a necessidade de maior integração entre métodos preditivos e as práticas de planejamento e gestão de obras, especialmente em contextos desafiadores como o brasileiro, caracterizado por grande variabilidade regional e complexidade administrativa. A adoção de técnicas de Aprendizado de Máquina representa não apenas uma inovação tecnológica, mas também uma ferramenta estratégica para melhorar a alocação de recursos e a transparência na execução de projetos públicos.

A ferramenta desenvolvida oferece potencial para aplicação prática, permitindo maior previsibilidade e confiabilidade em estimativas realizadas nas etapas iniciais de planejamento.

Em síntese, este trabalho não apenas avança na aplicação de técnicas modernas à engenharia civil, mas também fornece um ponto de partida sólido para estudos futuros e implementação prática em projetos de infraestrutura. O uso de aprendizado de máquina, aliado a dados históricos robustos, configura-se como uma abordagem promissora para atender às crescentes demandas por precisão e eficiência no setor de construção civil.

Referências

ADAMS, J. M. Quantitative modelling methods for the incorporation of uncertainty into construction project estimates. **Construction Management and Economics**, v. 24, n. 5, p. 483–493, 2006.

AGRAWAL, R. **Evaluation Metrics for Your Regression Model**. [*S.l.*], 2024. Available at: https://www.analyticsvidhya.com/blog/2021/06/evaluation-metrics-for-your-regression-model/.

ANALYSTS, I. S. of P. **Parametric Estimating Handbook**. Fourth edition. Vienna, VA, 2008. Disponível em: `https://www.ispa-cost.org`. Acesso em: 21 jul. 2024.

ANDRADE, A. C. d.; SOUZA, U. E. L. d. Críticas ao processo orçamentário tradicional e recomendações para confecção de um orçamento integrado ao processo de produção de um empreendimento. *In*: **Anais do III Simpósio Brasileiro de Gestão e Economia da Construção e Encontro Latino-Americano de Gestão e Economia da Construção**. **Proceedings** [...]. São Carlos: UFSCAR, 2003. p. 1–11.

BARROS, L. B. **Aplicação de redes neurais artificiais no contexto de estimativa de custos de construção de rodovias**. 106 p. Dissertation (Dissertação de Mestrado em Estruturas e Construção Civil) — Universidade de Brasília, Brasília, 2019.

BELTRÃO, L. M. P.; CARVALHO, M. T. M.; BLUMENSCHEIN, R. N.; PAIVA, Á. T.; FREITAS, M. V. R. Modelos para estimativa de custos com o uso de regressão linear: modelagem com obras penitenciárias. **Ambiente Construído**, v. 22, n. 3, p. 193–211, 2022.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. Berlin, Heidelberg: Springer, 2006. ISBN 9780387310732.

BRASIL. **Lei nº 14.133, de 1ª de abril de 2021. Dispõe sobre Licitações e Contratos Administrativos**. Brasília: Presidência da República, 2021. Disponível em: `http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14133.htm`. Acesso em: 6 out. 2024.

BREIMAN, L. Random forests. **Machine Learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BROWNLEE, J. **How to Develop Multi-Output Regression Models with Python**. 2023. Acesso em: 19 jun. 2024. Available at: https://machinelearningmastery.com/multi-output-regression-models-with-python/.

CHAYA. Random forest regression. **Level Up Coding**, acesso em: 14 nov. 2024., 2020. Available at: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. p. 785–794, 2016.

DOCUMENTATION, F. engine. **YeoJohnsonTransformer — 1.8.2**. [*S.l.*], 2024. Disponível em: `https://feature-engine.trainindata.com/en/1.8.x/user_guide/transformation/YeoJohnsonTransformer.html`. Acesso em: 10 out. 2024.

DYSERT, L. R. Developing a parametric model for estimating process control costs. **Cost Engineering**, v. 43, n. 2, p. 31–34, available at: `https://www.costengineering.eu/images/papers/Developing_a_Parametric_Model_for_Estimating_Process_Control_Costs.pdf`, February 2001.

FONSECA, F. C. R. **Proposta de um método probabilístico de estimativa de custos de construção**. Dissertation (Dissertação de Mestrado em Estruturas, Geotecnia, Construção Civil) — Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2013. Disponível em: `http://www.bdtd.uerj.br/handle/1/11592`.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, v. 29, n. 5, p. 1189–1232, 2001.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. Second edition. Sebastopol, CA: O'Reilly Media, 2019. Disponível em: `https://www.oreilly.com/`.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2016. ISBN 9780262035613.

GÜNAYDIN, H. M.; DOĞAN, S. Z. A neural network approach for early cost estimation of structural systems of buildings. **International Journal of Project Management**, v. 22, n. 7, p. 595–602, 2004.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. [*S.l.*]: Springer Science & Business Media, 2009.

HEGAZY, T.; AYED, A. Neural network model for parametric cost estimation of highway projects. **Journal of Construction Engineering and Management**, v. 124, n. 3, p. 210–218, 1998.

(IBRAOP), I. B. de Auditoria de O. P. **Manual de Auditoria de Obras Públicas**. Brasília: IBRAOP, 2016. Disponível em: `https://www.ibraop.org.br/wp-content/uploads/2016/12/Manual-de-Auditoria-de-Obras-Publicas.pdf`. Acesso em: 5 out. 2024.

INTERNATIONAL, A. **Recommended Practice 17R-97: Cost Estimate Classification System**. Morgantown, WV, 1997.

KATO, A. P. S.; LOPES, R. R.; PEREIRA, T. A.; AQUINO, E. F. Estudo de dados paramétricos para estimativa de custos em reservatórios de água. **Revista de Engenharia Hidráulica**, v. 29, n. 1, p. 45–57, 2022.

KIM, G. H.; AN, S. H.; KANG, K. I. Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. **Building and Environment**, v. 39, n. 10, p. 1235–1242, 2004.

LIBRELOTTO, L. I.; AVILA, A. V.; LOPES, C. L. **Orcamento de obras: construcao civil**. [*S.l.*], 2003. Disponível em: `http://pet.ecv.ufsc.br/site/downloads/apoio_didatico/ECV5307-OrÃğamento.Pdf`. Acesso em: 22 jul. 2024.

LOWE, D. J.; EMSLEY, M. W.; HARDING, A. Predicting construction cost using multiple regression techniques. **Journal of Construction Engineering and Management**, v. 132, n. 7, p. 750–758, 2006.

MAUÉS, F. C. A.; MELO, K. P.; LEÃO, C. B. O.; SERRA, S. M. B. Estimativa de custos paramétricos de construção de edifícios usando modelo de regressão linear. **Gestão & Tecnologia de Projetos**, São Carlos, v. 17, n. 2, 2022.

MEYER, E. R.; BURNS, T. J. Facility parametric cost estimating. *In*: **American Association of Cost Engineers (AACE) Transaction. 43rd Annual Meeting. Proceedings**. **Proceedings** [...]. Denver, USA: [*s.n.*], 1999.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA: MIT Press, 2012. ISBN 9780262018029.

NASA. **Cost Estimating Handbook**. Version 4.0. [*S.l.*], 2015. Disponível em: `https://www.nasa.gov/sites/default/files/files/01_CEH_Main_Body_02_27_15.pdf`. Acesso em: 19 set. 2024.

NumPy Community. **NumPy Documentation**. 2024. `https://numpy.org/doc/`. Acesso em: 9 jun. 2024.

OTERO, J. A. Uso de modelos paramétricos em estimativas de custo para construção de edifícios. *In*: **Encontro Nacional de Engenharia de Produção (ENEGEP 18)**. **Proceedings** [...]. Niterói, RJ: [*s.n.*], 1998. Disponível em: `http://www.abepro.org.br/biblioteca/ENEGEP1998_ART309.pdf`.

Pandas Development Team. **Pandas Documentation. Version 2.2.3**. [*S.l.*], 2024. Disponível em: `https://pandas.pydata.org/pandas-docs/stable/`. Acesso em: 28 jun. 2024.

PROKHORENKOVA, L.; GUSEV, G.; VOROBEV, A.; DOROGUSH, A. V.; GULIN, A. Catboost: unbiased boosting with categorical features. **ArXiv Preprint ArXiv:1706.09516v5**, disponível em: `https://arxiv.org/abs/1706.09516v5`. Acesso em: 22 nov. 2024., 2019.

RASCHKA, S. **MultilayerPerceptron: A simple multilayer neural network**. [*S.l.*], 2018. Available at: `https://rasbt.github.io/mlxtend/user_guide/classifier/MultiLayerPerceptron/`. Acesso em: 18 nov. 2024.

RODRIGUES, P. T. **Estimativa paramétrica de custo de obras com o uso de redes neurais artificiais**. 50 f. p. Trabalho de Conclusão de Curso (Graduação) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 2020.

SCIKIT-LEARN TEAM. **scikit-learn: Machine Learning in Python**. [*S.l.*], 2021. Available at: https://scikit-learn.org/.

SONMEZ, R. Parametric range estimating of building costs using regression models and bootstrap. **Journal of Construction Engineering and Management**, v. 134, n. 12, p. 1011–1016, 2008.

(TCU), T. de Contas da U. **Orientações para elaboração de planilhas orçamentárias de obras públicas**. Brasília, 2014. Disponível em: `https://www.tcu.gov.br`. Acesso em: 1 mar. 2024.

(TCU), T. de Contas da U. **Licitações e Contratos: orientações e jurisprudência do TCU**. 5ª edição. ed. Brasília, 2024. Disponível em: `https://www.tcu.gov.br`. Acesso em: 1 mar. 2024.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 58, n. 1, p. 267–288, 1996.

Anexo A - Rotinas de Programação

A.1 Código em Python utilizado no estudo

Listing A.1 – Análise de Dados em Python

```
# Python 3.5 is required
import sys
assert sys.version_info >= (3, 5)

# Scikit-Learn 0.20 is required
import sklearn
assert sklearn.__version__ >= "0.20"

# Common imports
import numpy as np
import os

# Para gerar grficos bonitos
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)

# Onde salvar as figuras
PROJECT_ROOT_DIR = "/content/sample_data"
CHAPTER_ID = "end_to_end_project"
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, "images", CHAPTER_ID)
os.makedirs(IMAGES_PATH, exist_ok=True)

def save_fig(fig_id, tight_layout=True, fig_extension="png", resolution=300):
    path = os.path.join(IMAGES_PATH, fig_id + "." + fig_extension)
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)
```

```
-----------------------------------------------------------------------------------------------



from google.colab import drive
drive.mount('/content/drive')

-----------------------------------------------------------------------------------------------



import pandas as pd

# Funo para carregar o arquivo Excel local
def load_housing_data(housing_path="/content/Base.xlsx"):
    return pd.read_excel(housing_path)

# Realize o upload do arquivo Base.xlsx para o ambiente do Colab
from google.colab import files
#uploaded = files.upload()  # Faz o upload do arquivo

# Carrega os dados
#housing = load_housing_data()
housing = pd.read_csv("/content/drive/MyDrive/TG/TG/Base  - Tabela principal.csv")

colunas_converter = ["Custo_Total_Atualizado","Quantitativo_Total","Custo_Unitario"]

for coluna in colunas_converter:
  housing[coluna] = housing[coluna].str.replace(",",".").astype(float)

housing.drop(columns = ["Custo_Total_Atualizado"], inplace = True)
housing

-----------------------------------------------------------------------------------------------



# Visualizando a estrutura de dados
!pip install ydata-profiling
from ydata_profiling import ProfileReport

# Gere o relatrio
profile = ProfileReport(housing, title="Housing Data Report", explorative=True)

# Exiba o relatrio diretamente no Jupyter Notebook
profile.to_notebook_iframe()



import pandas as pd
```

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import PowerTransformer

-------------------------------------------------------------------------------------------



df = housing.copy()
# Transformar colunas categricas em variveis dummy
df = pd.get_dummies(df, columns=['UF', 'Tipo_Obra', 'Tipologia_Benfeitoria', '
    Unidade_Referencia'], drop_first=True)

# 1. Remover outliers no target (Custo_Unitario) acima do percentil 0.95
upper_limit = df['Custo_Unitario'].quantile(0.95)
df = df[df['Custo_Unitario'] <= upper_limit]

upper_limit = df['Quantitativo_Total'].quantile(0.998)
#df = df[df['Quantitativo_Total'] <= upper_limit]

# Separar variveis dependentes e independentes
X = df.drop(columns=['Custo_Unitario', 'Titulo_Projeto'])
y = df['Custo_Unitario']

# Dividir os dados em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state
    =42)

# Identificar colunas numricas e categricas
num_cols = X_train.select_dtypes(include=['float64', 'int']).columns.tolist()
cat_cols = X_train.select_dtypes(include=['object']).columns.tolist()

colunas_train = X_train.columns
index_train = X_train.index

colunas_test = X_test.columns
index_test = X_test.index
# 2. Criar transformaes para variveis categricas e numricas
# Transformao de normalizao das numricas
# Instanciar o StandardScaler
scaler = StandardScaler()

# Ajustar o scaler e transformar X_train
```

```
#X_train = scaler.fit_transform(X_train)
#X_test = scaler.transform(X_test)

# Se desejar manter X_train como DataFrame
#X_train = pd.DataFrame(X_train, columns=colunas_train, index= index_train)
#X_test = pd.DataFrame(X_test, columns=colunas_test, index= index_test)

-------------------------------------------------------------------------------------------



from sklearn.metrics import mean_squared_error, r2_score

# Regresso linear sem transformao
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Avaliao do modelo sem transformao
print("Modelo sem transformao:")
print("MSE:", mean_squared_error(y_test, y_pred))
print("R:", r2_score(y_test, y_pred))

# Plot sem transformao
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel("Valores Reais (y_test)")
plt.ylabel("Valores Previstos (y_pred)")
plt.title("Modelo sem Transformao: y_pred vs y_test")
plt.show()

# Testando transformaes para a varivel alvo
transformations = {
    "Raiz Quadrada": np.sqrt,
    "Yeo-Johnson": PowerTransformer(method='yeo-johnson')
}

for name, transformer in transformations.items():
    # Transformao no y
    if name == "Yeo-Johnson":
        y_train_transformed = transformer.fit_transform(y_train.values.reshape(-1, 1))
    .flatten()
    else:
        y_train_transformed = transformer(y_train)

    # Treina o modelo com a varivel alvo transformada
    model.fit(X_train, y_train_transformed)
```

```
    # Predio no conjunto de teste
    y_pred_transformed = model.predict(X_test)

    # Aplicar inversa da transformao para trazer o y_pred para a escala original
    if name == "Yeo-Johnson":
        y_pred_original = transformer.inverse_transform(y_pred_transformed.reshape(-1,
     1)).flatten()
    elif name == "Raiz Quadrada":
        y_pred_original = y_pred_transformed ** 2

    # Avaliao do modelo com transformao
    print(f"\nModelo com transformao {name}:")
    print("MSE:", mean_squared_error(y_test, y_pred_original))
    print("R:", r2_score(y_test, y_pred_original))

    # Plot com transformao
    plt.figure(figsize=(8, 6))
    plt.scatter(y_test, y_pred_original, color='green' if name == "Raiz Quadrada" else
     'orange')
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
    plt.xlabel("Valores Reais (y_test)")
    plt.ylabel("Valores Previstos (y_pred)")
    plt.title(f"Modelo com Transformao {name}: y_pred vs y_test")
    plt.show()


-------------------------------------------------------------------------------------------



from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn.preprocessing import PowerTransformer
import numpy as np
import matplotlib.pyplot as plt
!pip install catboost
from catboost import CatBoostRegressor



# Modelos a serem testados
models = {
    "Ridge Regression": Ridge(alpha=1.0),
    "Lasso Regression": Lasso(alpha=0.1),
    "ElasticNet": ElasticNet(alpha=0.1, l1_ratio=0.5),
```

```
    "Random Forest": RandomForestRegressor(n_estimators=100, random_state=42),
    "XGBoost Regressor": XGBRegressor(n_estimators=100, random_state=42),
    "MLP Regressor": MLPRegressor(
        hidden_layer_sizes=(10, 5),  # Camadas menores para evitar overfitting
        activation='tanh',  # Funo de ativao tanh
        solver='adam',  # Otimizador Adam
        alpha=0.01,  # Regularizao L2 para evitar overfitting
        learning_rate='adaptive',  # Adapta a taxa de aprendizado conforme o modelo
    converge
        max_iter=500,  # Menor nmero de iteraes devido ao early stopping
        early_stopping=True,  # Para automaticamente se no houver melhora
        random_state=42
    ),
    "Gradient Boosting": GradientBoostingRegressor(n_estimators=100, random_state=42),
    "Support Vector Regressor": SVR(kernel='rbf', C=1.0, epsilon=0.1),
    "CatBoost Regressor": CatBoostRegressor(
        iterations=1000,  # Aumenta o nmero de iteraes para ajuste fino
        learning_rate=0.01,  # Reduz a taxa de aprendizado para treinamento mais
    preciso
        depth=10,  # Aumenta a profundidade das rvores para capturar relaes complexas
        l2_leaf_reg=3,  # Regularizao para evitar overfitting
        loss_function='RMSE',  # Funo de perda RMSE para melhor adequao em regresso
        eval_metric='RMSE',  # Mtrica de avaliao durante o treinamento
        random_seed=42,
        od_type='Iter',  # Tipo de parada precoce baseado em iteraes
        od_wait=100,  # Nmero de iteraes sem melhoria antes de parar
        verbose=5000  # Exibe progresso a cada 500 iteraes
    ),
}

# Teste cada modelo
for model_name, model in models.items():
    # Configurar a figura para subplots
    fig, axes = plt.subplots(1, 3, figsize=(18, 5))
    fig.suptitle(f"{model_name}", fontsize=16)

    # Regresso sem transformao
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Avaliao do modelo sem transformao
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    axes[0].scatter(y_test, y_pred, color='blue')
    axes[0].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw
    =2)
    axes[0].set_xlabel("Valores Reais (y_test)")
```

```
axes[0].set_ylabel("Valores Previstos (y_pred)")
axes[0].set_title(f"Sem Transformao\nMSE: {mse:.2f}, R: {r2:.2f}")

# Testando transformaes para a varivel alvo
transformations = {
    "Raiz Quadrada": np.sqrt,
    "Yeo-Johnson": PowerTransformer(method='yeo-johnson')
}

for idx, (name, transformer) in enumerate(transformations.items(), start=1):
    # Transformao no y
    if name == "Yeo-Johnson":
        y_train_transformed = transformer.fit_transform(y_train.values.reshape(-1,
 1)).flatten()
    else:
        y_train_transformed = transformer(y_train)

    # Treinar o modelo com a varivel alvo transformada
    model.fit(X_train, y_train_transformed)

    # Predio no conjunto de teste
    y_pred_transformed = model.predict(X_test)

    # Aplicar inversa da transformao para trazer o y_pred para a escala original
    if name == "Yeo-Johnson":
        y_pred_original = transformer.inverse_transform(y_pred_transformed.reshape
(-1, 1)).flatten()
    elif name == "Raiz Quadrada":
        y_pred_original = y_pred_transformed ** 2

    # Avaliao do modelo com transformao
    mse_transformed = mean_squared_error(y_test, y_pred_original)
    r2_transformed = r2_score(y_test, y_pred_original)

    # Plot com transformao
    axes[idx].scatter(y_test, y_pred_original, color='green' if name == "Raiz
Quadrada" else 'orange')
    axes[idx].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k
--', lw=2)
    axes[idx].set_xlabel("Valores Reais (y_test)")
    axes[idx].set_ylabel("Valores Previstos (y_pred)")
    axes[idx].set_title(f"Transformao {name}\nMSE: {mse_transformed:.2f}, R: {
r2_transformed:.2f}")

# Mostrar todos os grficos para o modelo atual
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

```
-----------------------------------------------------------------------------------------------



# Treinar o modelo Random Forest novamente para garantir que temos as previses mais
    recentes
best_model = RandomForestRegressor(n_estimators=100, random_state=42)
best_model.fit(X_train, y_train)
y_pred_rf = best_model.predict(X_test)  # Previses da Random Forest

# Copiar o DataFrame original para preservar a estrutura inicial
df_with_predictions = housing.copy()

# Resetar o ndice para garantir a correspondncia entre os dados originais e os valores
     de teste
df_with_predictions.reset_index(drop=True, inplace=True)

# Adicionar as colunas de valores reais e previstos ao DataFrame original, mantendo
    apenas as amostras de teste
df_with_predictions['Valor Real'] = None
df_with_predictions['Valor Previsto'] = None

# Preencher com os valores reais e previstos nos ndices correspondentes ao conjunto de
     teste
X_test_index = X_test.index  # ndices das amostras de teste
df_with_predictions.loc[X_test_index, 'Valor Real'] = y_test.values
df_with_predictions.loc[X_test_index, 'Valor Previsto'] = y_pred_rf

# Exibir o DataFrame com valores reais e previstos, ordenado pelos valores reais
df_with_predictions_filtered = df_with_predictions[~df_with_predictions['Valor
    Previsto'].isna()]
df_with_predictions_filtered.sort_values(by="Valor Real", ascending=False, inplace=
    True)
df_with_predictions_filtered['Diferenca_valores'] = abs(df_with_predictions_filtered['
    Valor Real'] - df_with_predictions_filtered['Valor Previsto'])
# Mostrar as primeiras 29 linhas para visualizao
previsoes = df_with_predictions_filtered.head(10)
previsoes.sort_values(by = 'Diferenca_valores', ascending = False)

-----------------------------------------------------------------------------------------------



import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
```

```
# Ajuste o modelo Random Forest com os dados de treinamento
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Obtenha as importncias das features e selecione as 10 mais importantes
importances = rf_model.feature_importances_
feature_names = X_train.columns
indices = np.argsort(importances)[::-1][:10]  # Apenas os 10 mais importantes

# Criao do grfico de importncia das features
plt.figure(figsize=(10, 6))
plt.title("Top 10 Feature Importances - Random Forest")
plt.bar(range(10), importances[indices], align="center")
plt.xticks(range(10), feature_names[indices], rotation=45, ha='right')
plt.xlabel("Features")
plt.ylabel("Importance Score")
plt.tight_layout()
plt.show()
```

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. REGISTRO N°	4. N° DE PÁGINAS
TC	27 de novembro de 2024	DCTA/ITA/TC-135/2024	81

5. TÍTULO E SUBTÍTULO:

Análise de métodos e proposta de ferramenta computacional para estimativas paramétricas de obras de engenharia civil.

6. AUTOR(ES):

Talles Eduardo Camargo dos Santos

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica - ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Construção Civil; Estimativa de Custo; Aprendizado de Máquina; Parametrização; Modelos Preditivos

9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Construção civil; Estimativa de custos; Aprendizagem (inteligência artificial); Controle preditivo; Programas de computadores; Algoritmos; Engenharia civil; Computação.

10. APRESENTAÇÃO: **(X) Nacional ( ) Internacional**

ITA, São José dos Campos. Curso de Graduação em Engenharia Civil-Aeronáutica. Orientadora: Profa. Dra. Maryangela Geimba Lima; coorientador: Cap. Eng. Paulo de Tarso Machado Leite Soares. Publicado em 2024.

11. RESUMO:

Este trabalho aborda o desenvolvimento de métodos inovadores para estimativa de custos no setor de construão civil, com foco na aplicação de técnicas paramétricas e aprendizado de máquina. Reconhecendo as limitações dos métodos tradicionais, como os baseados no Custo Unitário Básico (CUB), propõe-se a criação de uma ferramenta computacional para prever custos com maior precisão, mesmo em estágios iniciais dos projetos, quando os detalhes são escassos. A metodologia integra dados históricos de 204 obras, disponibilizados pelo Centro de Estudos e Pesquisa de Engenharia da Força Aérea Brasileira (CEPE), com algoritmos avançados de regressão, como Elastic Net, Support Vector Regressor e técnicas baseadas em árvores de decisão (Random Forest e XGBoost). Os melhores algoritmos se apresentaram como ajustes moderados para o contexto, atingindo coeficientes de determinação R2 próximos de 0,5. Os resultados destacam a evolução de modelos baseados em aprendizado de máquina na captura de interações complexas entre variáveis do projeto, contribuindo para menor risco financeiro e maior eficiência no gerenciamento de recursos. Potenciais melhorias e limitações foram identificadas e discutidas para que a futura implementação possa ser otimizada.

12. GRAU DE SIGILO:

(X) OSTENSIVO ( ) RESERVADO ( ) SECRETO