NOVA IMS Information Management School

Mestrado em

Gestão de Informação

MGI

Potencial De Uso De Técnicas De Aprendizado De Máquina Para Estimativa De Prazos Em Projetos

Deliane Amaro Duarte

Dissertação

apresentada(o) como requisito parcial para obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

POTENCIAL DE USO DE TÉCNICAS DE MACHINE LEARNING PARA ESTIMATIVA DE PRAZOS EM PROJETOS

por

Deliane Amaro Duarte

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Gestão de Informação, com especialização em Gestão do Conhecimento e Business Intelligence.

Orientador(a): Prof. Vítor Santos

Junho 2024

DECLARAÇÃO DE INTEGRIDADE

Declaro ter realizado o presente trabalho académico com integridade. Confirmo que não recorri à prática de plágio ou de qualquer outra forma de utilização indevida de informação ou de falsificação de resultados durante o processo de elaboração deste trabalho. Declaro ainda que tenho conhecimento das Regras de Conduta e do Código de Honra da NOVA Information Management School.

Deliane Amaro Duarte

Brasil, 2024

DEDICATÓRIA

Dedico este trabalho à minha família, amigos e
à todos que de alguma forma contribuíram
para que esta entrega fosse possível.

AGRADECIMENTOS

Agradeço a todas as pessoas que estiveram ao meu lado durante este desafio.

À minha mãe, minha fonte constante de coragem, por acreditar em mim mesmo quando sequer eu mesma consigo.

Ao meu marido, que diariamente é meu alicerce, me protege, cuida e sonha ao meu lado, mesmo quando tudo parece impossível.

Ao meu irmão, cuja leveza e coragem me movem e inspiram a sonhar cada vez mais.

À minha cunhada, por me inspirar e ensinar sobre apoio e coragem desmedida.

Ao meu orientador, Professor Vítor Santos, pelo seu talento incomparável de tornar desafios árduos em jornadas gratificantes e prazerosas.

Às minhas amigas Lorena e Liana, que com todo seu carinho, escuta e compreensão tornaram este processo mais leve.

Ao meu amigo João Lucas, cujo apoio é essencial para a continuidade da minha busca por novos desafios e crescimento contínuo.

Aos amigos que ganhei ao longo desta jornada — Dennis, Emna, Faustino, Neydu, Rebeca e Yousef — agradeço por pacientemente compartilharem não apenas técnicas, mas também valores preciosos que levarei comigo por toda a vida.

RESUMO

Complexidade, prazos apertados e limites orçamentários caracterizam a maior parte dos projetos atualmente, resultando em riscos que podem prejudicar significativamente o sucesso, levando a atrasos, desentendimentos e perdas financeiras. Estimativas mais assertivas se tornam cada vez mais importantes para uma gestão eficaz do projeto e mitigação de riscos, pois permite que as partes interessadas tomem decisões informadas e aloquem recursos de forma eficiente. Os métodos tradicionais de estimativa, como a opinião de especialistas e a estimativa paramétrica, têm sido usados há décadas, mas muitas vezes produzem resultados imprecisos devido à sua dependência da experiência humana e de dados históricos. Recentemente, os frameworks ágeis, como Scrum e Kanban, e a metodologia de Planejamento do Trabalho Avançado (AWP) foram analisados para determinar qual abordagem oferece maior flexibilidade e eficácia em ambientes de projeto dinâmicos e complexos. Este estudo explora como os avanços na aprendizagem de máquina e na análise de dados podem se integrar a esses frameworks para otimizar as estimativas de cronograma. Alguns métodos de aprendizado de máquina demonstraram potencial em uma variedade de disciplinas devido à sua capacidade de aprender com os dados e antecipar resultados com precisão. Para abordar esse problema, este estudo propõe a implementação de modelos de aprendizado de máquina como uma ferramenta auxiliar no processo de estimativa de prazos. O método empregado inclui o uso de validação cruzada com múltiplas dobras (k-fold cross-validation) para garantir a robustez e a generalização do modelo. Foram utilizados critérios de validação baseados em métricas como MSE (Erro Quadrático Médio), MAE (Erro Absoluto Médio) e R^2 (Coeficiente de Determinação), que permitiram avaliar a precisão e a confiabilidade das previsões geradas pelos modelos.

PALAVRAS-CHAVE

Gerenciamento de projetos; Aprendizado de Máquina; Estimativas; Estimativas de projeto; Frameworks Ágeis; AWP.

ABSTRACT

Complexity, tight deadlines, and budget constraints characterize most projects today, resulting in risks that can significantly undermine success, leading to delays, misunderstandings, and financial losses. More accurate estimates are increasingly important for effective project management and risk mitigation, as they allow stakeholders to make informed decisions and allocate resources efficiently. Traditional estimation methods, such as expert opinion and parametric estimation, have been used for decades but often produce inaccurate results due to their reliance on human experience and historical data. Recently, agile frameworks such as Scrum and Kanban, and the Advanced Work Packaging (AWP) methodology were analyzed to determine which approach offers greater flexibility and effectiveness in dynamic and complex project environments. This study explores how advancements in machine learning and data analysis can integrate with these frameworks to optimize schedule estimates. Some machine learning methods have shown potential across various disciplines due to their ability to learn from data and accurately anticipate outcomes. To address this issue, this study proposes the implementation of machine learning models as an auxiliary tool in the estimation process. The method employed includes the use of k-fold cross-validation to ensure the robustness and generalization of the model. Validation criteria based on metrics such as MSE (Mean Squared Error), MAE (Mean Absolute Error), and R^2 (Coefficient of Determination) were used to assess the accuracy and reliability of the predictions generated by the models.

KEY WORDS

Project Management; Machine Learning; Estimates; Project Schedule; Agile Frameworks; AWP.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ANN - Artificial Neural Networks (Redes Neurais Artificiais)

AWP - Advanced Work Packaging (Empacotamento de Trabalho Avançado)

CNNs - Convolutional Neural Networks (Redes Neurais Convolucionais)

DSR - Design Science Research (Metodologia Ciência do Design)

GANs - Generative Adversarial Networks (Redes Adversárias Generativas)

LSTM - Long Short-Term Memory (Memória de Longo e Curto Prazo)

MAE - Mean Absolute Error (Erro Absoluto Médio)

ML - Machine Learning (Aprendizado de Máquina)

MSE - Mean Squared Error (Erro Quadrático Médio)

PLN - Processamento de Linguagem Natural (Natural Language Processing)

PMBOK - Project Management Body of Knowledge (Corpo de Conhecimento em Gerenciamento de Projetos)

PMI - Project Management Institute (Instituto de Gerenciamento de Projetos)

RMSE - Root Mean Squared Error (Raiz do Erro Quadrático Médio)

RNNs - Recurrent Neural Networks (Redes Neurais Recorrentes)

R^2 - Coefficient of Determination (Coeficiente de Determinação)

SVM - Support Vector Machines (Máquinas de Vetores de Suporte)

1. INTRODUÇÃO

Nesta seção serão apresentados o contexto, a motivação, objetivos e resultados esperados desta pesquisa.

1.1. CONTEXTO

De acordo com o PMBOK 7ª Edição, projetos são considerados sistemas para entregar valor às organizações onde são executados. Este valor pode ser alcançado por meio da satisfação do cliente, criação de um novo produto ou serviço, melhoria de um processo ou implementação de uma nova mudança que possa aprimorar processos e operações. A gestão de valor é um aspecto fundamental, focando em motivar pessoas, desenvolver habilidades e fomentar sinergias e inovação com o objetivo final de otimizar o desempenho organizacional. Portanto, os projetos são vistos não apenas em termos de entregáveis, mas também pelo valor que trazem para a organização, estando alinhados com estratégias, missões e objetivos organizacionais e se caracterizam pela temporalidade, singularidade e progressividade e é por meio deles que é possível alcançar desafios que estão diretamente associados à implementação do planejamento estratégico.

1.2. MOTIVAÇÃO

De acordo com o Estudo de Benchmarking em Gerenciamento de Projetos 2010, o maior problema apontado pelos respondentes da pesquisa foi o não cumprimento de prazos, frequentemente baseado exclusivamente na intuição e experiência do gerente de projetos, o que aumenta o risco de decisões equivocadas. Para abordar esse problema, este estudo propõe a implementação de modelos de aprendizado de máquina como uma ferramenta auxiliar no processo de estimativa de prazos.

1.3. OBJETIVOS

A proposta desta dissertação é desenvolver modelos que possam ser implementados para portfólios de projetos de diferentes entidades, como governos, empresas, organizações não governamentais que contribuam para trazer maior previsibilidade nos projetos e, consequentemente, minimizar riscos.

Portanto, a principal pergunta a ser respondida com esta pesquisa é: como *machine learning* pode contribuir nas estimativas de projetos?

Para subsidiar e contribuir para a busca do questionamento supracitado, outras perguntas complementares farão parte desta dissertação:

1. Quais são os principais desafios enfrentados no processo de estimativa de prazos em projetos?
2. Como os modelos de *machine learning* podem ser aplicados para melhorar a estimativa de prazos em projetos de gerenciamento?
3. Como os modelos de *machine learning* podem lidar com a incerteza e a variabilidade nos projetos?
4. Quais são os principais desafios ou limitações na implementação de modelos de *machine learning* no contexto do gerenciamento de projetos?

1.4. RESULTADOS ESPERADOS E CONTRIBUIÇÃO

Este estudo tem como objetivo identificar padrões e desenvolver previsões mais precisas, oferecendo uma alternativa eficaz aos modelos de otimização tradicionais que dependem de equações matemáticas e exigem a predefinição de níveis de recursos por atividade. A adoção de modelos de *machine learning* introduz uma abordagem inovadora que pode contribuir significativamente para a redução das incertezas no processo de tomada de decisão. Espera-se que os resultados deste estudo incluam a criação de ferramentas preditivas mais eficazes para a gestão de prazos em projetos, melhorando potencialmente a eficiência e a eficácia da gestão de projetos em várias organizações. Ao final, a pesquisa pretende fornecer insights valiosos e práticos que possam ser aplicados para otimizar a gestão de projetos e reduzir os riscos associados à estimativa de prazos, beneficiando uma ampla gama de setores e indústrias.

2. REVISÃO DA LITERATURA

Este capítulo pretende apresentar a revisão da literatura utilizada na elaboração deste trabalho. Os principais conceitos a serem abordados são: gerenciamento de projetos, programas e portfólios, principais técnicas de estimativa, *machine learning* e utilização de *machine learning* em estimativas de projetos.

2.1. GERENCIAMENTO DE PROJETOS, PROGRAMAS E PORTFÓLIOS

Os indícios mais antigos da prática de gerenciar projetos são encontrados nas grandes construções da antiguidade, que necessitavam de planejamento, organização e coordenação eficaz. Entre os exemplos mais marcantes estão a construção das Pirâmides do Egito por volta de 2570 a.C., a edificação do Muro de Berlim aproximadamente em 500 a.C., diversos empreendimentos de engenharia da Roma Antiga entre 312 a.C. e o século IV d.C., e a elaboração da Grande Muralha da China, que se estendeu do século VII a.C. até o século XVI d.C. Estes projetos históricos ilustram como a humanidade sempre buscou alcançar objetivos específicos em um intervalo determinado de tempo, o que define um projeto de acordo com o PMI (Project Management Institute). Contudo, o gerenciamento de projetos, como uma disciplina formal, começou a se desenvolver no setor militar após a Segunda Guerra Mundial, conforme mencionado por Webster citado por Crawford em 2002. Desde meados da década de 1990, a adoção do gerenciamento de projetos tem se intensificado. Kerzner, em 2002, enfatizou que a implementação eficaz do gerenciamento de projetos é fundamental para a gestão avançada de projetos nas empresas. Para ser bem-sucedido, o processo de implantação deve ser alinhado com a cultura organizacional, incluir treinamentos abrangentes e contar com o apoio dos executivos, que devem valorizar o impacto positivo do gerenciamento de projetos na organização.

O PMBOK (Project Management Body of Knowledge) é o livro referência de gerenciamento de projetos, elaborado e continuamente atualizado pelo PMI. Em sua sétima edição, o PMBOK apresenta 7 principais áreas de desempenho, de acordo com a tabela abaixo.

Tabela 1 – Áreas de Desempenho do PMBOK 7ª edição

Áreas de Desempenho	Foco
Equipe	Refere-se ao gerenciamento e liderança de pessoas envolvidas no projeto, incluindo o desenvolvimento da equipe, a promoção de um ambiente colaborativo e a comunicação eficaz.
Stakeholders	Enfoca na identificação, engajamento e gestão das expectativas de todas as partes interessadas no projeto.

Ciclo de Vida do Projeto	Inclui o entendimento das fases pelas quais o projeto passa, desde a iniciação até a conclusão, e a aplicação de processos e práticas adequadas a cada fase.
Planejamento	Abrange o desenvolvimento de planos e estratégias para guiar a execução e controle do projeto.
Navegação pela Incerteza e Ambiguidade	Reconhece a natureza incerta dos projetos e enfatiza a necessidade de técnicas para lidar com incertezas e mudanças.
Entrega de Valor	Concentra-se na realização dos objetivos do projeto e na entrega de resultados que geram valor para os stakeholders.
Medição de Desempenho	Envolve o monitoramento, a medição e a análise do desempenho do projeto para garantir que ele esteja alinhado com os objetivos definidos.
Adaptação e Resiliência	Refere-se à capacidade de adaptar-se a mudanças e desafios durante o projeto, mantendo a resiliência e a capacidade de responder a imprevistos.

O PMBOK 7, apesar de ser focado principalmente em projetos, também aborda a relação e a distinção entre projetos, programas e portfólios. Esta relação é crucial para entender como diferentes iniciativas são gerenciadas dentro de uma organização:

- Projeto: É um esforço temporário realizado para criar um produto, serviço ou resultado único. Projetos são geralmente limitados por fatores como tempo, custo e recursos.
- Programa: Consiste em um grupo de projetos relacionados gerenciados de maneira coordenada para obter benefícios e controle que não estariam disponíveis se os projetos fossem gerenciados individualmente. Programas podem incluir elementos de trabalho relacionados fora do escopo dos projetos individuais.
- Portfólio: É uma coleção de projetos, programas, operações do dia-a-dia e outros trabalhos agrupados juntos para facilitar o gerenciamento efetivo desses empreendimentos para atingir objetivos estratégicos. Os portfólios não necessariamente envolvem projetos relacionados; eles são organizados com base em objetivos estratégicos específicos.

No âmbito da gestão organizacional, a integração entre projetos, programas e portfólios desempenha um papel crucial no alinhamento com os objetivos estratégicos. Esta integração pode ser compreendida como uma cadeia de conexões onde cada nível suporta e fortalece o outro, conduzindo a organização em direção às suas metas estratégicas.

- Projetos: Projetos são tipicamente táticos, focados em entregar resultados concretos ou inovações dentro de um escopo e prazo definidos. A seleção e priorização de projetos devem

refletir as necessidades estratégicas da organização, garantindo que cada projeto contribua para um aspecto maior dos objetivos de longo prazo.

- Programas: Agrupam vários projetos relacionados, gerenciando-os de forma coordenada para obter benefícios maiores do que seriam possíveis se os projetos fossem gerenciados isoladamente. Os programas são estratégicos por natureza, pois eles não apenas visam entregar seus resultados específicos, mas também garantir que os projetos sob sua gestão estejam alinhados e se complementam. Isso permite que a organização capitalize em sinergias, otimize recursos e gerencie riscos de maneira mais eficaz.

- Portfólios: Representam o nível mais alto de gestão estratégica. Um portfólio é uma coleção de projetos e programas que são selecionados e priorizados com base na estratégia global da organização. A gestão de portfólios envolve a avaliação contínua de projetos e programas para garantir que estejam alinhados com os objetivos estratégicos e que os recursos estejam sendo alocados de maneira eficiente. Esta abordagem holística assegura que a organização se mantenha focada em suas metas de longo prazo, adaptando-se às mudanças do ambiente e às oportunidades emergentes.

2.2. PRINCIPAIS TÉCNICAS DE ESTIMATIVAS DE PROJETO

O processo de estimar é um dos mais importantes de um projeto e abrange quase todas as áreas de desempenho. Seja ao estimar custos, esforço ou duração de atividades, a acuracidade das estimativas tende a influenciar diretamente nos resultados. Contudo, é importante ressaltar que a estimativa não se trata de uma ciência exata, mas algo que pode seguir uma série de ações determinísticas responsáveis por produzirem estimativas condizentes com a realidade, assim como afirma Maxim e Pressman (2016): As estimativas de custo e esforço nunca serão uma ciência exata. Muitas variáveis – fatores humanos, técnicos, ambientais e políticos – podem afetar o custo final do software e o esforço necessário para desenvolvê-lo. No entanto, seguir algumas etapas sistemáticas ou técnicas consolidadas ao longo dos anos tende a minimizar os riscos relacionados a este processo. Algumas das principais técnicas apresentadas no PMBOK 7ª edição estão listadas na tabela abaixo.

Tabela 2 – Técnicas de estimativa do PMBOK 7ª edição

Técnica	Resumo
Estimativa Analógica	Esta técnica utiliza o custo real de projetos anteriores, que são semelhantes ao projeto atual, como base para estimar o custo do projeto atual. É frequentemente usada nas fases iniciais de um projeto, quando informações detalhadas são limitadas.

Estimativa Paramétrica	Envolve o uso de relações estatísticas entre dados históricos e outras variáveis do projeto. Baseia-se nas taxas de custo unitário dos recursos e requer dados históricos de boa qualidade para precisão.
Estimativa Bottom-Up	Esta abordagem decompõe o trabalho do projeto em elementos menores e estima o custo de cada um. É uma técnica detalhada que pode resultar em estimativas mais precisas.
Estimativa de Três Pontos	Utiliza três valores diferentes (otimista, mais provável e pessimista) para estimar o custo. Esta técnica leva em conta a incerteza e a variabilidade nas estimativas de custos.
Análise de Reserva	Envolve a reserva de um contingente para cobrir custos imprevistos potenciais. É um componente crítico da gestão de custos, ajudando a garantir que o projeto possa se adaptar a mudanças e desafios inesperados.
Julgamento de Especialistas	Baseia-se na experiência e expertise de profissionais ou especialistas para estimar os custos das atividades do projeto.

2.3. PRINCIPAIS FRAMEWORKS DE PROJETO

Este capítulo tem como objetivo explorar diversos frameworks de gestão de projetos, destacando suas características, vantagens e aplicações práticas. Serão discutidos tanto frameworks tradicionais, como o PMI (PMBOK) e o Prince2, quanto metodologias ágeis, como o Scrum e o Kanban, que se destacam pela flexibilidade e capacidade de adaptação em ambientes de rápida mudança. Além disso, serão abordadas metodologias específicas como o Lean e o Six Sigma, que focam na eficiência operacional e na redução de defeitos, respectivamente, bem como o Advanced Work Packaging (AWP), uma abordagem especializada para a gestão de grandes projetos de construção.

Tabela 3 – Principais Frameworks de Projeto

Framework	Descrição	Principais características	Aplicações Comuns
Agile	Um conjunto de princípios para desenvolvimento de software e gestão de projetos que enfatiza flexibilidade e entrega rápida.	Iterações curtas (sprints), feedback contínuo, adaptação rápida às mudanças, colaboração com o cliente.	Desenvolvimento de software, projetos com alta incerteza.
Scrum	Uma metodologia ágil específica focada em entregas incrementais e em colaboração de equipe.	Sprints de 2-4 semanas, papéis específicos (Scrum Master, Product Owner, Equipe de Desenvolvimento), reuniões diárias (Daily Scrum), revisão e	Desenvolvimento de software, projetos ágeis em geral.

		retrospectiva ao final de cada sprint.	
Kanban	Um método ágil que utiliza um sistema visual de cartões para gerenciar o trabalho conforme ele se move através de um processo.	Visualização do fluxo de trabalho, limitações de trabalho em progresso (WIP), melhoria contínua, entrega contínua.	Manufatura, TI, desenvolvimento de software, serviços.
Lean	Uma abordagem que visa maximizar o valor para o cliente minimizando o desperdício.	Eliminação de desperdícios, melhoria contínua (Kaizen), entrega de valor, respeito às pessoas.	Manufatura, operações, desenvolvimento de software.
Prince2	Um método estruturado para gestão de projetos, amplamente utilizado no Reino Unido e internacionalmente.	Foco em justificativa de negócio, organização definida, abordagem baseada em processos, adaptação ao ambiente do projeto.	Projetos governamentais, grandes organizações.
Six Sigma	Uma metodologia de melhoria de processos focada na redução de defeitos e variabilidade.	DMAIC (Definir, Medir, Analisar, Melhorar, Controlar), forte foco em dados e estatísticas, certificação de níveis (Green Belt, Black Belt).	Manufatura, serviços, saúde, processos de negócios.
Advanced Work Packaging (AWP)	Uma metodologia de gestão de projetos para construção, desenvolvida pelo Construction Industry Institute (CII).	Planejamento antecipado, criação de pacotes de trabalho detalhados (CWPs e IWPs), colaboração entre todas as partes envolvidas, maior visibilidade e controle do projeto.	Projetos de construção, grandes empreendimentos industriais.
Waterfall	Um modelo de desenvolvimento sequencial, onde o progresso flui em uma direção através das fases do projeto.	Fases claramente definidas (Requisitos, Design, Implementação, Verificação, Manutenção), documentação extensa, mudanças difíceis de incorporar após o início.	Desenvolvimento de software tradicional, projetos com requisitos bem definidos.

Dadas as especificidades de cada framework ou metodologia, para este trabalho serão utilizados cronogramas de projeto em Advanced Work Packaging (AWP).

2.4. MACHINE LEARNING

De acordo com Tom M. Mitchell (2007), *Machine Learning*, é o campo da Inteligência Artificial que se concentra na questão de como programas de computador devem ser construídos para aprender a partir da experiência. Acredita-se que os primeiros estudos de *machine learning* tinham como foco uso de jogos como meio para simular comportamentos que, em humanos ou

animais, implicaria algum tipo de processo de aprendizagem e o precursor foi Arthur Samuel em 1959. De acordo com o estudo seria possível que um computador fosse programado para aprender a jogar damas e se saísse melhor no jogo do que as pessoas que escreveram o programa (Samuel, 1959). Essa ideia pioneira de Samuel em 1959 abriu caminho para o desenvolvimento do campo do aprendizado de máquina. Esta inovação de Samuel marcou um avanço significativo no campo do aprendizado de máquina, que é um subsetor da inteligência artificial. Este campo é caracterizado pela habilidade de máquinas em replicar comportamentos humanos inteligentes, permitindo que sistemas de IA executem tarefas complexas de maneira semelhante à resolução de problemas por humanos, como explicado por Sara Brown em 2021. Outro modo de entender o aprendizado de máquina é considerá-lo como um processo onde a performance de um agente em uma tarefa específica melhora com o tempo, baseado na análise de informações disponíveis. Por exemplo, ao analisar pares de valores (x, y) - onde x é a entrada de uma função desconhecida f, resultando em y = f(x) - um algoritmo que consegue prever y com precisão ao receber novos valores de x demonstra estar aprendendo a imitar o comportamento dessa função, como descrito por Russell e Norvig em 2013. Atualmente, o uso do *Machine Learning* se estendeu para áreas além da ciência da computação, incluindo medicina e biologia, conforme destacado por Ching e outros em 2018.

Mitchell (1997) descreve o aprendizado de máquina como o processo em que um programa de computador (modelo de aprendizado) aprende uma tarefa T a partir de exemplos ou experiências E. Conforme a performance P do programa melhora com a adição de mais exemplos E, pode-se dizer que o programa está efetivamente aprendendo a executar a tarefa T.

Existem diversas maneiras de aplicar *Machine Learning* (ML) em sistemas, variando de acordo com as necessidades específicas, o problema em questão, a quantidade de dados e a precisão exigida para o modelo. As quatro principais abordagens de ML, conforme Hurwitz e Kirsch (2018), são:

Tabela 4 – Abordagens de *Machine Learning*

Abordagem	Resumo
Aprendizado Supervisionado	Necessita de dados e classificações pré-definidos para aprender iterativamente e identificar padrões úteis em análises. Exemplo: classificar animais usando imagens e descrições.
Aprendizado Não Supervisionado	Opera sem supervisão humana em grandes conjuntos de dados não rotulados, classificando-os com base em padrões ou grupos identificados. Exemplo: detecção de spam em e-mails e sistemas de recomendação.

Aprendizado por Reforço	O sistema aprende por tentativa e erro, sem treinamento inicial, melhorando sua análise a cada nova tentativa. Exemplo: máquinas que jogam xadrez.
Aprendizagem Profunda (Deep Learning)	Utiliza redes neurais multi-camadas para processar dados não estruturados, simulando o funcionamento do cérebro humano em problemas complexos. Exemplo: aplicativos de reconhecimento de imagens.

Para esta investigação, as análises seguirão a Abordagem de Aprendizado Supervisionado.

2.3.1 Aprendizado Supervisionado

O aprendizado supervisionado é uma categoria de *machine learning* que utiliza conjuntos de dados rotulados por especialistas em uma determinada área (HAYKIN, 2001). Esse tipo de aprendizado infere uma função ou regra a partir dos dados de treinamento previamente classificados, permitindo a atribuição de classes a objetos nunca antes vistos (ALPAYDIN, 2004; MITCHELL, 1997). Em outras palavras, o objetivo do aprendizado supervisionado é prever uma saída para cada dado de entrada, baseando-se nos padrões aprendidos durante o treinamento com dados rotulados (MITCHELL, 1997). Um exemplo prático de algoritmos de aprendizado supervisionado é a classificação de e-mails como SPAM ou não-SPAM.

2.3.2 Aprendizado Não Supervisionado

O aprendizado não supervisionado trabalha com conjuntos de dados que não possuem rótulos. Dessa forma, os algoritmos buscam identificar como os dados estão organizados (MITCHELL, 1997). O objetivo do aprendizado não supervisionado é descobrir padrões desconhecidos no conjunto de dados, permitindo o reconhecimento de agrupamentos (ALPAYDIN, 2004; MITCHELL, 1997). Um exemplo didático de aplicação de algoritmos de aprendizado não supervisionado é o agrupamento de balões de diferentes cores com base em suas semelhanças de cor.

2.3.2 Aprendizado por Reforço

O aprendizado por reforço baseia-se na ideia de que, se uma ação é seguida por estados satisfatórios ou por uma melhoria no estado, a tendência para produzir essa ação aumenta, ou seja, é reforçada. Estendendo essa ideia, ações podem ser selecionadas com base na informação sobre os estados que elas podem produzir, o que introduz aspectos de controle com realimentação.Problemas em Aprendizado por Reforço (AR) são caracterizados por um agente que deve aprender comportamentos por meio de interações de tentativa e erro em um ambiente dinâmico (Kaelbling e Littman, 1996). O AR não é definido como um conjunto específico de algoritmos de aprendizado, mas

sim como uma classe de problemas de aprendizado. Qualquer algoritmo que resolva bem esses problemas será considerado um algoritmo de aprendizado por reforço (Sutton e Barto, 1998).

2.3.4 Deep Learning (Aprendizagem Profundo)

Deep Learning é uma sub-área da Inteligência Artificial, especificamente do *Machine Learning*, focada no desenvolvimento de técnicas para resolver problemas complexos onde não é possível estabelecer regras determinísticas. Para isso, a máquina deve aprender uma hierarquia de conceitos a partir de versões mais simples (DENG; YU et al., 2014). Além disso, não é necessário que um agente humano defina formal e extensivamente as regras, pois o sistema deve ser capaz de inferir a partir da experiência.

2.3.5 Abordagens, Categorias e Algorítmos

Para cada abordagem existem algoritmos específicos, que foram desenvolvidos a partir dos princípios de várias técnicas de aprendizado de máquina, esses algoritmos podem ser aplicados em diferentes contextos para fins variados. Alguns dos algoritmos disponíveis no mercado e suas bases de raciocínio constam na tabela a seguir:

Tabela 5 – Abordagens, categorias e Algorítmos de *Machine Learning*

Abordagem	Categoria	Algorítimos/Técnicas
Aprendizado Supervisionado	Regressão	Regressão Linear - Regressão Logística - PCA (Análise de Componentes Principais) - HCA (Análise Hierárquica de Agrupamentos) - PLS (Regressão por Mínimos Quadrados Parciais) - iPLS (Regressão Parcial por Mínimos Quadrados por Intervalos) - siPLS (Regressão Parcial por Mínimos Quadrados por Intervalos de Sinergia) - Elastic Net - Ridge Regression - Lasso Regression
	Classificação	Redes Neurais Artificiais Perceptron de Múltiplas Camadas (ANN - MLP) - Máquinas de Vetores de Suporte (SVM) - C4.5 (Árvore de Decisão) - Florestas Aleatórias (Random Forest) - IBK (Instance-Based K) - AdaBoost (Adaptive Boosting) - Bagging - LogitBoost - XGBoost (Extreme Gradient Boosting) - Gradient Boosting - k-Nearest Neighbors (k-NN) - Naive Bayes - LightGBM - CatBoost

Aprendizado Não Supervisionado	Agrupamento	K-means - DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - Mean Shift - Gaussian Mixture Models (GMM) - Clustering (Agrupamento)
	Problema do Coquetel	Cocktail Party Problem (Separação de fontes sonoras)
	Dimensionalidade	t-SNE (t-Distributed Stochastic Neighbor Embedding) - UMAP (Uniform Manifold Approximation and Projection)
Aprendizado por Reforço	Decisão Markoviana	Processos de Decisão de Markov (MDP)
	Algoritmos	Q-learning - Sarsa - Q(λ) - Dyna - Deep Q-Network (DQN) - Proximal Policy Optimization (PPO) - Trust Region Policy Optimization (TRPO) - Asynchronous Advantage Actor-Critic (A3C)
	Decisão Markoviana Parcial	Processos de Decisão de Markov Parcialmente Observáveis (POMDP)
	Construção de Mapas	Map Building (Construção de Mapas)
Deep Learning (Aprendizado Profundo)	Redes Neurais	Redes Neurais Artificiais (ANN) - Redes Neurais Convolucionais (CNNs) - Redes Neurais Recorrentes (RNNs) - Long Short-Term Memory (LSTM) - Redes Generativas Adversariais (GANs) - Autoencoders

Neste trabalho, com o objetivo de realizar uma análise da aplicação de *machine learning* no processo de estimativa de duração de atividades de um projeto, neste trabalho serão utilizados algoritmos C4.5 (Árvore de Decisão), Florestas Aleatórias, Regressão Linear e Gradient Boosting. Para um melhor entendimento das aplicações, seguem as definições e abrangências:

Tabela 6 – Algoritmos utilizados

Algorítmo	**Definição e Abrangência**
C4.5 (Árvore de Decisão)	O algoritmo C4.5 é uma técnica de aprendizado supervisionado usada para classificação e regressão. Ele gera uma árvore de decisão com base em conjuntos de dados rotulados, permitindo tanto a classificação de novos dados quanto a previsão de valores contínuos com base nas regras derivadas

	das árvores criadas.
Random Forest (Floresta Aleatória)	As Florestas Aleatórias são um conjunto de árvores de decisão usadas tanto para classificação quanto para regressão. Elas funcionam construindo múltiplas árvores de decisão durante o treinamento e emitindo a média ou modo das previsões individuais das árvores para obter a previsão final.
Regressão Linear	A Regressão Linear é uma técnica de aprendizado supervisionado que modela a relação entre uma variável dependente e uma ou mais variáveis independentes através de uma equação linear. É amplamente usada para prever valores numéricos contínuos com base em dados históricos.
Gradient Boosting	O Gradient Boosting é um método de aprendizado de máquina que cria um modelo preditivo forte a partir da combinação de vários modelos fracos, geralmente árvores de decisão. Ele trabalha iterativamente, ajustando cada árvore para corrigir os erros das árvores anteriores, melhorando assim a precisão das previsões.

○

2.5. *MACHINE LEARNING* EM ESTIMATIVAS DE PROJETOS

As dificuldades com estimativas em projetos são objeto de estudo e um problema amplamente explorado na literatura e, ao longo dos estudos que envolvem estimativas para projetos de desenvolvimento de software remetem à década de 90, quando diversas técnicas foram propostas, adaptadas ou criadas com o objetivo de auxiliar a estimativa de esforço (WEN et al., 2012). Um estudo realizado por Idri, Hosni e Abran (2016) conclui que apesar do grande número de estudos de estimativa de esforço, não há consenso que determine o melhor método único. No entanto, Shepperd & Kadoda (2001) sugerem que é mais frutífero determinar o melhor modelo num contexto particular do que determinar o melhor modelo único, uma vez que os modelos de estimativa se comportam de forma diferente de um conjunto de dados para outro, o que os torna instáveis. Além disso, Nayebi et al. (2015) propõem uma abordagem baseada em dois tipos de medidas de desempenho para selecionar o modelo que produz estimativas mais precisas para um conjunto de dados específico. Ainda de acordo com Idri, Hosni e Abran (2016), as técnicas que envolvem *machine learning* têm obtido maior êxito na tarefa de estimativa de esforço em relação às estimativas por especialistas e métodos algorítmicos.

3. METODOLOGIA

Para fornecer uma base teórica e prática robusta a este estudo, será adotada a metodologia de Pesquisa em Design Science Research (DSR). Esta técnica foca no desenvolvimento de um artefato visando agregar valor à otimização de um sistema e, por sua vez, é um método que fundamenta e operacionaliza a pesquisa, que tem como objetivo um artefato ou uma prescrição. A Design Science Research representa o fundamento epistemológico no estudo do artificial.

Design Science Research (DSR) é uma abordagem de pesquisa utilizada em diversas disciplinas, como sistemas de informação, engenharia de software, e gestão de tecnologia da informação. Não é atribuída a uma única pessoa a criação do Design Science Research; em vez disso, ela evoluiu ao longo do tempo através das contribuições de vários acadêmicos, no entanto, Herbert A. Simon foi uma figura influente no estabelecimento das bases para o Design Science com sua obra "The Sciences of the Artificial", publicada pela primeira vez em 1969, onde discutiu a natureza do design e sua importância como uma forma de ciência. A distinção entre os ambientes natural e artificial foi originalmente feita por Herbert Simon (1969, 1996). De acordo com Simon (1996), a ciência natural diz respeito a um conjunto de conhecimentos sobre uma classe de objetos e fenômenos do mundo, explorando suas características, comportamentos e interações. Assim, as disciplinas científicas naturais têm a tarefa de investigar e ensinar como as coisas são e funcionam. Este raciocínio aplica-se tanto aos fenômenos naturais, como biologia, química e física, quanto aos fenômenos sociais, como economia e sociologia.Além disso, iintroduz a ideia de estudar o universo "artificial", definindo as "ciências do artificial" como disciplinas que se ocupam da "concepção de artefatos que realizem objetivos" (SIMON, 1996, p. 198). Em outras palavras, as ciências do artificial focam em como as coisas devem ser para funcionar de acordo com objetivos específicos. Este campo é particularmente relevante para áreas como a engenharia, que ensina a criar e projetar artefatos com propriedades desejadas e objetivos definidos (SIMON, 1996).

No campo dos sistemas de informação, um marco importante foi o trabalho de Hevner, March, Park, e Ram, que, em 2004, publicaram "Design Science in Information Systems Research" no MIS Quarterly. Este trabalho ajudou a consolidar os princípios e as diretrizes para realizar pesquisa em Design Science na área de sistemas de informação, delineando como os artefatos de DSR deveriam ser criados e avaliados. Simon (1996) defende a criação de uma ciência dedicada a propor como construir artefatos com determinadas propriedades desejadas, ou seja, como projetá-los. Ele chama essa disciplina de "Ciência do Projeto" ou Design Science. "Ao projeto interessa o quê e como as coisas devem ser, a concepção de artefatos que realizem objetivos." (SIMON, 1996, p. 198). A

principal missão da Design Science é desenvolver conhecimento para a concepção e desenvolvimento de artefatos (VAN AKEN, 2004).

A condução da Design Science Research (DSR) conforme é concebida nos dias atuais possui fatores fundamentais para o êxito da pesquisa: o rigor e a relevância. A relevância trata principalmente de qual problema busca ser resolvido e sua importância para as organizações e/ou sociedade, afinal, serão as pessoas que utilizarão o resultado da pesquisa e do conhecimento para solucionar problemas. O rigor se refere ao quanto a pesquisa pode ser considerada válida, confiável e contribuir para a base de conhecimento existente em determinada área. A imagem a seguir representa as principais etapas da Design Science Research.

Figura 1 - Rigor e Relevância em DSR (Dresch, 2015)

O ambiente diz respeito a onde o problema é observado e é neste contexto em que o artefato irá operar. A composição do ambiente é de pessoas, organizações e tecnologia. Contudo, antes da idealização de Design Science Research como é conhecida atualmente, alguns outros métodos com propostas semelhantes, quase sempre proveniente de sistemas de informação, haviam sido mapeados. Mario Bunge (1980) foi um dos autores que estabeleceu alguns passos para a condução da pesquisa tecnológica, conforme imagem a seguir.

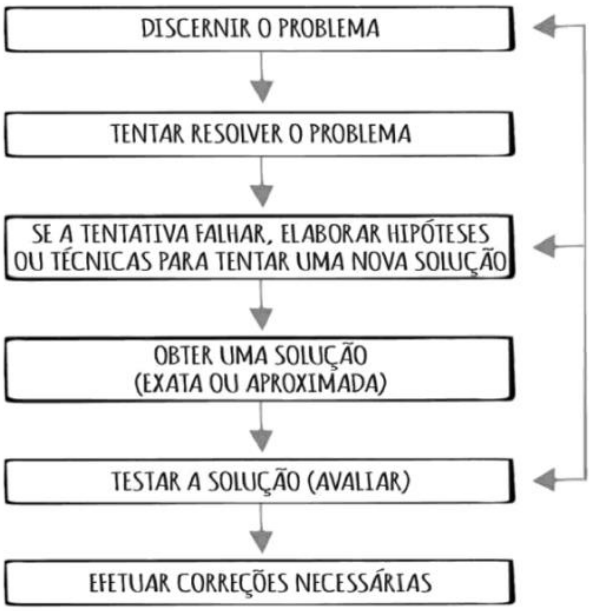

Figura 2 - Passos para condução de Pesquisa Tecnológica (Bunge, 1980)

Para Bunge (1980), mais do que a identificação, é fundamental, discernir o problema, ou seja, a colocação precisa do problema a ser estudado ou da tecnologia a ser desenvolvida. Desta forma, uma vez compreendido o problema, o pesquisador poderá passar à etapa seguinte, que tem como objetivo resolvê-lo com o conhecimento existente, tanto teórico como empírico. Bunge argumenta a favor de uma abordagem sistemática para a pesquisa científica, enfatizando a importância da formulação clara de problemas, da construção de modelos teóricos e da aplicação rigorosa de métodos empíricos para testar hipóteses. Ele defende uma visão realista da ciência, na qual a ciência procura descrever e entender a realidade objetiva, e critica abordagens relativistas ou puramente subjetivas do conhecimento científico. No contexto do Design Science Research (DSR), as ideias de Bunge sobre a pesquisa sistemática, a importância dos modelos teóricos e a validação empírica dos resultados são particularmente relevantes. Embora Bunge não tenha se dedicado especificamente ao DSR em seus trabalhos, seus princípios metodológicos influenciam a abordagem do DSR de várias maneiras: no enfoque sistemático, na modelagem teórica e na validação empírica.

4. ESTUDO EMPÍRICO

Esta seção será o núcleo as técnicas de aprendizado de máquina são aplicadas a dados reais de projetos para explorar sua capacidade de estimar prazos de forma eficaz. Esta seção detalha o processo desde a concepção do modelo, passando pela seleção e tratamento dos dados, até a implementação prática do algoritmo e a avaliação rigorosa de seu desempenho. A finalidade é não apenas mostrar a viabilidade das técnicas de aprendizado de máquina para a estimativa de prazos, mas também delinear as melhores práticas, desafios enfrentados e soluções encontradas ao longo do caminho.

Inicialmente, serão apresentadas as pressupostos e práticas que fundamentam o desenvolvimento do modelo. Isso inclui uma discussão sobre a escolha das variáveis, a relevância dos dados históricos de projetos e a justificativa para a seleção de uma determinada técnica de aprendizado de máquina. Segue-se a definição do modelo, onde a construção do algoritmo é explicada em detalhes. Este segmento aborda desde a preparação e codificação dos dados até a escolha específica dos parâmetros do modelo e a lógica por trás das decisões de modelagem. A implementação prática do modelo é o próximo foco, detalhando o processo de treinamento do algoritmo, a seleção de características, e como os dados são utilizados para ajustar o modelo à realidade dos projetos. Este processo é essencial para transformar a teoria em prática, permitindo que o modelo aprenda com os dados disponíveis e faça previsões precisas sobre a duração das atividades do projeto.

4.1. PRESSUPOSTOS

A maioria dos frameworks de gestão de projetos não estabelece como premissa a codificação das atividades, uma prática que pode facilitar significativamente a análise e estimativa de prazos em projetos. Uma notável exceção é o Advanced Work Packaging (AWP), amplamente adotado em projetos de capital, que implementa a codificação das atividades como um componente integral de sua metodologia. Essa abordagem permite uma organização e planejamento mais detalhados do trabalho, contribuindo para uma execução de projeto mais eficiente e previsível. Neste contexto, o presente trabalho visa explorar a aplicação de técnicas de aprendizado de máquina para a estimativa de prazos em projetos, focalizando em cronogramas onde a codificação das atividades é realizada de maneira manual, conforme prática no AWP, e em situações onde não há codificação prévia, aplicando processamento de linguagem natural (PLN) para estruturar e analisar os dados. Esta abordagem dual permite uma avaliação comparativa da eficácia das técnicas de aprendizado de máquina em contextos distintos de organização de dados de projeto, destacando a importância da estruturação de dados no processo de estimativa de prazos.

4.1.1 Codificação das atividades

A codificação das atividades, no contexto de aprendizado de máquina, refere-se à transformação de dados textuais ou categóricos (como descrições de atividades) em um formato numérico ou em categorias padronizadas que podem ser processadas por algoritmos de aprendizado de máquina. Essa transformação é crucial, pois os algoritmos de aprendizado de máquina requerem dados em um formato que possam interpretar e processar eficientemente, seja através de codificação manual, como no caso do AWP.

Natureza da Codificação: A necessidade de codificar dados textuais ou categóricos em formatos numéricos ou categorias padronizadas surge da exigência técnica dos algoritmos de aprendizado de máquina, que operam mais eficientemente com dados estruturados.

Métodos de Codificação: Este trabalho abordará tanto a codificação manual, específica do framework AWP para avaliar a viabilidade e eficácia na estimativa de prazos de projeto.

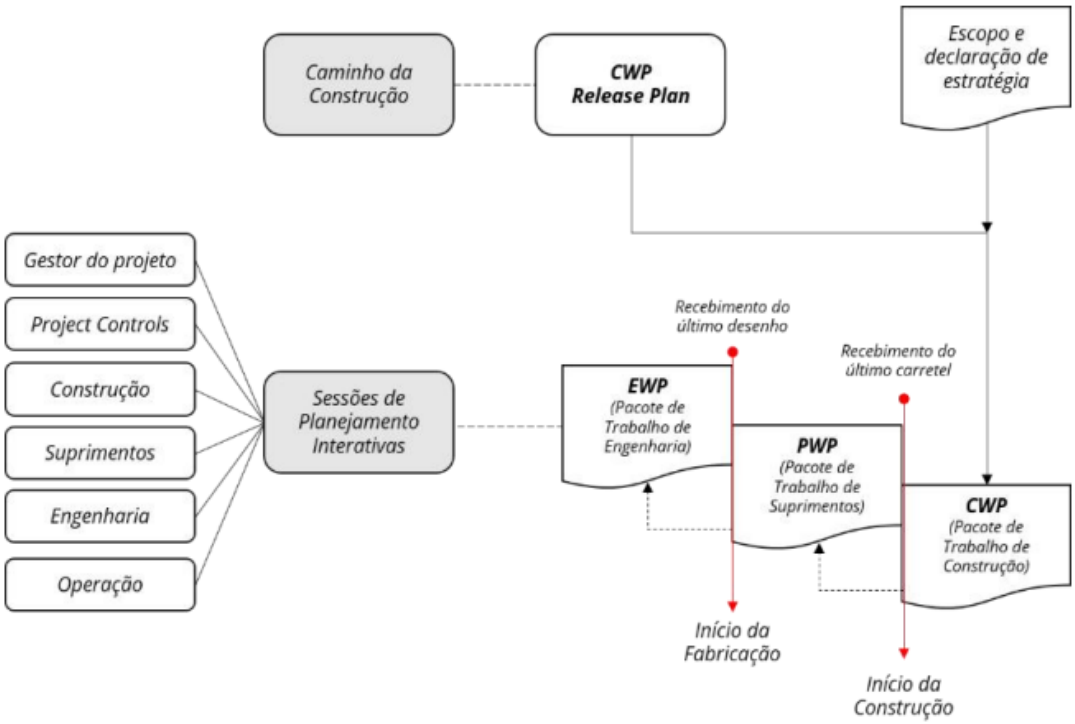

Figura 3 - Codificação AWP

Fonte: Adaptado de Ryan (2017, p. 40)

Os IWPs (Pacotes de Trabalho de Instalação) são segmentos de trabalho que compreendem cerca de 500 horas e uma única disciplina. Eles são elaborados a partir da divisão dos CWPs (Pacotes de Trabalho de Construção) após a remoção de restrições primárias. Segundo Ryan (2017), "em termos gerais, pacotes menores são mais vantajosos para os encarregados, pois são mais fáceis de rastrear,

fechar e orientar sobre como o trabalho deve ser dividido e sequenciado". Esses pacotes são desenvolvidos 90 dias antes da execução do CWP correspondente.

Os EWPs (Pacotes de Trabalho de Engenharia) são documentos e desenhos que detalham as atividades de engenharia necessárias para a conclusão de um projeto. Eles servem como uma base para os CWPs (Pacotes de Trabalho de Construção) e são essenciais para garantir que todas as especificações técnicas e requisitos sejam atendidos. Os EWPs são desenvolvidos antes do início das atividades de construção e fornecem uma visão clara e detalhada do trabalho de engenharia que precisa ser realizado. Os PWPs (Pacotes de Trabalho de Planejamento) são conjuntos de atividades planejadas que detalham a sequência e a metodologia das tarefas a serem executadas em um projeto. Eles incluem cronogramas, recursos necessários e planos de execução detalhados, assegurando que o trabalho seja realizado de maneira eficiente e dentro do prazo. Os PWPs são criados com base nas informações dos EWPs e dos CWPs, fornecendo uma abordagem estruturada para a gestão e execução das atividades do projeto.

4.2. DEFINIÇÃO DO MODELO

Os modelos utilizados para este estudo foram os algoritmos de Regressão Linear Lasso, Random Forest, Decision Tree e Gradient Boosting. A Regressão Lasso é uma variante da regressão linear que aplica penalização L1 aos coeficientes de regressão, útil para seleção automática de variáveis e redução da complexidade do modelo. O Random Forest e o Decision Tree são métodos baseados em árvores de decisão, com o Random Forest utilizando múltiplas árvores para melhorar a precisão e controlar o overfitting. O Gradient Boosting constrói modelos sequenciais para corrigir erros dos modelos anteriores, melhorando a precisão preditiva. Estes métodos foram aplicados para garantir um modelo preditivo robusto e preciso.

4.2.1 Atributos do Modelo

As variáveis utilizadas no modelo foram selecionadas a partir do conjunto de dados do cronograma.. As variáveis preditoras foram categorizadas em colunas numéricas e categóricas. Antes de serem utilizadas no modelo, as variáveis numéricas foram padronizadas para garantir que todas estivessem na mesma escala. A variável categórica 'Peso da Atividade' foi especialmente importante e utilizada como peso do pacote no treinamento do modelo. Algumas das variáveis categóricas foram codificadas usando a técnica de One Hot Encoding, que transforma cada categoria em uma coluna binária, facilitando o processamento por algoritmos de aprendizado de máquina. Por exemplo, uma coluna "Cor" com categorias "Vermelho", "Verde" e "Azul" seria transformada em três colunas binárias: "Cor_Vermelho", "Cor_Verde" e "Cor_Azul". Esta técnica é crucial para a correta

interpretação das variáveis categóricas por parte dos algoritmos de aprendizado de máquina, conforme descrito por Hastie, Tibshirani e Friedman (2009).
Essa referência é amplamente utilizada no campo de aprendizado de máquina e oferece uma base teórica sólida sobre diversas técnicas de pré-processamento de dados, incluindo o One Hot Encoding.
É fundamental compreender e categorizar corretamente os diversos tipos de atributos presentes no conjunto de dados. Esta classificação permite não apenas a otimização do modelo preditivo, mas também a eficiência no processo de análise. Os atributos do conjunto de dados podem ser divididos nas seguintes categorias: alvo, preditivos, ignorados e meta. Kuhn e Johnson (2013) afirmam que "a importância de selecionar as variáveis corretas é fundamental para construir modelos eficazes". Segundo Witten, Frank, Hall, e Pal (2016), os atributos podem ser classificados em quatro principais categorias: alvo, preditivos, ignorados e meta.

Tabela 7 – Tipos de Atributos, Definições e Exemplos

Tipo de Atributo	**Definição**	**Exemplo**
Atributo Alvo (Target Variable)	O atributo alvo é aquele que se deseja prever ou explicar. É o principal foco de interesse em um problema de predição	Em um problema de previsão de duração de atividades, a variável Duração é o alvo.
Atributos Preditivos (Predictor Variables)	São os atributos utilizados para prever a variável alvo. Estes atributos fornecem a base de dados sobre a qual os modelos de aprendizado de máquina fazem suas previsões.	Nível da estrutura de tópicos, Percentual de conclusão, Peso da Atividade, entre outros, podem ser usados para prever a duração.
Atributos Ignorados (Ignored Variables)	Atributos que não têm utilidade no processo de análise ou modelagem preditiva. São excluídos para simplificar o modelo e evitar sobrecarga de dados.	Id Exclusiva e Id, que são identificadores únicos e não contribuem para a predição.
Atributos Meta (Meta Variables)	Atributos que não são usados diretamente para predição, mas podem ser úteis para descrever, segmentar ou documentar os dados. Eles podem fornecer contexto adicional e ajudar na interpretação dos resultados.	Nome e Responsável são exemplos de atributos meta

A tabela a seguir apresenta uma visão detalhada dos atributos utilizados neste estudo, incluindo o tipo de dado, uma breve descrição e a função de cada atributo (se preditiva, meta, alvo ou ignorada). Esta classificação é essencial para entender a estrutura do conjunto de dados e a justificativa para a seleção e exclusão de variáveis durante o processo de modelagem preditiva.

Tabela 8 – Atributos do Modelo

Atributo	Tipo de Dado	Descrição	Função
Id Exclusiva	Numérico	Identificador único para cada entrada no cronograma.	Meta
Id	Numérico	Linha do cronograma	Meta
Nível da estrutura de tópicos	Numérico	Nível hierárquico da tarefa no cronograma	Preditiva
Resumo	Categórico	Indica se é resumo das demais atividades	Ignorada
Nome	Texto	Descritivo da atividade	Meta
Duração	Numérico	Data prevista de duração em dias	Alvo
Início	Data	Data de início da atividade	Temporal
Término	Data	Data de término da atividade	Temporal
% concluída	Numérico	Percentual já concluído	Preditiva
Peso da Atividade	Numérico	Peso atribuído à atividade	Preditiva
Criticidade	Categórico	Indicação se é crítica ou não ao projeto	Meta
Tipo de tarefa	Categórico	Tipo de Tarefa (planejamento, execução, etc.)	Preditiva
Responsável	Texto	Nome do responsável	Ignorada
Fase	Categórico	Fase à qual a atividade está atrelada	Ignorada

4.2.2 Estrutura do Modelo

O modelo foi estruturado utilizando um *pipeline* do scikit-learn, que inclui as etapas de padronização dos dados e ajuste dos modelos de Regressão Linear Lasso, Random Forest, Decision Tree e Gradient Boosting. O *pipeline* facilita a aplicação consistente das transformações necessárias e o treinamento dos modelos em um único fluxo de trabalho. O uso de *pipelines* garante que as mesmas transformações aplicadas aos dados de treinamento sejam aplicadas aos dados de teste e novos dados, evitando problemas de fuga de dados e garantindo a reprodutibilidade. Esta abordagem permite a comparação eficiente entre diferentes algoritmos e a seleção do melhor modelo para a tarefa de predição.

4.3. IMPLEMENTAÇÃO

4.3.1 Preparação dos Dados

A preparação dos dados envolveu várias etapas importantes:

1. Leitura dos Dados: Os dados foram carregados a partir de arquivos Excel para os cronogramas 113 e 114.

2. Conversão da Duração: A coluna de duração, originalmente em formato textual (por exemplo, "5 dias"), foi convertida para um formato numérico para facilitar o processamento.

3. Seleção de Variáveis: Foram selecionadas as variáveis preditoras, excluindo os identificadores e a variável alvo.

4. Tratamento de Dados Faltantes: Valores ausentes nas colunas foram imputados usando estratégias diferentes para variáveis numéricas (média) e categóricas (moda).

5. Divisão dos Dados: Os dados do cronograma 113 foram divididos em conjuntos de treinamento e teste para permitir a avaliação do modelo. A divisão da amostra em dados de treinamento e de teste é fundamental para avaliar a capacidade de um modelo em generalizar novas observações, garantindo que ele não esteja apenas ajustado aos dados nos quais foi treinado. Essa prática permite verificar se o modelo apresenta boa performance não apenas nos dados utilizados para seu ajuste (treinamento), mas também na previsão de novas atividades (teste). As divisões mais comuns são 60:40, 70:30 ou 80:20, e a escolha da proporção depende do tamanho do conjunto de dados disponível. Em geral, quanto maior o número de observações, maior será a proporção do conjunto inicial utilizado para o treinamento, o que permite ao modelo aprender melhor os padrões dos dados. No contexto deste projeto, que visa predizer a duração das atividades de um cronograma baseado no peso delas, utilizamos uma amostra composta por 1509 exemplos. Optou-se pela divisão 70:30, utilizando 70% dos dados (n = 1056) para o treinamento dos algoritmos. Esses dados de treinamento são usados para ajustar o modelo, permitindo que ele aprenda os padrões e relações entre o peso das atividades e sua duração. Os 30% restantes dos dados (n = 453) foram reservados para testar a performance preditiva dos modelos ajustados. Esses dados de teste são fundamentais para avaliar a capacidade do modelo em generalizar suas previsões para novas atividades que não foram vistas durante o treinamento. Essa divisão

ajuda a garantir que o modelo não esteja apenas memorizando os dados de treinamento, mas sim aprendendo de forma a ser útil em situações reais futuras. Apesar da divisão em conjuntos de treinamento e teste, notou-se que o melhor resultado foi obtido por meio de validação cruzada, utilizando 10 folds. A validação cruzada com 10 folds permite uma avaliação mais robusta e confiável da performance do modelo, garantindo que todas as observações sejam usadas tanto para treinamento quanto para teste em diferentes iterações. Isso ajuda a reduzir a variabilidade nos resultados e proporciona uma estimativa mais precisa da capacidade de generalização do modelo.

4.3.2 Detalhes da Implementação

A implementação do modelo foi realizada usando o scikit-learn, uma biblioteca robusta para aprendizado de máquina em Python. O *pipeline* criado incluiu as seguintes etapas:

1. Imputação e Escalonamento: Um Column Transformer foi utilizado para aplicar a imputação de valores faltantes e o escalonamento das variáveis numéricas. As variáveis categóricas foram codificadas usando OneHotEncoding;
2. Treinamento do Modelo: O modelo foi treinado usando os dados de treinamento.
3. O desempenho do modelo foi avaliado no conjunto de testes utilizando as métricas de Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE), Erro Absoluto Médio (MAE) e o coeficiente de determinação (R^2).

4.3.3 Desenvolvimento do Modelo

O desenvolvimento do modelo envolveu a busca pelo melhor parâmetro de regularização por meio de técnicas de validação. Esse processo permitiu explorar diferentes valores de alpha e identificar aquele que minimiza o erro quadrático médio (MSE) no conjunto de validação. Após determinar o valor ótimo de alpha, o modelo foi treinado com este parâmetro utilizando todo o conjunto de treinamento. Em seguida, o modelo foi avaliado no conjunto de teste para verificar sua performance preditiva. Esta abordagem assegura que o modelo está bem ajustado e tem uma capacidade robusta de generalização.

4.4. VALIDAÇÃO DO MODELO

A validação do modelo foi realizada utilizando o conjunto de testes separado anteriormente. As métricas escolhidas para avaliação do desempenho incluíram o Erro Quadrático Médio (MSE), que mede a média dos quadrados dos erros ou desvios, isto é, a diferença entre os

valores previstos pelo modelo e os valores reais; a Raiz do Erro Quadrático Médio (RMSE), que é a raiz quadrada do MSE e proporciona uma medida de erro na mesma unidade dos dados; o Erro Absoluto Médio (MAE), que calcula a média dos valores absolutos dos erros, fornecendo uma visão da magnitude dos erros independentemente de sua direção; e o coeficiente de determinação (R^2), que quantifica quanto da variação nos dados é explicada pelo modelo. Estas métricas são amplamente utilizadas em problemas de regressão devido à sua capacidade de fornecer uma avaliação abrangente do desempenho do modelo, cada uma destacando aspectos distintos dos erros de previsão.

4.4.1 Métricas e Metodologia de Validação

A validação do modelo foi realizada utilizando o conjunto de testes separado anteriormente. As métricas escolhidas para avaliação do desempenho incluíram o Erro Quadrático Médio (MSE), que mede a média dos quadrados dos erros ou desvios, isto é, a diferença entre os valores previstos pelo modelo e os valores reais. Além do MSE, foram utilizadas outras métricas como o Erro Quadrático Médio da Raiz (RMSE), que é a raiz quadrada do MSE, fornecendo uma medida da magnitude dos erros em unidades originais; o Erro Absoluto Médio (MAE), que mede a média das magnitudes dos erros em previsões, sem considerar sua direção; e o coeficiente de determinação R^2, que fornece uma indicação de quão bem os valores observados são replicados pelo modelo, variando entre 0 e 1.

4.4.1 Erro Quadrático Médio (MSE)

O MSE calcula a média dos quadrados dos erros entre as previsões do modelo e os valores reais. Ele eleva ao quadrado as diferenças, o que tem o efeito de ampliar e, portanto, dar maior peso aos erros maiores.O MSE é útil especialmente quando grandes erros são particularmente indesejáveis. Ao quadrar os erros, o MSE penaliza desvios maiores de forma mais severa que os menores, o que pode ser crucial dependendo do contexto de aplicação.

4.4.2 Raiz do Erro Quadrático Médio (RMSE)

O RMSE é a raiz quadrada do MSE. Isso converte o erro de volta para as unidades originais dos dados, tornando-o mais interpretável.Por estar na mesma escala que a variável dependente, o RMSE permite uma comparação mais direta entre a magnitude do erro e os valores reais, facilitando a interpretação dos resultados.

4.4.3 Erro Absoluto Médio (MAE)

O MAE mede a média dos valores absolutos das diferenças entre previsões e valores reais. Ao contrário do MSE, ele não eleva ao quadrado os erros, tratando todos os desvios igualmente.O MAE é particularmente útil quando se deseja evitar penalizar muito os erros grandes. Ele fornece uma medida direta da média dos erros de previsão, oferecendo uma perspectiva clara sobre o desempenho real do modelo em termos de erro absoluto.

4.4.4 Coeficiente de Determinação (R^2)

O R^2 é uma medida estatística que representa a proporção da variação na variável dependente que é previsível a partir das variáveis independentes.
Um valor de R^2 próximo de 1 indica que uma grande proporção da variação na variável dependente é explicável pelas variáveis independentes no modelo. Um R^2 baixo indica que o modelo não explica muito da variação dos dados. O R^2 é especialmente útil para comparar a eficácia de diferentes modelos em termos de quão bem eles ajustam os mesmos dados observados.

4.4.2 Resultados da Validação

Os resultados da validação do modelo foram os seguintes:

Para garantir a robustez e reprodutibilidade, o *pipeline* completo foi implementado com os seguintes passos adicionais:

1. Avaliação do Melhor Modelo:Após encontrar o melhor parâmetro, o modelo foi reavaliado tanto no conjunto de treinamento quanto no conjunto de testes. Isso foi feito para garantir que o ajuste do parâmetro tenha efetivamente melhorado o desempenho do modelo e não apenas ajustado ao conjunto de treinamento.
2. Previsão e Validação Cruzada:Para verificar a estabilidade do modelo, foi realizada uma validação cruzada com 10 folds, calculando o MSE médio para garantir que o modelo generalize bem para dados não vistos.

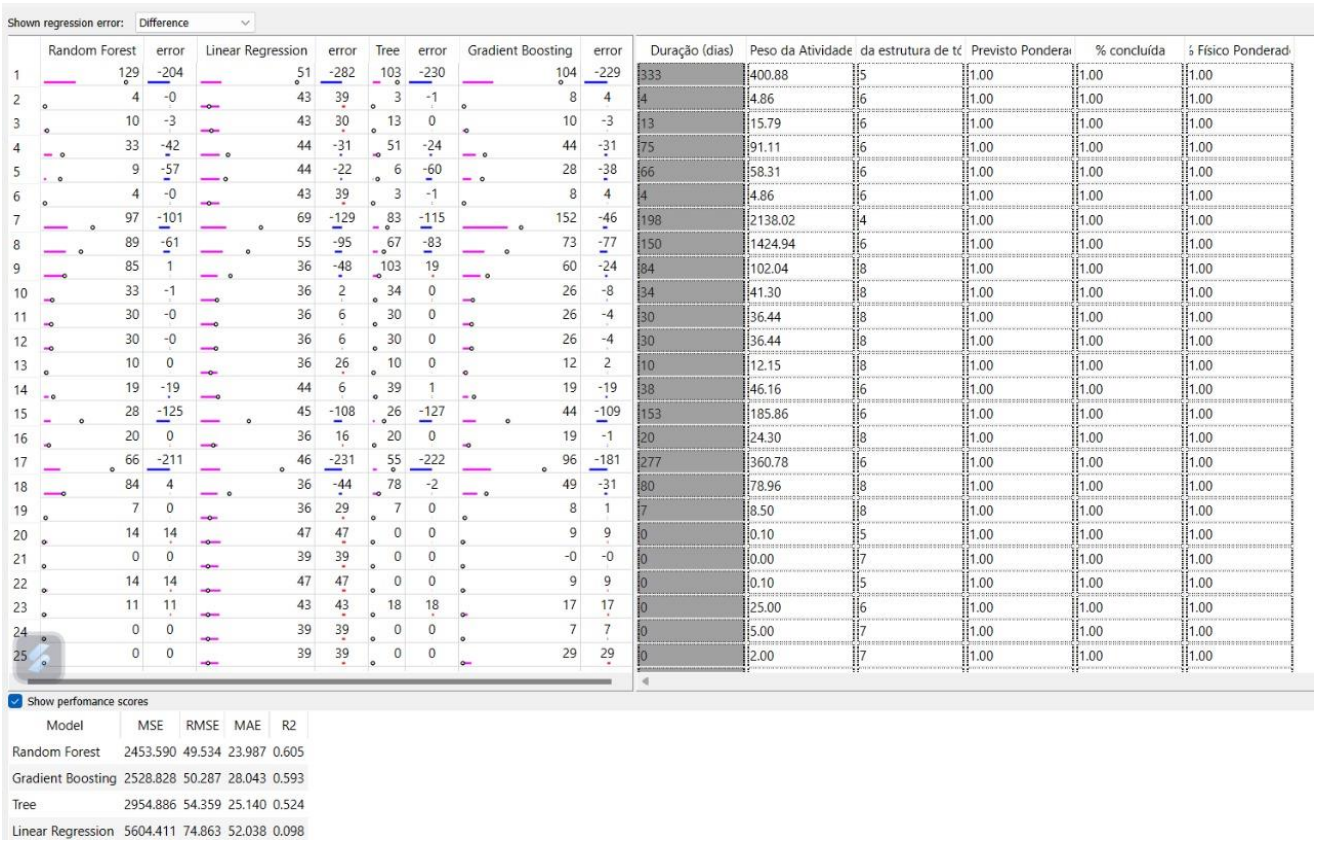

Figura 4 – Resultados da Validação

A tabela acima demonstra não só os resultados obtidos, mas também os desvios linha a linha do projeto. Colunas de Modelo: A tabela mostra os erros de previsão para quatro diferentes modelos de *machine learning* — Random Forest, Linear Regression, Decision Tree (Tree) e Gradient Boosting e viabiliza a análise de diversas perspectivas:

1. Erros de Previsão: Cada modelo tem uma coluna indicando o erro de previsão para cada entrada. O erro é calculado como a diferença entre o valor previsto pelo modelo e o valor real (observado).
2. Colunas de Dados: Há colunas para Duração (dias), Peso da Atividade, Nível da estrutura de tarefas, % Concluída, e Físico Ponderado. Estas colunas representam os atributos usados para alimentar os modelos.
3. Valores de Desempenho do Modelo: Na parte inferior, há um resumo das métricas de desempenho para cada modelo, incluindo MSE (Erro Quadrático Médio), RMSE (Raiz do Erro Quadrático Médio), MAE (Erro Absoluto Médio) e R^2 (coeficiente de determinação).

5. RESULTADOS E DISCUSSÃO

Esses dados mostram que os modelos Random Forest e Gradient Boosting apresentam desempenhos comparáveis com um MSE e RMSE mais baixos e um coeficiente R^2 de 0.605 e 0.593, respectivamente. sugerindo uma boa capacidade de previsão dos valores reais. No entanto, a Regressão Linear apresentou um desempenho notavelmente inferior, de 0.098, indicando um ajuste inadequado aos dados. A Árvore de Decisão também mostrou um desempenho razoável, próximo ao dos modelos de melhor desempenho. Diversos fatores afetam estes resultados, dentre eles:

1. Qualidade e Quantidade dos Dados: A precisão do modelo é altamente dependente da qualidade e quantidade dos dados utilizados. Se os dados do cronograma 114 contêm variabilidade significativa e casos atípicos, isso pode influenciar os resultados. A coleta de dados adicionais e a melhoria da qualidade dos dados podem ajudar a reduzir o overfitting.
2. Seleção de Atributos: A seleção de atributos é crucial para o desempenho do modelo. A análise da influência dos atributos, como Peso da Atividade, Concluída, e Nível da Estrutura, sobre os erros de previsão pode revelar quão sensíveis os modelos são às mudanças nessas variáveis. Por exemplo, atividades com maior Peso da Atividade e maior Concluída parecem ter previsões mais precisas, sugerindo que esses atributos são bem capturados pelos modelos de Random Forest e Gradient Boosting.
3. Tratamento de Dados Faltantes: A imputação de valores faltantes é uma etapa importante. Utilizou-se a média para variáveis numéricas e a moda para variáveis categóricas. No entanto, a estratégia de imputação deve ser escolhida com base na distribuição dos dados e no contexto do problema.
4. Escalonamento de Variáveis: Padronizar as variáveis numéricas garante que todas estejam na mesma escala, o que é essencial para algoritmos de regressão linear e Lasso. Sem esse passo, variáveis com valores maiores poderiam dominar o modelo.
5. Aplicabilidade Prática: O modelo foi desenvolvido para prever a duração das atividades no cronograma 113 com base nos dados do cronograma 114. A aplicabilidade prática deste modelo depende de como bem os dados de 114 representam os dados futuros de 113. Se houver diferenças significativas nos contextos dos dois cronogramas, o desempenho do modelo pode ser comprometido.
6. Disparidade nos Erros de Previsão: Nota-se uma variação significativa nos erros de previsão entre os diferentes modelos. Por exemplo, o modelo de Random Forest e Gradient Boosting mostra consistentemente menores erros em comparação com a Regressão Linear e Decision Tree em muitas das observações. Essa consistência sugere que os modelos baseados em

ensino, como Random Forest e Gradient Boosting, são mais eficazes em capturar a complexidade dos dados e em generalizar sobre o conjunto de testes.

6. CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação investigou o uso potencial de técnicas de *machine learning* para estimar prazos de conclusão de projetos. Utilizando a metodologia de Design Science Research, onde modelos de *machine learning* foram desenvolvidos e aplicados a dados de cronogramas de projetos. Os resultados indicaram que o modelo de regressão treinado no cronograma 114 e testado no cronograma 113 obteve, mostrando uma capacidade razoável de previsão, embora com um MAE de 23.98 e um R^2 de 0.605, indicando que há margem para melhorias, porém, os resultados iniciais podem ser considerados satisfatórios.

Os achados sugerem que, embora os modelos de *machine learning* possuam potencial significativo para melhorar as estimativas de prazos de projetos, a inclusão de mais variáveis que influenciam a duração das atividades e a exploração de modelos mais avançados podem aumentar ainda mais a precisão das previsões. Em particular, variáveis como complexidade das tarefas, recursos alocados e dependências entre atividades mostraram-se promissoras para futuras pesquisas.

Apesar das contribuições valiosas deste estudo, é possível verificar algumas limitações. Primeiramente, a variabilidade inerente das atividades de projetos não foi totalmente capturada pelos modelos. Em segundo lugar, embora os resultados tenham sido promissores no contexto de projetos usando o framework AWP (Advanced Work Packaging), são ainda incipientes e menos eficazes quando aplicados a outros frameworks e metodologias de projeto, como Scrum, PRINCE2 e Kanban, que apesar de terem maior flexibilidade na adaptação às mudanças, não possuem padronizações. Ademais, as iterações curtas e o foco em entregas incrementais corrobora para tal incompatibilidade.

Quanto às perguntas iniciais de investigação, conclui-se que na busca por melhorar as estimativas de prazos em projetos, diversos desafios emergem como barreiras significativas. A previsão de interrupções e atrasos, por exemplo, continua sendo uma tarefa complicada, principalmente devido à subjetividade e ao otimismo frequentemente presentes nas estimativas de duração das tarefas fornecidas por indivíduos. Além disso, a complexidade de gerenciar simultaneamente múltiplas variáveis e restrições, como a disponibilidade de recursos e as interdependências entre tarefas, acrescenta outra camada de dificuldade ao processo.

No contexto do gerenciamento de projetos, os modelos de *machine learning* oferecem um caminho promissor para superar esses obstáculos. Através da análise de dados históricos de projetos, esses modelos têm a capacidade de identificar padrões de atrasos e sucessos, proporcionando estimativas de duração das tarefas mais alinhadas com a realidade. Além disso, esses sistemas são capazes de processar uma variedade de variáveis complexas de forma simultânea e de ajustar as previsões à medida que novas informações sobre o escopo ou os recursos do projeto se tornam disponíveis.

No entanto, a implementação de modelos de *machine learning* no gerenciamento de projetos não está isenta de desafios. A eficácia desses modelos depende significativamente da disponibilidade de grandes volumes de dados históricos de alta qualidade, uma condição que pode ser difícil de atender em organizações com práticas inadequadas de documentação e registro. Além disso, existe uma resistência natural à mudança por parte das equipes de projeto, muitas das quais podem ser céticas quanto à capacidade de máquinas em prever de forma precisa os complexos processos humanos envolvidos em projetos. Assim, embora os modelos de *machine learning* apresentem uma solução viável e inovadora para os desafios de estimativa de prazos em projetos, a sua implementação requer não apenas uma base técnica sólida, mas também uma mudança cultural dentro das organizações para que sejam plenamente eficazes, e integrar tal modelo à ferramentas de gerenciamento de projetos utilizados em determinada organização, aliada à sua revisão periódica do modelo, podem levar a um considerável aumento da assertividade nas estimativas.

Ademais, recomenda-se que em trabalhos futuros alguns aspectos importantes sejam observados:

1. Diversificação dos Dados: Estudos futuros devem considerar uma maior diversidade de dados de projetos, abrangendo diferentes setores, tamanhos e complexidades. Isso permitirá uma melhor generalização dos modelos de *machine learning*.
2. Incorporação de Mais Variáveis: Recomenda-se a inclusão de variáveis adicionais que possam impactar a duração das atividades, como detalhes sobre a equipe, condições climáticas (para projetos de construção) e outras condições externas.
3. Exploração de Modelos Avançados: A pesquisa futura deve explorar modelos de *machine learning* mais avançados, como redes neurais profundas e modelos de boosting, que podem capturar complexidades e interações não lineares entre as características do projeto.
4. Comparação entre Frameworks: Estudos comparativos entre diferentes frameworks de gerenciamento de projetos devem ser conduzidos para avaliar a eficácia dos modelos de *machine learning* em cada contexto específico. Isso ajudará a identificar quais metodologias são mais compatíveis com as técnicas de *machine learning*.

Em suma, esta pesquisa contribui para a literatura de gerenciamento de projetos e *machine learning* ao demonstrar a aplicabilidade e os desafios das técnicas de *machine learning* na previsão de prazos de projetos. Apesar dos resultados promissores com o uso do framework AWP, há uma necessidade

clara de mais pesquisas para adaptar e validar esses modelos em outros contextos e metodologias de projeto.

- **REFERÊNCIAS BIBLIOGRÁFICAS**

Alpaydin, E. (2004). Introduction to machine learning. MIT Press.

Blum, A. (2007). Machine learning theory. Carnegie Mellon University, School of Computer Science.

Boden, M. A. (1996). Artificial intelligence. Elsevier.

Bunge, M. (1997). Ciencia, técnica y desarrollo. Buenos Aires: Sudamericana.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Brown, S. (2021). Título do artigo. MIT Sloan. Recuperado de https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Dresch, A., Lacerda, D. P., & Antunes Júnior, J. A. V. (2015). Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia. Maria Eduarda Fett Tabajara.

Gray, C., & Larson, E. (2009). Gerenciamento de Projetos: O processo gerencial (4ª ed.). AMGH Editora Ltda.

Crawford, J. K. (2002). The Strategic Project Office: A Guide to Improving Organizational Performance. New York: Marcel Dekker Inc.

De Fernandes Teixeira, J. (2014). Inteligência artificial. Pia Sociedade de São Paulo: Editora Paulus.

Han, J., & Kamber, M. (2006). Data Mining: Concepts and Techniques (2ª ed.). Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. New York: Springer.

Haykin, S. (2001). Redes neurais: Princípio e prática. Bookman.

Hurwitz, J., & Kirsch, D. (2018). Machine Learning For Dummies, IBM Limited Edition. United States of America. ISBN: 978-1-119-45494-6.

Kaelbling, L. P., & Littman, M. L. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 237-285.

Kerzner, H. (2002). Gestão de Projetos: As melhores práticas (Trad. Marco Antonio Viana Borges, Marcelo Klippel e Gustavo Severo de Borba). Porto Alegre: Bookman.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer.

March, S. T., & Smith, G. F. (1995). Design and natural science research in Information Technology. Decision Support Systems, 15, 251-266.

Maxim, B., & Pressman, R., S. (2016). Engenharia de software: Uma abordagem profissional. Porto Alegre: AMGH Editora Ltda.

Mitchell, T. (1997). Machine learning. McGraw Hill.

Natarajan, B. K. (1991). Machine learning: A theoretical approach. Morgan Kaufmann.

Pondel, J., & Pondel, M. (2016). The concept of project management platform using BI and Big Data technology. In Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1, 166-173. Recuperado em 25 de maio de 2024, de https://www.example.com (ajustar URL conforme o local de acesso real).

PMP, G. R. (2017). Even More Schedule for Sale: Advanced Work Packaging, for Construction Projects. AuthorHouse.

Project Management Institute. (2013). Guia PMBOK® (7ª ed.).

Russell, S., & Norvig, P. (2009). Artificial Intelligence: A Modern Approach (3ª ed.). Prentice Hall Press.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3), 210–229.

Simon, H. A. (1996). The Sciences of the Artificial (3ª ed.). MIT Press.

Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. MIT Press.

Van Aken, J. E. (2004). Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. Journal of Management Studies, 41(2), 219-246.

Webster, F. (1999). Setting the stage for a new profession. PM Network, April.

Wen, J., Li, S., Hu, Q., & Cai, Y. (2012). Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology, 54(1), 41-59.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques (4ª ed.). Morgan Kaufmann.