

UNIVERSIDADE ESTADUAL PAULISTA - UNESP
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Heitor Nunes Rosa

**Predição do Desempenho de Hélices de Pequeno Porte com
XGBoost: Efeitos do processo de Imputação da Solidez por
métodos de regressão**

Ilha Solteira, SP

2022

UNIVERSIDADE ESTADUAL PAULISTA - UNESP
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Predição do Desempenho de Hélices de Pequeno Porte com XGBoost: Efeitos do processo de Imputação da Solidez por métodos de regressão

Trabalho de Graduação apresentado à Faculdade de Engenharia de Ilha Solteira - UNESP - como parte dos requisitos para obtenção do título de Engenheiro Mecânico.

Heitor Nunes Rosa

Discente

Prof. Emanuel Rocha Woiski

Orientador

Ilha Solteira, SP

2022

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Rosa, Heitor Nunes.

R788p

Predição do desempenho de hélices de pequeno porte com XGBoost: efeitos do processo de Imputação da solidez por métodos de regressão / Heitor Nunes Rosa. -- Ilha Solteira: [s.n.], 2022

51 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2022

Orientador: Emanuel Rocha Woiski

Inclui bibliografia

1. Aeronáutica. 2. Hélices. 3. Modelo substituto. 4. Machine learning.

João Josué Barbosa,
Serviço Técnico de Biblioteca e Documentação
Diretor Técnico
CRB 8-5642

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE ENGENHARIA - CAMPUS DE ILHA SOLTEIRA

CURSO DE ENGENHARIA MECÂNICA

ATA DA DEFESA – TRABALHO DE GRADUAÇÃO

TÍTULO: Predição do Desempenho de Hélices de Pequeno Porte com XGBoost: Efeitos do processo de Imputação da Solidez por métodos de regressão

ALUNO: Heitor Nunes Rosa

RA: 152054819

ORIENTADOR: Emanuel Rocha Woiski

Aprovado (X) - Reprovado () pela Comissão Examinadora

Comissão Examinadora:



Prof.

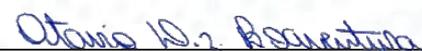
Emanuel Rocha Woiski (Orientador)

Prof.



Amarildo Tabone Paschoalini

Eng.



Otávio Duarte Zotelli Boaventura

Ilha Solteira(SP) 05 de março de 2022.

Agradecimentos

Agradeço a Deus e a Nossa Senhora pela Vossa presença e amparo em minhas dificuldades e decisões durante toda essa trajetória que foi a graduação de Engenharia Mecânica.

Aos meus pais, Angela Maria Nunes e Luciano Gualberto Van Haute Rosa, por todo apoio financeiro e emocional durante os cinco anos de faculdade.

Aos meus amigos de morada, Gustavo da Silva, Guilherme Santana, Rafael Martarelli, Leonardo Suter, pela paciência e amizade que proporcionaram momentos mais leves a minha estada em Ilha Solteira, e apoio além de necessário nas piores situações dessa caminhada.

Aos meus colegas de classe pelos louvores e sofrimentos compartilhados durante a graduação, em particular a André Verzoto, Natália Sayuri, Ana Carloni, Pedro Zanovelly e Marvin Pereira, amigos insubstituíveis, seja para trabalho ou companherismo.

A todos os membros, sem exceção, da Equipe Zebra, aos que vieram antes de mim e aos após minha saída. Sem a Equipe, eu não teria este Trabalho de Graduação e muitos menos poderia ter me desenvolvido como profissional e pessoa.

Aos meus professores de Graduação, em particular a Fábio Chavarette, Mara Lúcia Lopes, Márcio Antonio Bazani, Miguel Ângelo Menezes, pelos seus ensinamentos fundamentais para minha formação como Engenheiro. Em mais particular, ao Emanuel Rocha Woiski, meu orientador, por todas as nossas conversas, discussões e correções, dentro e fora do escopo do trabalho, levo-o como amigo.

A minha namorada, Bárbara Carvalho, por toda a paciência, dedicação, carinho e amor nessa etapa final da faculdade.

Por fim, e não menos importante, ao meu falecido irmão Hugo Nunes Rosa, pela sua admiração e observância a mim, e ao meu filho que esta para vir, Hugo Carvalho Rosa, ao qual serei um pai exemplar e profundo amigo, como muitos foram para mim.

Resumo

Este trabalho tem como objetivo principal a elaboração de um modelo substituto para a predição do desempenho de hélices de pequeno porte com a aplicação de métodos de *Machine Learning*. Os modelos foram projetados utilizando a linguagem de programação *Python*, com a utilização do algoritmo de regressão *XGBoost*, e com base no banco de dados disponibilizados pela Universidade de Illinois em Urbana-Champaign. Fez-se também uma manipulação de dados para o cálculo da Solidez, para se avaliar sua influência no desempenho da hélice. Neste trabalho, lidou-se com dados faltantes, que, apesar do algoritmo escolhido ser robusto, aplicou-se um método de imputação para se verificar se haveria uma melhoria em seu desempenho. Os hiperparâmetros do modelo foram refinados por um processo de otimização bayesiana. Foram obtidos modelos satisfatórios para o desenvolvimento de projetos preliminares robustos. Este modelo será útil para proporcionar uma mais rápida e eficiente seleção e projeto de hélice.

Palavras-chave: Aeronáutica, Hélices, Modelo Substituto, *Machine Learning*.

Abstract

This work has as main objective the elaboration of a surrogate model for the prediction of the performance of small propellers with the application of Machine Learning methods. The models were designed using the Python programming language, using the XGBoost regression algorithm, and based on the database provided by the University of Illinois at Urbana-Champaign. A data manipulation was also carried out to calculate the solidity, in order to evaluate its influence on the propeller performance. In this work, we dealt with missing data, which, despite the chosen algorithm being robust, an imputation method was applied to verify if there was an improvement in its performance. The model hyperparameters were tuned by a Bayesian optimization process. Satisfactory models were obtained for the development of preliminary designs of propellers with a relatively small prediction interval. This model will be useful to provide a better design phase, providing a faster and more efficient selection of propellers.

Keywords: Aeronautics, Propeller, Surrogate Model, Machine Learning.

Lista de Figuras

1	VANT efetuando mapeamento aéreo para extração de informações geográficas.	12
2	Representação Esquemática da funcionalidade do modelo substituto.	13
3	Eficiência da hélice em função de sua razão de avanço. O gráfico denota diferentes inclinações para a mesma hélice.	17
4	Esquematização do parâmetro solidez das hélices.	17
5	Comparativo entre Programação Tradicional e <i>Machine Learning</i>	18
6	Parcelas do erro de predição em função da complexidade do modelo.	21
7	Erros de validação e treino em função de sua complexidade.	21
8	Validação Cruzada com 20% das instâncias para teste, 20% do restante para validação.	22
9	Três condições usuais do comportamento da curva de aprendizado: (a) <i>underfitting</i> ; (b) ideal; (c) <i>overfitting</i>	22
10	Estrutura de uma árvore de decisão	26
11	Funcionamento do <i>Gradient Boosting</i>	27
12	Influência da taxa de aprendizado no treinamento do modelo	31
13	Coefficientes de Tração C_T e de Potência C_W , como função da Razão de Avanço J , parametrizados de forma contínua pelo Passo por Diâmetro P/D , para as instâncias da classe APC Sport.	34
14	Coefficientes estáticos C_{T_o} e C_{W_o} por Rotação N parametrizados pelo Passo por Diâmetro P/D , para hélices da família APC Sport.	35
15	Regressões lineares para os Coeficientes de Tração dinâmico C_T e estático C_{T_o} , como função da Solidez.	36
16	Histogramas dos dados originais e após a aplicação da imputação.	36
17	Dispersão dos dados originais e após a aplicação da imputação.	37
18	Histórico de otimização dos hiperparâmetros (Três Parâmetros).	38
19	Importância dos hiperparâmetros para a otimização (Três Parâmetros).	38
20	Convergência dos hiperparâmetros por iteração (Três Parâmetros).	39
21	Importância das variáveis independentes (Três Parâmetros).	40
22	Curva de aprendizado do modelo (Três Parâmetros).	40
23	Resíduo das amostras de treino e teste (Três Parâmetros).	40
24	Histórico de otimização dos hiperparâmetros (Quatro Parâmetros Sem Imputação).	41
25	Importância dos hiperparâmetros para a otimização (Quatro Parâmetros Sem Imputação).	41
26	Convergência dos hiperparâmetros por iteração (Quatro Parâmetros Sem Imputação).	42

27	Importância das variáveis independentes (Quatro Parâmetros Sem Imputação).	42
28	Curva de aprendizado do modelo (Quatro Parâmetros Sem Imputação). . .	43
29	Resíduo das amostras de treino e teste (Quatro Parâmetros Sem Imputação). 43	
30	Histórico de otimização dos hiperparâmetros (Quatro Parâmetros Com Imputação).	44
31	Importância dos hiperparâmetros para a otimização (Quatro Parâmetros Com Imputação).	44
32	Convergência dos hiperparâmetros por iteração (Quatro Parâmetros Com Imputação).	45
33	Importância das variáveis independentes (Quatro Parâmetros Com Imputação).	46
34	Curva de aprendizado do modelo (Quatro Parâmetros Com Imputação). . .	46
35	Resíduo das amostras de treino e teste (Quatro Parâmetros Com Imputação). 46	

Lista de Tabelas

1	Distribuições iniciais dos hiperparâmetros para a aplicação do algoritmo TPE.	31
2	Valores do quantil para cada intervalo de predição	32
3	Correlação de Pearson entre as variáveis	35
4	Correlação de Pearson entre as variáveis estáticas	35
5	Parâmetros das distribuições original e após a aplicação da imputação. . .	37
6	Hiperparâmetros Finais (Três Parâmetros).	39
7	Hiperparâmetros Finais (Quatro Parâmetros Sem Imputação).	42
8	Hiperparâmetros Finais (Quatro Parâmetros Com Imputação).	45
9	REQM obtidas pelas três aproximações consideradas	47

Sumário

1	Introdução	12
1.1	Motivação	12
1.2	Objetivos	13
2	Estado da Arte	14
2.1	Programas de análise	14
2.1.1	PropSelector	14
2.1.2	JBLADE	14
2.1.3	QPROP Propeller/Windmill	14
2.1.4	XROTOR	14
2.2	Ensaio Experimentais	15
2.2.1	Banco de Dados da UIUC	15
3	Fundamentação Teórica	16
3.1	Características da hélice	16
3.2	<i>Machine Learning</i>	18
3.2.1	Compromisso entre viés e variância	20
3.2.2	Conjuntos de treino, validação e teste	20
3.2.3	Validação Cruzada	21
3.2.4	<i>Curva de Aprendizado</i>	22
3.2.5	Dados Faltantes	23
3.2.6	Imputação de dados	24
3.2.7	Otimização de Hiperparâmetros	25
3.3	Algoritmos de Regressão	25
3.3.1	Árvores de Regressão	25
3.3.2	<i>Gradient Boosting</i>	27
3.3.3	<i>XGBoost</i>	28
3.3.4	Hiperparâmetros	28
3.3.5	Importância de atributos	29
4	Metodologia e Desenvolvimento do Trabalho	30
4.1	Procedimento Aplicado	30
4.2	<i>Frameworks</i> Utilizados	33
5	Resultados e Discussão	34
5.1	Análise Exploratória	34
5.2	Imputação de dados	36
5.3	Modelagem: Três Parâmetros	37
5.4	Modelagem: Quatro Parâmetros Sem Imputação	41

5.5	Modelagem: Quatro Parâmetros Com Imputação	44
6	Conclusão e Trabalhos Futuros	48
	Referências	49

1 Introdução

1.1 Motivação

Veículo aéreo não-tripulado, ou VANT, é um termo genérico que identifica uma aeronave que pode voar sem tripulação, normalmente projetada para operar em situações perigosas e repetitivas em regiões consideradas hostis ou de difícil acesso (FURTADO et al., 2008). Por apresentar baixo custo e flexibilidade, VANTs vem sendo amplamente utilizados para diversas aplicações. (MATIAS; GUZATTO; SILVEIRA, 2014) propuseram uma metodologia de extração de informações geográficas com base em fotografias aéreas obtidas por VANTs. Cardoso, Queiros e Santos (2018) discorrem sobre a possibilidade de se utilizar drones como ferramentas de monitoramento ambiental da Floresta Amazônica. Doherty e Rudol (2007) apresentam a situação atual de VANTs para situações de Busca e Salvamento, dividindo em duas etapas, identificação de corpos e entrega de medicamentos e suprimentos às vítimas.

Figura 1 – VANT efetuando mapeamento aéreo para extração de informações geográficas.

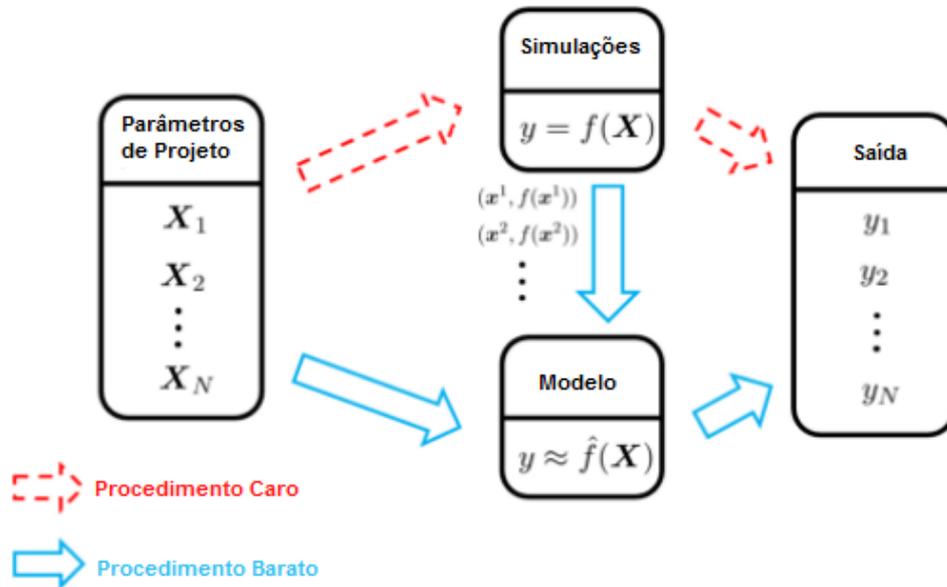


Fonte: Adonai Topografia (2017)

Para o projeto de um VANT, algo que deve ser levado em consideração é o seu sistema propulsivo. Boas predições de suas curvas de desempenho garantem um projeto preliminar robusto e confiável. Para isso, faz-se uso de métodos numéricos e experimentais, entretanto, ambos tem suas desvantagens. Modelos numéricos demandam tempo, custo computacional e uma geometria definida da hélice para serem aplicados, além de um conhecimento avançado do projetista. Ensaio experimentais demandam túneis de vento e equipamentos calibrados para a medida correta das hélices. Outra alternativa se trata

de modelos substitutos, que consistem em modelos compactos para estimar resultados complexos, baseados em dados experimentais ou de simulações (SANTOS, 2018).

Figura 2 – Representação Esquemática da funcionalidade do modelo substituto.



Fonte: Adaptado de Keane, Forrester e Sobester (2008)

Assim, o projetista pode prever rapidamente o desempenho de uma hélice, escolhendo parâmetros ótimos para o projeto preliminar da hélice, ou tomar decisões como compra de hélices comerciais com base nos parâmetros fornecidos pelo fabricante.

1.2 Objetivos

Considerando a importância de um sistema propulsivo para o projeto de um VANT e as dificuldades de previsão de seu desempenho, neste trabalho pretende-se:

- Elaborar e aplicar um modelo substituto para a previsão do desempenho de hélices com métodos de *Machine Learning*, utilizando regressão.
- Avaliar a influência dos parâmetros no desempenho das hélices, em particular, da solidez.
- Verificar o desempenho de algoritmos robustos a dados faltantes, quando implementados vários procedimentos de imputação de dados, comparando-os entre si.
- Divulgar a aplicação de *Machine Learning* como ferramenta de projeto no contexto de Engenharia Mecânica.

2 Estado da Arte

Nesta seção, irá se discorrer sobre os programas paramétricos disponíveis para a análise e projeto de uma hélice. Ao fim da seção, um breve resumo dos ensaios experimentais conduzidos nessa área.

2.1 Programas de análise

2.1.1 PropSelector

PropSelector é um programa que foi desenvolvido por Brian Robert Gyles, que fornece como saída o desempenho de duas a quatro hélices de pás de aeromodelos e é baseado nas relações dos dados das hélices da Nota Técnica NACA No.698 (LESLEY, 1939). Existe uma versão estendida deste programa chamado *PropSelector Extended*, que permite a entrada de altitude e fornece mais valores de saída, como coeficiente de tração da hélice, número de Mach da ponta e inclinação a 75% de raio da pá da hélice (GYLES, 1999).

2.1.2 JBLADE

JBLADE é um código aberto de projeto e análise de hélice aberto desenvolvido na UBI, como parte de uma tese de doutorado. Ele se baseia na teoria do elemento de pá modificada (GLAUERT, 1983). Com JBLADE, podem-se estimar as curvas de desempenho de uma determinada hélice e, após a análise, os resultados são mostrados em uma interface gráfica, para tornar mais fácil construir e analisar as simulações (SILVESTRE; MORGADO; PASCOA, 2013).

2.1.3 QPROP Propeller/Windmill

QPROP é um programa de análise e projeto de hélices, criado pelo professor Mark J. Drela, de Massachusetts Institute of Technology (MIT), que é baseado em uma formulação aerodinâmica teórica que usa uma extensão da formulação clássica de elemento de pá e vórtice. QPROP mostra como saída a análise do desempenho de uma combinação moto propulsora (DRELA, 2007).

2.1.4 XROTOR

XROTOR é um programa que usado principalmente para projeto e análise de dutos e hélices de pontas livres. Ele contém algumas rotinas baseadas em menus que realizam uma variedade de funções, como: projeto de um rotor de perda mínima induzida; entrada solicitada de uma geometria de rotor arbitrária; modificação da geometria de um rotor; otimização de um rotor para perda induzida mínima; análise do desempenho de um rotor

com muitos parâmetros operacionais; efeitos de turbilhonamento de entrada; análise estrutural e correções para torção sob carga; previsões de ruído; interpolação da geometria para um raio de interesse; plotagem dos resultados da análise (DRELA; YOUNGREN, 2003).

2.2 Ensaios Experimentais

Comparados com a documentação sobre o desempenho da hélice para aviões em grande escala, os dados sobre hélices de pequeno porte não são muitos, porém, o interesse é crescente. Os testes realizados por Bailey (1978) documentaram sete modelos de avião de madeira *TopFliteTM* com hélices cujo diâmetro variava de 9 a 14 polegadas. Foi relatado que esses resultados mostraram eficiências da hélice 7,5% a 15% menor do que as hélices maiores de 36 polegadas de diâmetro com proporções de passo para diâmetro semelhantes testadas por Durand e Lesley (1923). Degradação semelhante no desempenho foi medida posteriormente por Bass (1986) para hélices maiores que 24 polegadas, e também por Asson e Dunn (1992), mostrando dados em dois modelos de madeira da marca Zinger de 14 polegadas de diâmetro. Merchant e Miller (2006) conduziram testes em 30 hélices de aeromodelo variando em diâmetro de 6 a 22 polegadas, mas apenas um subconjunto das medições em sete hélices foi disponibilizado. Ol, Zeune e Logan (2008) realizou medições em muitas hélices destinadas ao uso em VANTs e fez detalhadas comparações com a análise, revelando efeitos importantes dos baixos números de Reynolds. Por fim, Brandt e Selig (2011) conduziram experimentos em 136 hélices de pequeno porte, mantendo todos dados documentados e disponibilizados no site de Illinois em Urbana - Champaign (UIUC). Em razão deste trabalho fazer uso desses dados, este banco de dados será descrito com mais detalhes.

2.2.1 Banco de Dados da UIUC

No trabalho conduzido por Brandt e Selig (2011), foram realizados testes em 79 hélices bi-pás, as quais tinham um diâmetro variando de 7 a 19 polegadas. Essas hélices eram de marcas diferentes: Aeronaut, APC, Graupner, GWS, Kavon, Kyosho, Master Airscrew, Rev up e Zingali. Foram obtidos os Coeficientes de Tração e Potência, bem como a Eficiência, para uma dada Razão de Avanço. Foram, também, avaliados os respectivos valores estáticos (com Razão de Avanço nula). Vale ressaltar que foram registrados os valores de rotação das hélices, a fim de se averiguarem os possíveis efeitos de compressibilidade com o aumento do número de Reynolds, pois este é definido proporcionalmente à rotação da hélice, conforme se verá na eq. 6.

3 Fundamentação Teórica

Nesta seção, será comentada sobre a fundamentação teórica relacionada à hélice e métodos aqui utilizados para a predição do seu desempenho.

3.1 Características da hélice

Hélices são denominadas asas rotativas, pois produzem sustentação e arrasto por causa da sua rotação. Em geral, uma hélice é definida por dois parâmetros, o seu diâmetro e o seu passo. Convencionalmente, o passo de uma hélice é definido pelo seu avanço em uma dada rotação, em um raio de referência, conforme a eq. 1, em que P , $r_{3/4}$ e $\beta_{3/4}$ são, respectivamente, o passo, o raio, a corda e a inclinação para 75% de seu raio total (ROSA; TOPOROSKI, 2006).

$$P = 2\pi r_{3/4} \tan(\beta_{3/4}) \quad (1)$$

Como um exemplo, para uma hélice definida como 12x6, 12 representa seu diâmetro e 6 seu passo, tradicionalmente utilizado em polegadas.

O desempenho de uma hélice é avaliado por meio de três parâmetros: o Coeficiente de Potência C_W , eq. 2, o Coeficiente de Tração C_T , eq. 3, e a Eficiência η , eq. 4, que dependem principalmente da Razão de Avanço J , eq. 5, do número de Reynolds Re , eq. 6, e da sua geometria.

$$C_W = \frac{W}{\rho n^3 D^5} \quad (2)$$

$$C_T = \frac{T}{\rho n^2 D^4} \quad (3)$$

$$\eta = J \frac{C_T}{C_W} \quad (4)$$

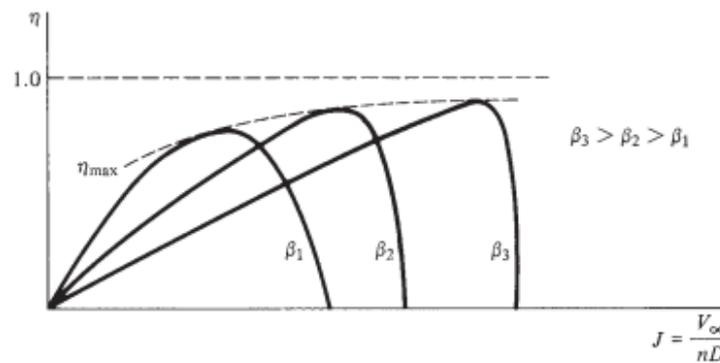
$$J = \frac{V}{nD} \quad (5)$$

$$Re = \frac{\rho n r_{3/4} c_{3/4}}{\mu} \quad (6)$$

V é a velocidade relativa do fluxo de ar sobre a hélice, n é a rotação da hélice, D é o seu diâmetro, ρ é a densidade do ar, $c_{3/4}$ é a corda para 75% de seu raio total $D/2$, μ é a viscosidade do ar, T a tração e W a potência, em unidades consistentes, para que os valores resultem adimensionais.

Hélices com passos maiores e com V e D fixados, ou seja, com inclinações $\beta_{3/4}$ maiores, tendem a ser mais eficientes para baixas rotações (maiores J), enquanto que hélices com passos menores são mais eficientes para altas rotações, conforme apresentado na Fig. 3.

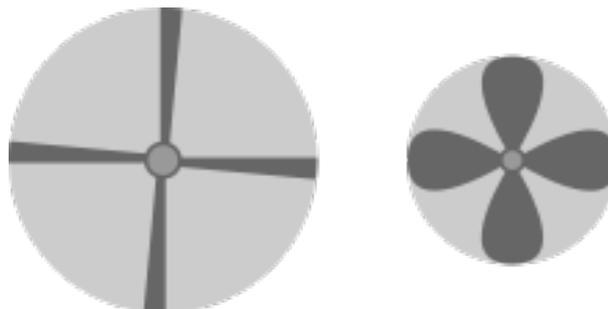
Figura 3 – Eficiência da hélice em função de sua razão de avanço. O gráfico denota diferentes inclinações para a mesma hélice.



Fonte: Anderson e Bowden (2005)

Outro parâmetro importante, mas geralmente negligenciado, é a Solidez da hélice. Solidez consiste na razão entre área projetada pela hélice e a área varrida pela sua rotação. Uma esquematização de duas hélices com valores contrastantes para a Solidez pode ser vista na fig. 4, sendo a hélice à direita, aquela com maior Solidez.

Figura 4 – Esquematização do parâmetro solidez das hélices.



Fonte: Wikipedia (2021)

Stack et al. (1950) conduziram experimentos envolvendo a Solidez, chegando à conclusão que o seu aumento pode indicar um melhor desempenho da hélice em rotações elevadas. Duquette e Visser (2003) investigaram numericamente as implicações do aumento da Solidez em turbinas de eixo horizontal. Para todos os casos examinados, obteve-se um aumento no Coeficiente de Potência.

Para o cálculo da solidez σ , utiliza-se a eq. 7.

$$\sigma = \frac{4B}{\pi D^2} \int_0^{D/2} c(r) dr \quad (7)$$

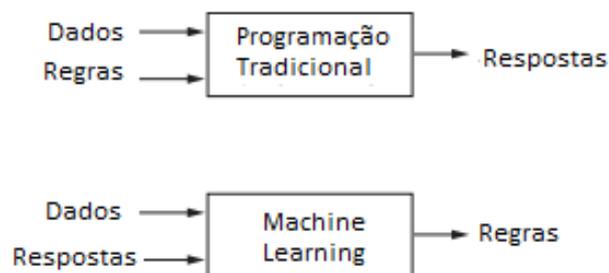
B é o número de pás. Dada a natureza da aferição da distribuição da corda $c(r)$ para o banco de dados utilizado, uma integração numérica deve ser aplicada.

3.2 *Machine Learning*

Machine Learning, ou aprendizado de máquinas, implica o uso de algoritmos computacionais com vistas à obtenção automatizada de padrões a partir de conjuntos de dados. Segundo Chollet (2021), um sistema de aprendizado de máquina é treinado, em vez de programado explicitamente. Alimentado com um certo número de exemplos relevantes para uma dada tarefa ou problema a ser resolvido, o sistema recupera padrões estatísticos naqueles exemplos, permitindo que estabeleça por si mesmo regras para automatizar a tarefa.

A aplicação de *Machine Learning* está ligada ao aprendizado a partir dos próprios dados, de regras para produzir previsões ou inferências, ou ainda à compreensão das relações entre os dados, em contraposição aos métodos tradicionais de programação, em que as regras são previamente estabelecidas e funcionam como filtro sobre os dados, conforme ilustrado na Fig. 5.

Figura 5 – Comparativo entre Programação Tradicional e *Machine Learning*.



Fonte: Adaptado de Chollet (2021)

Machine Learning pode ser aplicada a uma enorme diversidade de problemas, tais como filtragem de spam, processamento de linguagem natural, motores de busca, diagnósticos médicos, locomoção de robôs, veículos autônomos e muitos outros (ANZAI, 2012).

Pode ser utilizada, também, em tarefas na área de Engenharia, por exemplo, na predição de tensões compressivas de concreto (YEH, 1998) ou em classificação de falhas em placas de aço (BUSCEMA; TERZI; TASTLE, 2010).

Os dados, adequados para uma aplicação dos algoritmos computacionais de *Machine Learning*, exigem, geralmente, uma estrutura matricial, com os seguintes elementos e respectivas nomenclaturas:

- **Atributos:** As colunas, também conhecidos como *features*, ou variáveis, são valores que descrevem certas características de cada instância, ou observação. Eles podem ser classificados como preditores (ou variáveis independentes) e alvos (ou variáveis dependentes). Preditores são os atributos que, em todo ou em parte, serão utilizados pelos algoritmos de aprendizado para realizar uma estimativa representada pelos alvos. Seus valores podem ser categóricos ordinais ou cardinais, ou numéricos discretos ou contínuos.
- **Instâncias:** As linhas, também conhecidas como observações, vetor de entrada ou vetor de atributos, as instâncias são descritas pelo conjunto, ou de apenas parte, dos valores dos atributos da linha correspondente.

De acordo com a estrutura dos dados e a questão da qual quer-se obter uma resposta, um problema de *Machine Learning* pode ser classificado em categorias, conforme Chollet (2021):

- **Aprendizado Supervisionado:** A estrutura dos dados para o treinamento contém os valores dos alvos. Portanto, a função do algoritmo é a de encontrar uma relação entre as variáveis independentes - as *features* - e os alvos dependentes e aplicar o mapeamento para efetuar predições vinculando instâncias novas da mesma população, aos alvos. Exemplo: Predição de valores de venda de casas baseados em seus diversos atributos.
- **Aprendizado Não-Supervisionado:** A estrutura de dados para o treinamento não possui valores para os alvos. Então, o algoritmo deve encontrar as relações ou padrões subjacentes entre as instâncias, com base simplesmente em seus atributos e, com isso, determinar o lugar próprio para cada nova instância da mesma população. Exemplo: Classificação dos perfis de clientes de uma loja.
- **Aprendizado por Reforço:** O algoritmo recebe informações sobre seu ambiente e aprende a escolher ações que maximizarão recompensas e minimizarão penalidades. Exemplo: Veículos autônomos.
- **Aprendizado Semi-Supervisionado:** Uma parte significativa das instâncias não possuem valores dos alvos. Exemplo: Detecção de *spams* em uma caixa eletrônica, onde apenas temos conhecimento de alguns emails que não são *spams*.

Outra categorização em que o aprendizado pode ser classificado é quanto a sua tarefa, ou problema a ser resolvido:

- **Classificação:** Aprendizado Supervisionado em que as instâncias estão distribuídas entre os alvos, que são valores categóricos denominados classes, O objetivo é a atribuição de qualquer nova instância da mesma população a uma dada classe do conjunto de classes.
- **Regressão:** Também é um Aprendizado Supervisionado, tendo como alvo valores contínuos e cujo objetivo é obter-se uma função entre os atributos e os alvos. Note-se que no presente trabalho, lidar-se-ão apenas com algoritmos de regressão.
- **Clusterização ou Agrupamento:** Aprendizado Não-Supervisionado, em que as instâncias são agrupadas em conjuntos, devido aos valores de seus próprios atributos.

Pode-se observar pelos exemplos o quão abrangente a aplicação de *Machine Learning* pode ser. No entanto, algumas considerações devem ser levadas em conta no desenvolvimento de um modelo.

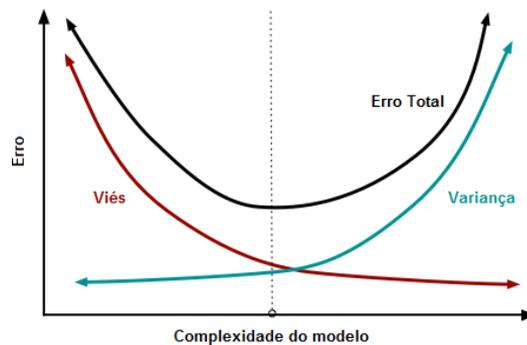
3.2.1 Compromisso entre viés e variância

Viés e variância são parcelas dos erros estatísticos de predição que permeiam todas as análises de *Machine Learning* (FRIEDMAN, 2017). Variância está vinculada aos erros inerentes a alta complexidade do modelo. Seu aumento com a aplicação do modelo a novos dados, indica que este está se ajustando aos ruídos aleatórios dos dados observados (*overfitting*). Viés, por sua vez, é o erro inerente a baixa complexidade do modelo utilizado. O seu aumento com a aplicação do modelo a dados novos, indica que este não é capaz de se adaptar as relações úteis à análise (*underfitting*). Estes erros são sempre somados e o valor resultante depende das características do modelo e dos dados. A minimização do erro total deve ser sempre buscada e, dentre outros fatores, está vinculada à complexidade escolhida ao modelo e a um compromisso: Quanto mais complexo, maior a variância, quanto menor a complexidade, maior o viés, como ilustrado na Fig. 6, .

3.2.2 Conjuntos de treino, validação e teste

O compromisso entre viés e variância não pode ser observado apenas com instâncias utilizadas para o treino: o aumento de complexidade tornará o modelo mais ajustado e específico ao conjunto e, com isso, menos generalizado. O aumento da variância será observado caso o modelo seja aplicado a um conjunto de validação, composto de instâncias não relacionadas ao treino. Portanto, para a escolha de uma complexidade, ou refinamento do modelo, deve-se avaliar a variação do erro de validação, o qual possui um valor mínimo indicando um balanceamento entre variância e viés, conforme observado na

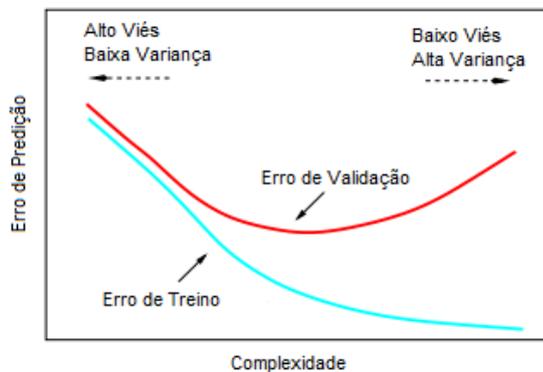
Figura 6 – Parcelas do erro de predição em função da complexidade do modelo.



Fonte: Adaptado de Bonfim (2020)

Fig. 7. Existe também um terceiro conjunto de dados de teste, instâncias que não foram utilizadas para o treino ou validação, avaliando como o modelo treinado com um conjunto de hiperparâmetros se comporta para a adição novas instâncias. Em muitos casos, dada a limitação de observações, o conjunto de validação usado é o mesmo que o de teste.

Figura 7 – Erros de validação e treino em função de sua complexidade.



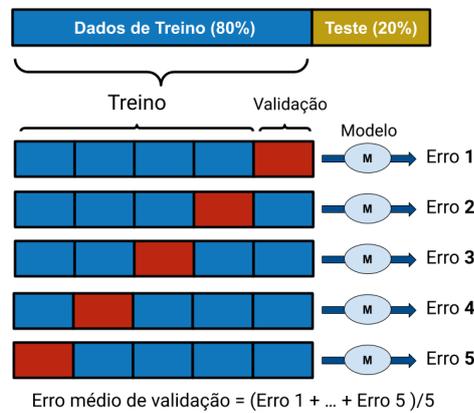
Fonte: Adaptado de Friedman (2017)

3.2.3 Validação Cruzada

A Validação Cruzada (*Cross-Validation*) consiste no particionamento sucessivo de todo o conjunto de dados de treinamento em dois conjuntos mutuamente exclusivos, respectivamente de treinamento e de validação (FRIEDMAN, 2017). Na Fig. 8 está representado um conjunto de partições em que sucessivamente 1/5 das instâncias de treinamento é separada para validação em cada uma delas. O erro médio final do processo será a média aritmética da soma dos resultados para cada partição. A vantagem desse procedimento é a diminuição do erro sistemático do conjunto de dados, ao passo que a desvantagem é o aumento do custo computacional. Deve ser notado que o conjunto de instâncias de teste,

neste caso 20% do total, está completamente segregado.

Figura 8 – Validação Cruzada com 20% das instâncias para teste, 20% do restante para validação.

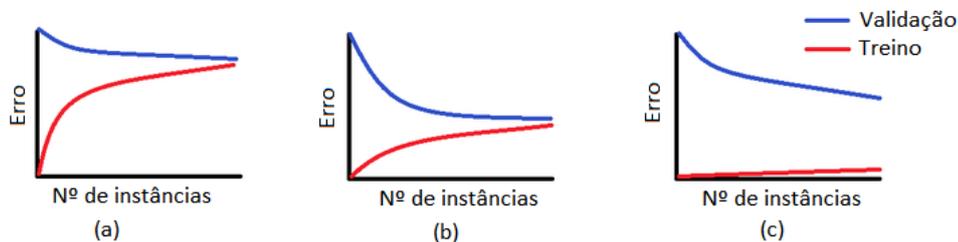


Fonte: Scaccia (2020)

3.2.4 Curva de Aprendizado

A Curva de Aprendizado (*Learning Curve*) é uma representação gráfica, cuja abscissa é o número de instâncias de treinamento e de validação, e cuja ordenada é uma medida do respectivo erro. A curva de aprendizado pode ser utilizada para se verificar se os resultados da aplicação do modelo apresentam mais viés ou variância, seja nos dados de treinamento ou de validação, podendo-se avaliar se o modelo se beneficiará da adição de mais instâncias à base de dados (GÉRON, 2019). Observe-se os três casos da curva de aprendizado na Fig. 9.

Figura 9 – Três condições usuais do comportamento da curva de aprendizado:
 (a) *underfitting*; (b) ideal; (c) *overfitting*.



Fonte: Adaptado de Amro (2011)

Nos três casos da Fig. 9, nota-se um aumento no erro de treino e redução do erro de validação com o aumento do número de instâncias. De fato, para uma complexidade fixada, à medida que cresce o número de instâncias de treinamento, mais difícil é para o modelo se adaptar, o que tende a aumentar o seu erro de treinamento. No entanto, melhora-se a sua capacidade de generalização, uma vez que ele tem mais informações

sobre a população subjacente e, conseqüentemente, maior capacidade de predição não enviesada, diminuindo, assim, o erro de validação. No caso (a), ambas as curvas de erro são assintoticamente elevadas, revelando muito viés (*underfitting*) com aquela complexidade. No caso (c), os erros de treinamento se mantêm muito baixos, ao passo que os erros de validação se mantêm assintoticamente altos, demonstrando muita variância (*overfitting*) com a dada complexidade. Já no caso (b), ambas as curvas assumem assintoticamente os mesmos valores baixos, para aquela complexidade, a situação ideal. Vale ressaltar-se que, se a métrica escolhida for um valor maior, como acurácia ou erro negativo, a discussão se altera completamente.

3.2.5 Dados Faltantes

Devido às dificuldades inerentes à coleta e registro dos dados, frequentemente um número significativo de instâncias não possui valores válidos para todos os seus atributos. De fato, dados faltantes são um problema corriqueiro na ciência de dados, dificultando a análise e reduzindo a eficácia de modelos de predição. Os dados faltantes podem, de acordo com a causa da falta, serem classificados nas seguintes categorias (BUUREN, 2018):

- MCAR (*Missing Completely At Random*): Quando as causas de falta não estão relacionadas com os dados, nem com o seu procedimento de coleta e registro. São eventos puramente aleatórios, uma situação idealizada.
- MAR (*Missing At Random*): Trata-se de uma categoria mais abrangente que o MCAR. Refere-se a situações em que as probabilidades dos dados estarem faltantes dependem de um atributo com valores conhecidos, mas não dos próprios valores ausentes. Por exemplo, sensores de velocidade são afetados por condições climáticas em um voo aeronáutico.
- MNAR (*Missing Not At Random*): Trata-se de quando as causas de terem dados faltantes se dá por seus próprios valores. Por exemplo, a tentativa de obter dados com uma balança com pesos acima do máximo que ela suporta.

Para lidar com os dados faltantes, um dos procedimentos mais comuns é removerem-se as observações com falta de dados, resultando, exclusivamente no caso MCAR, em uma coleção de dados não enviesados (BUUREN, 2018). Entretanto, para as demais categorias, esse procedimento poderá degradar a habilidade de detectar-se efeitos de interesse e, de fato, gerar um viés nos dados. Uma alternativa a esse método supressivo é a manutenção da base original dos dados, associada à utilização de algoritmos robustos diante de valores faltantes, representando as Árvores de Regressão um excelente exemplo desse tipo de algoritmo. Ainda uma terceira alternativa, a ser explorada no presente trabalho, é o preenchimento, ou a imputação, dos dados faltantes. Com a escolha dessa

alternativa, abrem-se inúmeras possibilidades, descritas em seguida, cada qual com suas vantagens e desvantagens.

3.2.6 Imputação de dados

Ao invés da simples supressão de dados, no processo de imputação, os dados faltantes são preenchidos utilizando critérios selecionados, de forma que o conjunto resultante dos dados adquira certas propriedades desejadas, ao mesmo tempo que rejeite aquelas indesejadas (BUUREN, 2018). Dos diversos critérios possíveis, destacam-se, com suas vantagens e desvantagens, os seguintes:

- Imputação por alguma média: Os dados faltantes são imputados pela média ou mediana dos dados não faltantes do atributo correspondente. A vantagem é a desnecessidade do conhecimento das relações daquele atributo com os demais, ao passo que a desvantagem é a subestimação da variância e o enviesamento da média (ou mediana) quando os dados faltantes não são MCAR.
- Imputação por modelos preditivos determinísticos: Este método consiste em inicialmente imputarem-se os dados faltantes pela média e, em seguida, aplicar-se um modelo para, utilizando-se a informação presente atualizada na base de dados, sucessivamente realizarem-se previsões, assumindo-se cada atributo com valores ausentes como alvo e os demais atributos como preditores. Trata-se de uma aplicação que garante estimativas de média e variância não enviesadas, além de correlações também não enviesadas no categoria MAR, quando a causa dos dados faltantes é um dos preditores considerados. Algumas desvantagens são o fortalecimento artificial das correlações entre as variáveis utilizadas, o que pode levar a modelos enviesados, além da ênfase na multicolinearidade, prejudicando a acuracidade dos modelos e dificultando a interpretabilidade dos resultados.
- Imputação por modelos preditivos estocásticos: A diferença entre este método e o anterior é a aplicação de um ruído aleatório às estimativas, com média nula e cuja variância é a mesma da distribuição dos resíduos provenientes da aplicação do modelo de previsão. Trata-se de uma forma de se contornar o problema do fortalecimento das correlações, garantindo-se um modelo não enviesado e evitando-se a multicolinearidade. A desvantagem é a possível imputação de dados que deixam de representar a distribuição original, podendo assumir valores irrealistas, como números negativos para atributos estritamente positivos.

A imputação de dados pode ainda ser realizada de forma estratificada, quando se levam em conta apenas as informações provenientes das instâncias de uma mesma classe para isso (SILVA, 2010). No presente trabalho, utilizar-se-á imputação estratificada.

3.2.7 Otimização de Hiperparâmetros

Nas seções anteriores, discutiu-se sobre a complexidade do modelo e seus efeitos no viés e na variância. A complexidade é, em geral, controlada pela seleção de um conjunto de hiperparâmetros. Estes consistem em parâmetros impostos ao modelo anteriormente ao seu treinamento, por exemplo, o estabelecimento a priori do grau do polinômio de uma regressão polinomial. A dificuldade da escolha de um conjunto “ótimo” de hiperparâmetros se prende à ausência de um (hiper)modelo a se otimizar, o que leva muitos projetos a recorrerem a uma busca exaustiva, demandando elevado custo computacional, ou randômica, não necessariamente garantindo um valor ótimo para a complexidade do modelo (ARCHETTI; CANDELIERI, 2019). Porém, outras alternativas vem sendo amplamente exploradas em busca de melhores resultados, dentre as quais se destaca, a ser empregado no presente trabalho, o algoritmo TPE (*Tree-Structured Parzen Estimator*) (BERGSTRA et al., 2011).

O algoritmo TPE é uma otimização bayesiana não paramétrica, com o qual se gera um modelo substituto, ou superfície de resposta, com base nos resultados da função objetivo, e a atualiza a medida que a função objetivo é avaliada para novos conjuntos de hiperparâmetros, se aproximando cada vez mais da distribuição multidimensional no espaço de hiperparâmetros em torno do erro mínimo (talvez) global (BERGSTRA et al., 2011). Como estimativa inicial pode ser utilizada uma distribuição pouco informada de hiperparâmetros, sendo as mais usuais, as distribuições Uniforme ou Log-Uniforme. A cada iteração, essa distribuição é atualizada até que sua alteração se torne pequena em algum sentido, ou número máximo de iterações for atingido.

3.3 Algoritmos de Regressão

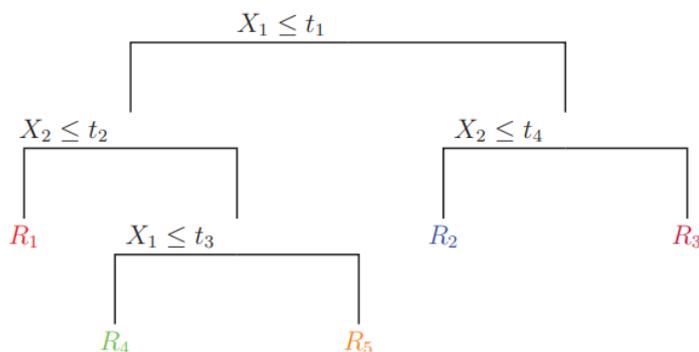
Para o presente trabalho, foi escolhido como algoritmo de regressão o *XGBoost* (CHEN; GUESTRIN, 2016), em razão de sua robustez e precisão, além da possibilidade da avaliação do impacto da utilização de diversos métodos de imputação para o aprimoramento das predições, uma vez que o algoritmo é robusto a dados faltantes. Mas antes de detalhar-se o funcionamento do *XGBoost* como um todo, deve-se discorrer sobre os algoritmos que o compõe, os quais são as Árvores de Regressão e o *Gradient Boosting* (FRIEDMAN, 2001).

3.3.1 Árvores de Regressão

Uma Árvore de Regressão é uma estrutura hierárquica de nós e ramos, na forma de uma árvore de cabeça para baixo. Seu princípio de crescimento é baseado em uma estratégia gulosa (ou egoísta), sendo localmente ótima em suas partições sob os nós. Assim, a partir do nó-raiz, conjuntos de testes lógicos são realizados sob determinados atributos (nós não terminais), particionando a árvore em ramos com novos nós. Este

processo é então repetido, até que um nó terminal (nó-folha ou variável dependente) com o valor da predição seja alcançado (MURPHY, 2012). Caso o nó-folha tenha mais de uma instância do alvo, o valor ajustado é o da média dos resultados. Na repartição na Fig. 10, X_i são os atributos, t_i são os valores de *threshold* e R_i são as regiões resultantes da partição. O atributo que será particionado é escolhido pela determinação do respectivo Ganho de Informação, ou seja, qual atributo, que, ao particionar a árvore, irá contribuir mais para a redução do erro total.

Figura 10 – Estrutura de uma árvore de decisão



Fonte: Murphy (2012)

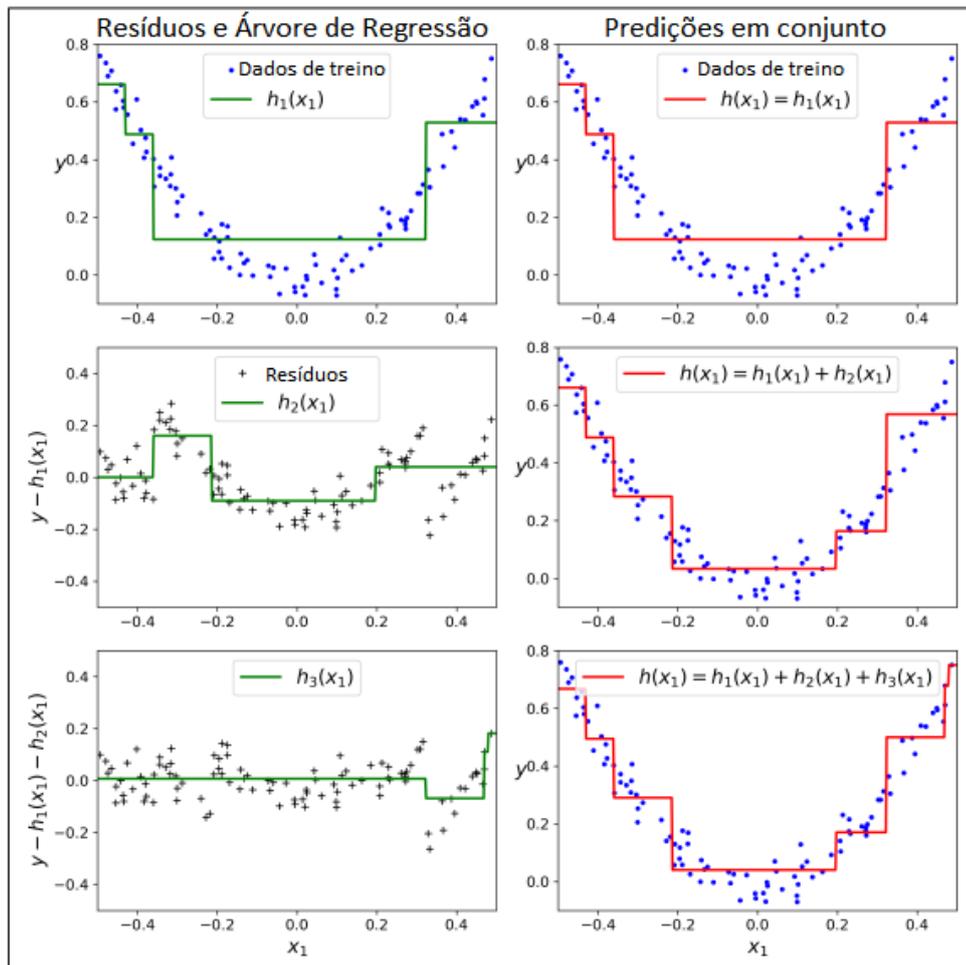
As vantagens da aplicação do algoritmo são sua fácil interpretabilidade, podendo-se verificar o fluxo de decisões dos valores, sua estrutura flexível e altamente controlável, seu funcionamento robusto a dados faltantes, sua capacidade de identificar relações não lineares e sua desnecessidade de pré-processamento dos dados. Entretanto, a Árvore de Regressão possui desvantagens que a tornam imprecisa em comparação a outros algoritmos. Pequenas alterações nos dados podem levar a árvores completamente diferentes, sendo assim propensa a sobreajuste dos dados. Por ser um algoritmo que funciona por partições com base em valores limiares, não consegue interpretar diferenças entre valores intermitentes às instâncias treinadas. Esse funcionamento também leva a um número limitado de predições, se equivalendo ao número de folhas da árvore. Por fim, possui uma quantidade considerável de hiperparâmetros, como quantidade mínima de instâncias em um nó para permitir uma partição, a profundidade máxima da árvore e a quantidade mínima de instâncias para um nó se tornar uma folha, o que dificulta a seleção do melhor conjunto de hiperparâmetros.

Embora a Árvore de Regressão seja um algoritmo de previsão considerado fraco (FRIEDMAN, 2017), sua estrutura simples permite que ela seja empregada em conjuntos de árvores (*ensembles*), aprimorando em muito o desempenho do modelo final. Existem diversos métodos com conjuntos de árvores. A técnica de conjunto utilizada para o presente trabalho é o *Gradient Boosting*.

3.3.2 Gradient Boosting

Gradient Boosting é um algoritmo *ensemble* inspirado no método de otimização do tipo gradiente descendente, o qual direciona a busca do mínimo da função de custo do problema para o maior incremento possível no vetor de atributos e o multiplica por uma taxa de aprendizado negativa. De modo análogo, o algoritmo *Gradient Boosting* melhora o seu desempenho com base nos termos residuais das árvores anteriores (FRIEDMAN, 2001).

Figura 11 – Funcionamento do *Gradient Boosting*



Fonte: Adaptado de Géron (2019)

A Fig. 11 ilustra o funcionamento simplificado do algoritmo para um único atributo x_1 e alvo y . Na primeira iteração do algoritmo treina-se uma Árvore de Regressão $h_1(x_1)$ ao conjunto de dados e é adicionado ao conjunto $h(x_1)$. Os resíduos, ou seja, a diferença entre os dados de treino e o conjunto $h(x_1)$, são usados para se treinar outra Árvore $h_2(x_1)$, novamente adicionada ao conjunto $h(x_1)$. O processo é feito sucessivamente até a adição de novas árvores não contribuírem mais para a redução do erro. A contribuição de cada árvore é controlada pela taxa de aprendizado, que no caso da figura é a unidade.

Além disso, todas as árvores têm profundidade máxima 2. No lado direito da figura, o resultado do conjunto pode ser analisado passo a passo.

Conforme Friedman (2001), é convencionalizado que, para problemas de regressão, a função de custo é obtida usando-se a metade da norma dos resíduos, os pseudo-resíduos, para facilitar o cálculo das derivadas. *Gradient Boosting* apresenta todas as vantagens das árvores de regressão, além de mitigar consideravelmente as desvantagens. Entretanto, *Gradient Boosting* levanta novas dificuldades, como o aumento computacional para o treino do modelo, a perda de interpretabilidade dos resultados e a elevada quantidade de hiperparâmetros.

3.3.3 *XGBoost*

XGBoost, ou *Extreme Gradient Boosting*, trata-se de um aprimoramento do conceito de *Gradient Boosting*, citando o desenvolvedor:

”[...] Tanto *XGBoost* quanto *Gradient Boosting* seguem o princípio do gradiente. No entanto, há diferença nos detalhes de modelagem. Especificamente, o *XGBoost* usou uma formalização de modelo mais regularizada para controlar o sobreajuste, o que lhe dá melhor desempenho. [...] *XGBoost* se refere ao objetivo da engenharia de empurrar o limite de recursos de computação para algoritmos de árvore otimizados.”(CHEN, 2015)

Dessa forma, o que difere do algoritmo tradicional *Gradient Boosting* é a introdução de um termo de regularização e a expansão do gradiente para a segunda ordem (hessiana), aumentando o controle de complexidade do modelo e sua convergência ao resultado ótimo. Além dessas alterações, adicionou-se a utilização de matrizes esparsas e estruturas de dados aprimoradas para melhor processamento dos dados no treino e predição de resultados, reduzindo drasticamente o custo computacional (CHEN; GUESTRIN, 2016). A única desvantagem desse algoritmo em relação ao *Gradient Boosting* é a presença de ainda mais hiperparâmetros a se alterar a fim de se obter um modelo ótimo.

3.3.4 Hiperparâmetros

XGboost possui diversos hiperparâmetros, porém, só serão tratados alguns mais importantes para o presente trabalho, os quais serão referenciados conforme a documentação do *XGboost*. O motivo dessa escolha é o objetivo dos outros hiperparâmetros, que estão mais direcionados ao controle computacional do treinamento, como por exemplo, a porcentagem dos dados utilizados para treino, ou a quantidade de atributos a serem utilizados nas partições das árvores. Como o número de instâncias no presente trabalho não é muito volumoso e serão no máximo quatro os atributos preditores utilizados, não foram considerados todos para a otimização. Será feita uma divisão dos hiperparâmetros

que controlam as árvores de regressão, os termos de regularização, e os que controlam o modelo de conjunto.

São os seguintes os hiperparâmetros que controlam as árvores de regressão:

- **max_depth**: Máxima profundidade das árvores. Controla o nível de subdivisões as árvores podem fazer. Quanto maior seu parâmetro, mais complexo torna seu modelo. Limite: $[1, \infty)$
- **min_child_weight**: Soma mínima da hessiana em um nó para realizar uma divisão. Caso esteja utilizando a função custo de pseudo-resíduos, a soma da hessiana acaba sendo o número de instâncias mínimas em um nó para realizar a divisão. Quanto maior o seu valor, mais conservativo é o modelo. Limite: $[0, \infty)$

Os hiperparâmetros vinculados à regularização são os seguintes:

- **lambda**: Termo de regularização L2, Aumentando-se seu valor, mais conservativo é o modelo. Limite: $[0, \infty)$
- **alpha**: Termo de regularização L1, Aumentando-se seu valor, mais conservativo é o modelo. Limite: $[0, \infty)$

Os hiperparâmetros que controlam o modelo de conjunto são:

- **eta**: Taxa de aprendizado, utilizada para controlar o ajuste do modelo. Altos valores garantem a um treinamento mais rápido, porém, pode sobreajustar o modelo. Limite: $(0, 1]$
- **n_estimators**: Número de iterações máximas permitidas para o ajuste do modelo. Quanto maior o número, se alcançado, mais complexidade é adicionada, podendo sobreajustar o modelo. Limite: $[1, \infty)$
- **early_stopping_rounds**: Controla a parada do aprendizado do algoritmo, antes de alcançado o **n_estimators**, após uma quantidade de iterações (*rounds*) que não contribuem para a redução do erro. Limite: $[1, \mathbf{n_estimators}]$

3.3.5 Importância de atributos

Para cada repartição de uma árvore, avalia-se qual dos atributos contribuirá mais com a redução do erro local. Este valor é armazenado para todas as repartições e todas as árvores do conjunto e, treinado o modelo, estes valores são somados por atributo e normalizado para que a somatória de todos seja a unidade. A importância é assim definida. Sua aplicação pode ser expandida para a importância dos hiperparâmetros em uma otimização, definindo o alvo como a métrica de erro.

4 Metodologia e Desenvolvimento do Trabalho

4.1 Procedimento Aplicado

Inicialmente, coletaram-se os dados primários provenientes de centenas de arquivos *csv* da base de dados da UIUC, que contém os dados referentes a 136 hélices distintas. Os arquivos, por sua vez, já estão, na base de dados, divididos em dois tipos. O que será denominado tipo 1, com 1037 arquivos, contém o ensaio de uma hélice, cada qual com Passo e Diâmetro definidos e a uma dada Rotação, com quatro colunas: Razão de Avanço, Coeficiente de Tração, Coeficiente de Potência e Eficiência. O que será chamado de tipo 2, com 79 arquivos, contém, para cada nome de Hélice e Fabricante, a distribuição da corda ao longo do raio.

A razão da discrepância entre o número de hélices distintas e o número de arquivos do tipo 2 é que, embora todos os ensaios tenham sido realizados no mesmo túnel de vento e seguindo as mesmas condições experimentais, apenas aos que fizeram parte do trabalho de Brandt e Selig (2011) foi registrada a distribuição da corda.

Com ferramentas de manipulação de arquivos de texto e de *strings*, extraíram-se os dados dos arquivos do tipo 1, estruturando-os em uma matriz com os seguintes atributos: Nome da Hélice, Fabricante, Passo por Diâmetro, Razão de Avanço, Rotação, Coeficiente de Tração, Coeficiente de Potência e Eficiência. Para o tipo 2, aplicou-se uma integração numérica, obtendo-se, então, a Solidez. Os resultados foram estruturados como outra matriz, com os atributos de Nome da Hélice e Solidez.

Utilizando-se o Nome da Hélice como atributo em comum entre ambas as matrizes, produziu-se uma mescla, resultando em uma nova matriz com 16455 instâncias, com todos os 8 atributos citados e com muitos dados faltantes, ressaltando-se, dentre eles, pela importância no presente trabalho, a Solidez. Todos os dados faltantes, 6936 instâncias (42% do conjunto total), podem ser classificados como MAR, pois dependem de fatores externos aos seus valores.

Após a estruturação dos dados, procedeu-se a uma análise exploratória, com o objetivo de se extraírem informações relevantes à modelagem. A análise consistiu da verificação de monotonicidade entre as variáveis, utilizando-se diagramas de dispersão, e também da obtenção dos coeficientes de correlação de Pearson entre os pares de variáveis. A avaliação desses coeficientes é imprescindível para o processo de imputação por regressão linear, uma vez que valores do módulo dos coeficientes próximos da unidade indicam maior relação de linearidade entre pares de variáveis. Em seguida, aplicaram-se três métodos distintos de imputação de dados faltantes à Solidez. Os métodos de imputação utilizados foram os seguintes: pela média, pela regressão linear determinística e estocástica. Entretanto, seja qual for o método, a imputação foi aplicada de maneira estratificada, ou seja, as classes, definidas por cada fabricante da hélice, segmentam o conjunto em subconjuntos, para que se possa efetuar o treinamento e a posterior imputação dos dados.

Para a imputação pela média, a aplicação do valor médio aos dados faltantes de Solidez de cada instância é direta, sem a necessidade de conhecimento das demais variáveis. Para a imputação linear determinística, utilizou-se a variável com maior correlação de Pearson com a Solidez. A mesma variável foi utilizada para a regressão linear estocástica. Para esta, utilizou-se a distribuição t-Student para a determinação do ruído aleatório que será adicionado à aos valores imputados garantindo a estocasticidade do processo. A razão de ter se escolhido esta distribuição se dá pelo conjunto reduzido de instâncias por classe, amostras de uma população, necessitando de uma distribuição que a represente corretamente. Os parâmetros para a modelagem da distribuição são média nula, variância igual a da distribuição dos resíduos provenientes da aplicação da regressão e o número de graus de liberdade sendo a quantidade de instâncias por classe menos um.

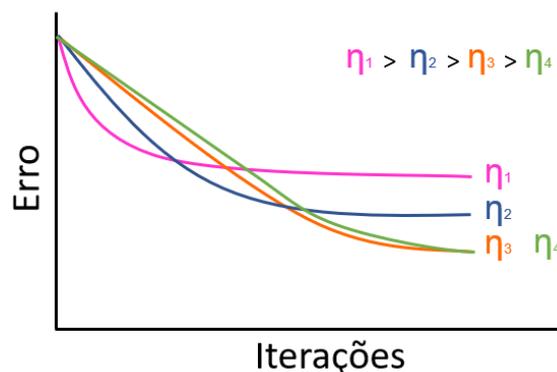
Após o processo de imputação, desenvolveu-se o treinamento dos diversos modelos pelo método de *XGBoost* do Coeficiente de Tração, cada qual com condições distintas: um modelo desconsiderando a Solidez, outro considerando a Solidez sem aplicação da imputação e finalmente os modelos considerando a Solidez, mas com a aplicação de cada um dos três métodos de imputação. Para o refinamento de hiperparâmetros, fixou-se *n_estimators* em 999, e *early_stopping_rounds* em 10, e aplicou-se o algoritmo de TPE para a otimização dos demais, cujas distribuições iniciais estão apresentadas na Tab. 1.

Tabela 1 – Distribuições iniciais dos hiperparâmetros para a aplicação do algoritmo TPE.

Hiperparâmetro	Intervalo	Distribuição
max_depth	10 a 30	Uniforme (inteiros)
min_child_weight	0 a 10	Uniforme (inteiros)
lambda	0 a 10	Uniforme (inteiros)
alpha	10^{-6} a 1	Log-Uniforme
eta	0,05 a 0,1	Uniforme

Fonte: Próprio autor.

Figura 12 – Influência da taxa de aprendizado no treinamento do modelo



Fonte: Próprio autor.

O número de iterações foi fixado em razão da forma como está relacionado a taxa de aprendizado. Observe-se a fig. 12, onde η_i é a taxa de aprendizado i , com a qual alguma medida de erro evolui com o número de iterações. Menores taxas de aprendizado provocam uma redução mais lenta do erro, aumentando, portanto, o número de iterações necessárias para chegar-se a um valor em que a adição de mais iterações não resultará em uma melhoria significativa. Note-se que existe, ainda, uma taxa de aprendizagem mínima, abaixo da qual não haverá redução apreciável do erro.

Das 16645 instâncias, o conjunto de teste foi obtido, selecionando-se aleatoriamente 20% delas de forma estratificada nas classes de Nome de Fabricante, para garantir uma boa representação de cada classe. Garantiu-se também que as instâncias fossem com Solidez não imputada, não havendo dados faltantes em seu conjunto. Com isso, restaram 80% das instâncias para compor o conjunto de treinamento (53% com dados faltantes ou imputados). Com o conjunto de teste apartado, o refinamento dos hiperparâmetros foi, então, conduzido por uma validação cruzada, com 3 separações, ao conjunto de treinamento.

Realizado o treinamento, avaliaram-se as curvas de aprendizado para a observância do compromisso entre viés e variância, bem como a necessidade ou não de novas instâncias. Então, aplicaram-se os modelos treinados ao conjunto de testes para se obter uma avaliação não enviesada da sua performance.

Para a análise do erro de predição, foi utilizado o parâmetro Raiz do Erro Quadrático Médio (REQM), ótimo para a avaliação do intervalo de predição. Devido a sua definição, REQM está naturalmente associado ao desvio padrão do espaço amostral dos resíduos s . Assumindo-se uma distribuição normal dos resíduos e possuir a mesma variância para todo o intervalo do alvo (homocedasticidade), com média nula, os limites superior e inferior do erro podem ser determinados, multiplicando-se REQM (ou s) pelo quantil z , conforme a eq. 8, onde \hat{y}_i é a predição $f(x_i)$ e y_i é o valor real. Na Tab. 2, são apresentados os valores de z calculados para cada intervalo de predição.

$$\hat{y}_i = y_i \pm z.s \tag{8}$$

Tabela 2 – Valores do quantil para cada intervalo de predição

Intervalo de predição (%)	z
67	1,00
90	1,64
95	1,96
99	2,58

Fonte: Próprio autor.

4.2 *Frameworks Utilizados*

O trabalho foi desenvolvido em sua totalidade na linguagem de programação *Python 3* (ROSSUM; DRAKE, 2009), no ambiente de desenvolvimento *Jupyter Notebook*. Com as bibliotecas *pandas* (MCKINNEY et al., 2010), *numpy* (HARRIS et al., 2020) e *scikit-learn* (BUITINCK et al., 2013), toda a manipulação e engenharia de atributos pode ser desenvolvida, bem como a imputação de dados faltantes. Para a análise exploratória dos dados, utilizaram-se as bibliotecas *seaborn* (WASKOM, 2021), *yellowbrick* (BENGFORT et al., 2018) e *missingno* (BILOGUR, 2018). O treinamento do modelo foi conduzido com a biblioteca própria do *XGBoost* se fazendo uso da API (*Application Programming Interface*) de *scikit-learn*. Por fim, para otimização de hiperparâmetros, utilizou-se o *framework* de *Optuna* (AKIBA et al., 2019).

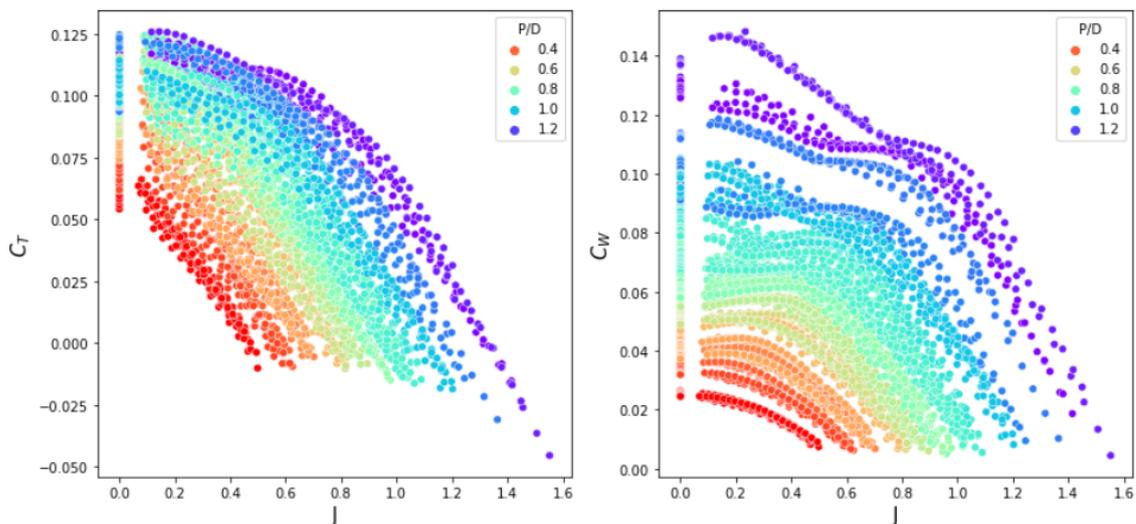
5 Resultados e Discussão

5.1 Análise Exploratória

Inicialmente, procedeu-se a uma análise exploratória dos dados, para se compreenderem as relações de dependência entre as variáveis Razão de Avanço J , Passo por Diâmetro P/D , Rotação N , Coeficiente de Tração C_T e de Potência C_W .

Como pode se observar na Fig. 13, em que estão representadas todas as instâncias da classe de hélices do fabricante APC Sport, a Razão de Avanço J desempenha um papel fundamental sobre os Coeficientes de Tração C_T e de Potência C_W , tal que, com o aumento de J , os valores dos coeficientes decaem rapidamente. As curvas são parametrizadas de forma contínua pelo Passo por Diâmetro P/D , sendo a legenda um guia da gradação de cores com a variação contínua dos valores. Observa-se um aumento dos Coeficientes com o aumento de P/D , apesar de as curvas se sobreporem para valores maiores de P/D .

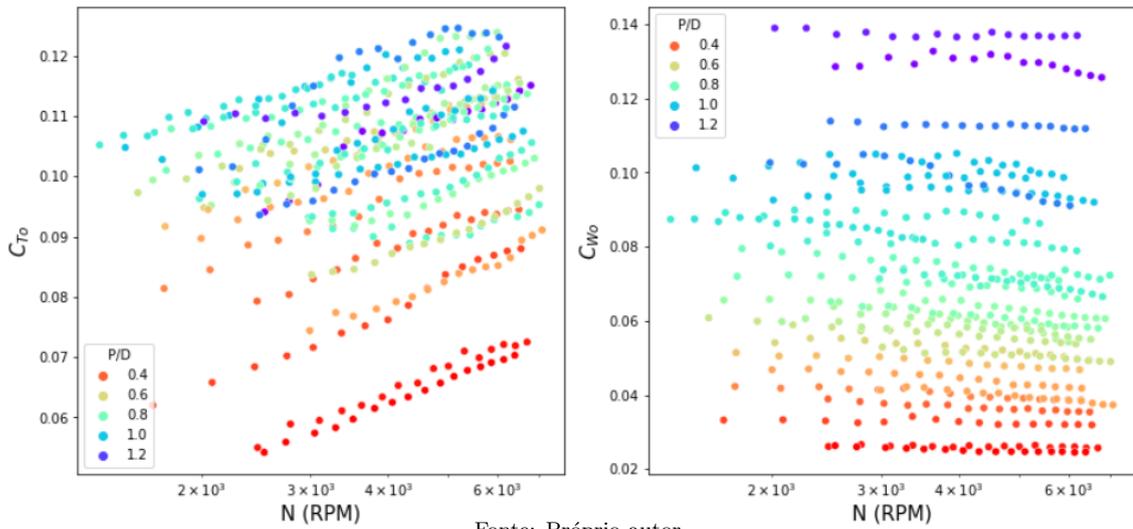
Figura 13 – Coeficientes de Tração C_T e de Potência C_W , como função da Razão de Avanço J , parametrizados de forma contínua pelo Passo por Diâmetro P/D , para as instâncias da classe APC Sport.



Fonte: Próprio autor.

Analisou-se o comportamento das hélices com a Rotação N . Não foi possível obterem-se informações com o gráfico de dispersão, uma vez que a Razão de Avanço J exerce uma forte influência sobre os coeficientes. Em razão disso, produziu-se um gráfico de dispersão considerando-se apenas a condição estática, ou seja, Razão de Avanço J nula. Pode-se observar na Fig. 14, que o Coeficiente de Tração estático C_{P_0} aumenta com a Rotação N , porém, esta não causa grandes alterações no Coeficiente de Potência estático C_{W_0} . Brandt e Selig (2011) apontam o aumento do número de Reynolds, como uma influência significativa nos coeficientes C_T e C_W , o que, infelizmente, não pudemos confirmar com os dados disponíveis, pois as condições climáticas de pressão, temperatura e umidade dos ensaios não foram informadas.

Figura 14 – Coeficientes estáticos C_{T_o} e C_{W_o} por Rotação N parametrizados pelo Passo por Diâmetro P/D , para hélices da família APC Sport.



Fonte: Próprio autor.

3. Avaliou-se a correlação de Pearson entre as variáveis, como pode ser visto na Tab.

Tabela 3 – Correlação de Pearson entre as variáveis

Coefficientes	D	P	P/D	Solidez	J	N
Tração	0,02	0,18	0,26	0,07	-0,80	-0,03
Potência	-0,02	0,37	0,51	0,09	-0,50	-0,05

Fonte: Próprio autor.

A adimensionalização do Passo permite que haja uma correlação mais forte da variável para com os coeficientes. O Passo por Diâmetro apresenta-se também como mais linearmente correlacionado que a Razão de Avanço para o Coeficiente de Potência. Sobre a Solidez, ela não aparenta ter uma forte correlação linear com os coeficientes, o que não significa que o parâmetro não seja menos importante para os coeficientes, uma vez sua relação pode ser não linear. Conforme feito para a análise da Rotação, analisaremos os coeficientes estáticos, apresentado na Tab. 4.

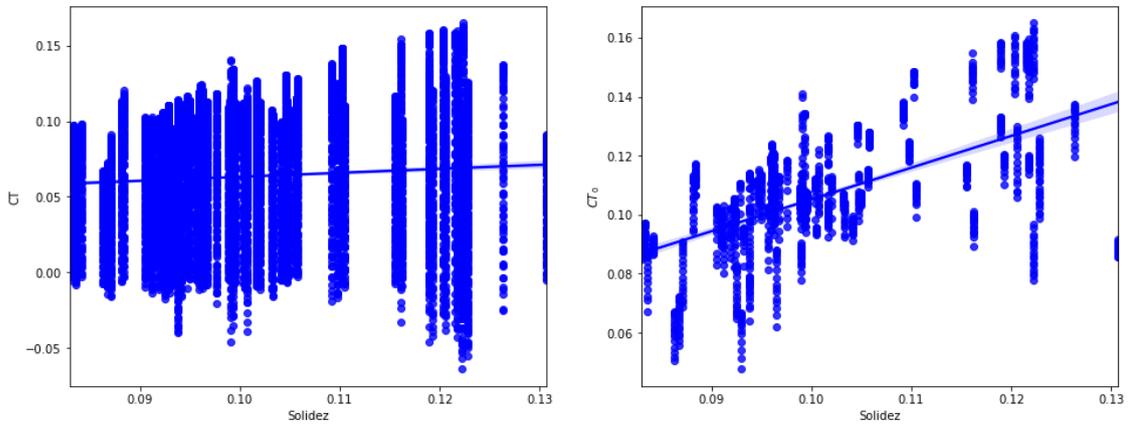
Tabela 4 – Correlação de Pearson entre as variáveis estáticas

Coefficientes Estáticos	D	P	P/D	Solidez	N
Tração	-0,17	0,24	0,47	0,60	0,17
Potência	-0,02	0,56	0,78	0,27	-0,12

Fonte: Próprio autor.

A Solidez apresenta uma correlação linear forte com o Coeficiente de Tração estático, podendo este ser usado para a imputação dos dados por regressão linear, de acordo com a Fig. 15.

Figura 15 – Regressões lineares para os Coeficientes de Tração dinâmico C_T e estático C_{T_0} , como função da Solidez.

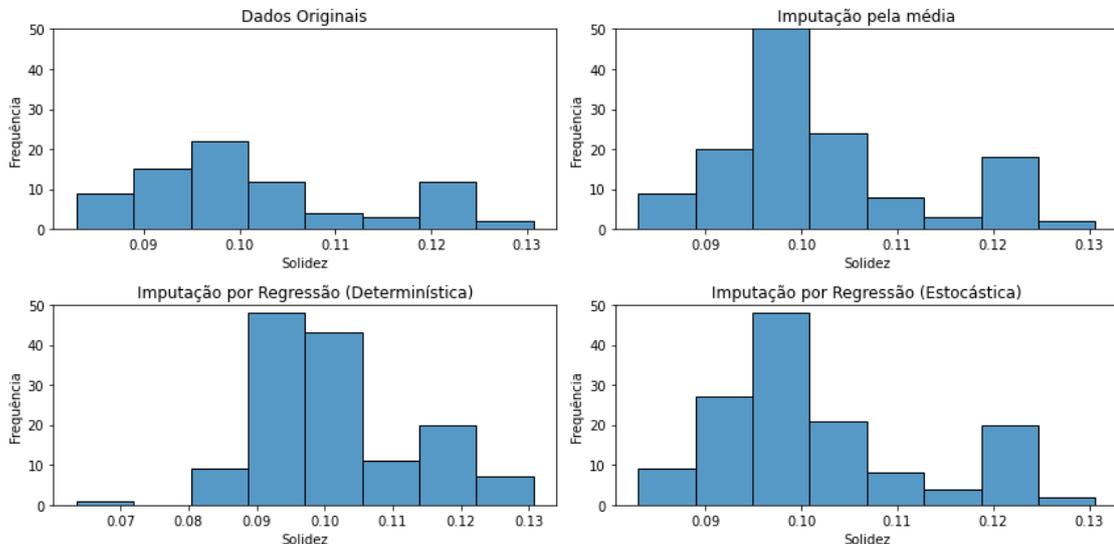


Fonte: Próprio autor.

5.2 Imputação de dados

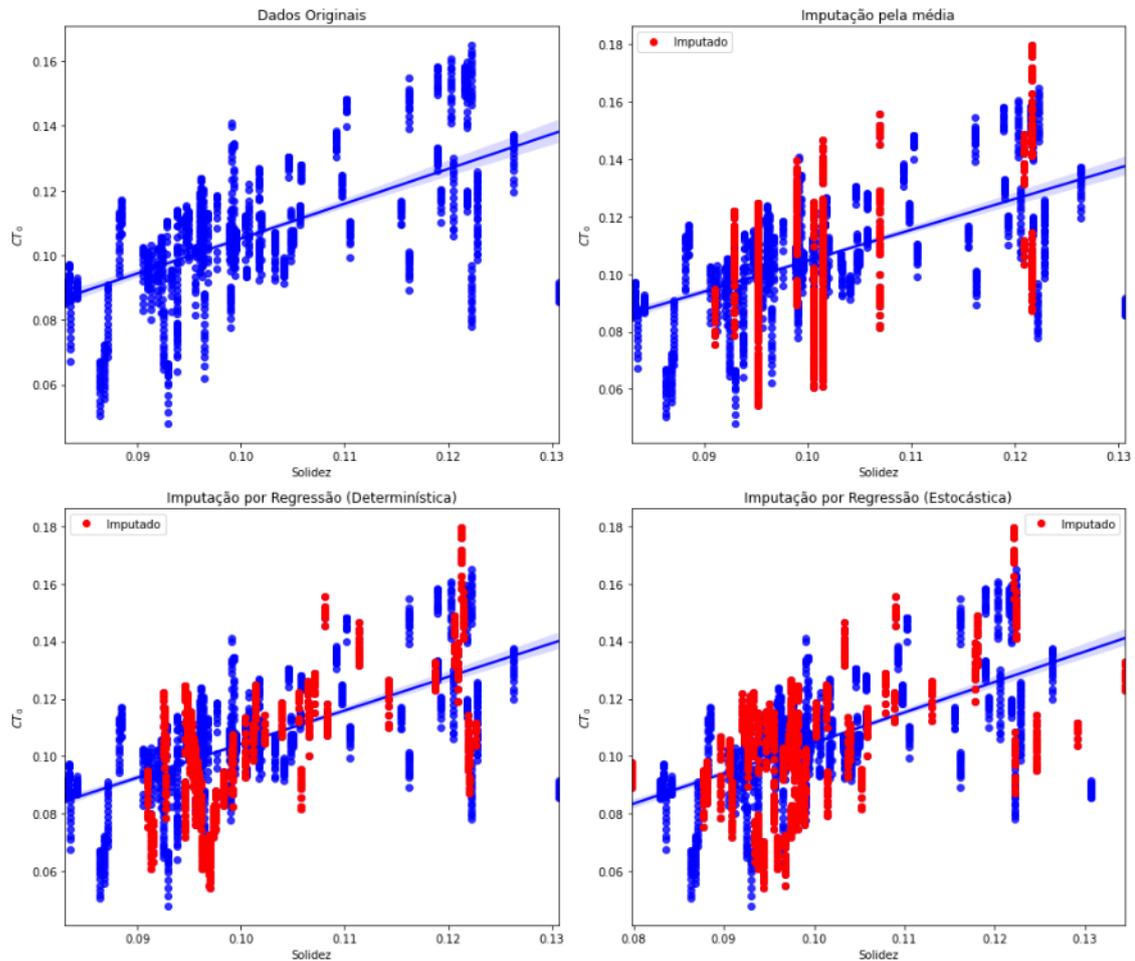
Os histogramas e gráfico de dispersão dos métodos de imputação podem ser vistos respectivamente na Fig. 16 e 17. Observa-se que o método de imputação por regressão estocástica apresenta uma distribuição mais semelhante aos dados originais, bem como ser mais variada no gráfico de dispersão. Fez-se também uma análise quantitativa com base nos parâmetros da distribuição apresentados na Tab. 5. O método aplicando a média distorce o Desvio Padrão e Correlação mais agressivamente que os outros métodos. A Determinística é o método que menos varia a média e a Correlação, apesar que fortemente distorcer o Desvio Padrão, alterando o histograma da distribuição da Solidez. Por essas razões e por ser conceitualmente mais adequado, o método de imputação estocástica foi escolhido entre os demais para a imputação dos dados.

Figura 16 – Histogramas dos dados originais e após a aplicação da imputação.



Fonte: Próprio autor.

Figura 17 – Dispersão dos dados originais e após a aplicação da imputação.



Fonte: Próprio autor.

Tabela 5 – Parâmetros das distribuições original e após a aplicação da imputação.

Método de Imputação	Média	Desvio padrão	Correlação (Tração Estática)
Sem imputação	0,1019	0,0117	0,60
Média	0,1014	0,0102	0,52
Determinística	0,1017	0,0107	0,59
Estocástica	0,1015	0,0113	0,57

Fonte: Próprio autor.

5.3 Modelagem: Três Parâmetros

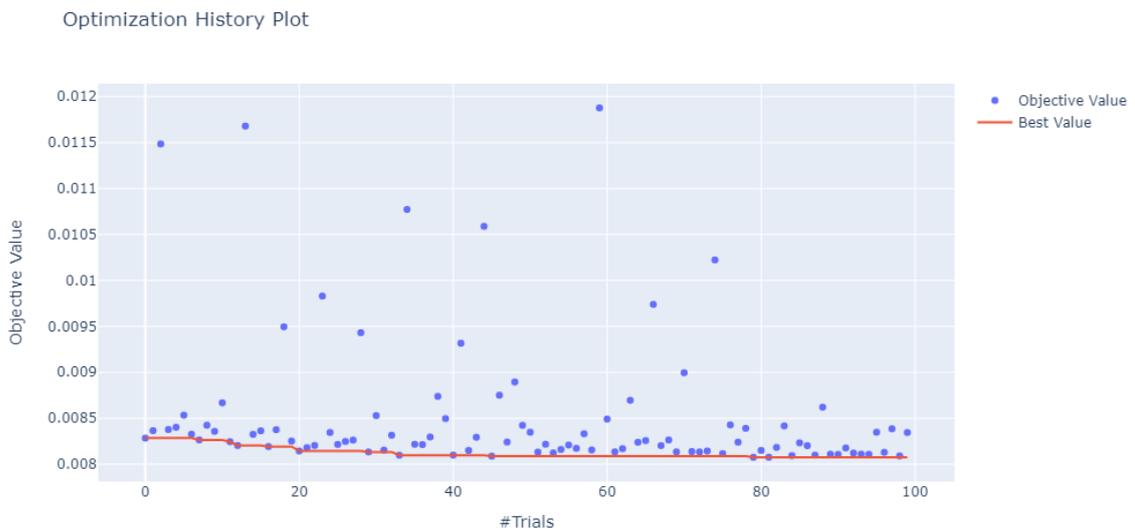
A primeira modelagem considerada foi a modelagem assumindo três parâmetros: Razão de Avanço, passo por diâmetro, rotação. A razão de ter feito essa escolha é que a hélice não costuma ter sua geometria anunciada pelo fabricante. Este modelo pode ser utilizado para a avaliação prévia de hélices comerciais a fim de reduzir a quantidade de hélices a se testar em um projeto.

O decréscimo da REQM de validação pode ser observado no histórico de otimização,

na Fig. 18. O valor do erro de validação convergiu para 0,00808 e erro de treino para 0,00461. Após 79 iterações, não houve decréscimo do erro na otimização.

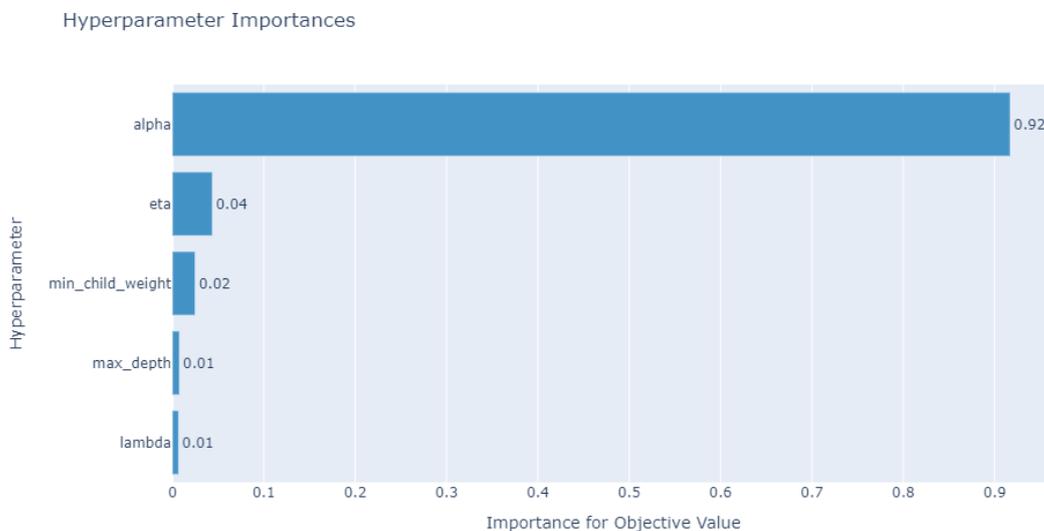
Na Fig. 19, temos a importância dos hiperparâmetros para a alteração do erro dentro dos limites estipulados. A regularização L1 (alpha) mostrou-se o hiperparâmetro mais influente no processo de otimização, uma vez que é uma penalização mais agressiva do algoritmo, seguido da taxa de aprendizado que controla a complexidade do modelo.

Figura 18 – Histórico de otimização dos hiperparâmetros (Três Parâmetros).



Fonte: Próprio autor.

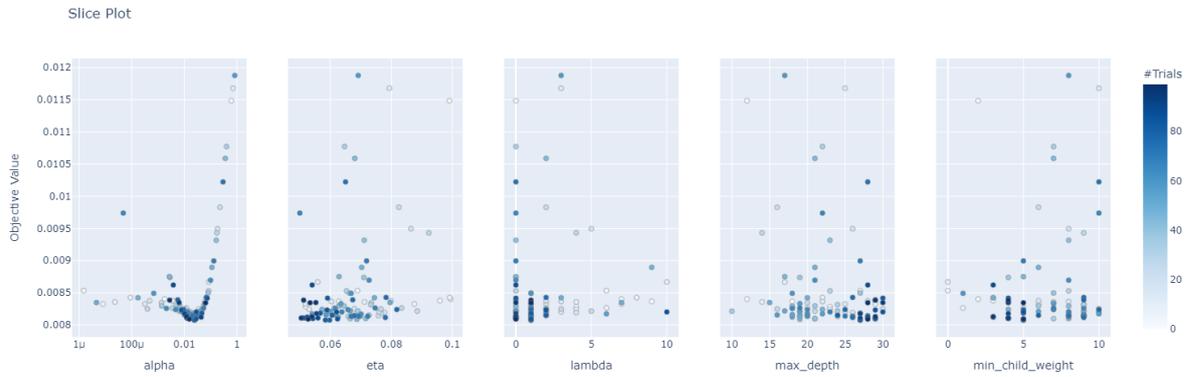
Figura 19 – Importância dos hiperparâmetros para a otimização (Três Parâmetros).



Fonte: Próprio autor.

Observa-se na Fig. 20, uma convergência significativa para a regularização L1 a um resultado ótimo, e uma convergência da tava de aprendizado para valores menores.

Figura 20 – Convergência dos hiperparâmetros por iteração (Três Parâmetros).



Fonte: Próprio autor.

Tabela 6 – Hiperparâmetros Finais (Três Parâmetros).

Hiperparâmetro	Valores
max_depth	27
min_child_weight	5
lambda	1
alpha	0,024
eta	0,059
n_estimators	269

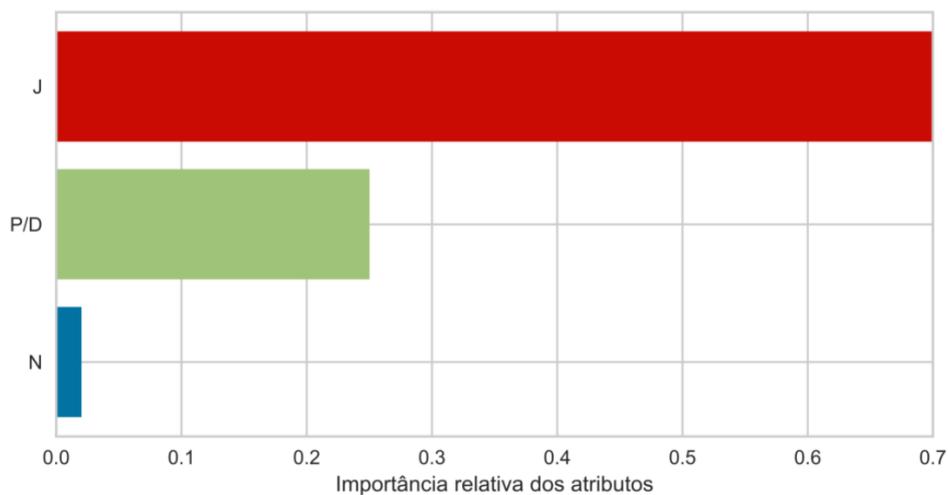
Fonte: Próprio autor.

Desenvolvido o modelo, analisou-se a importância das variáveis. Conforme a Fig. 21, a Razão de Avanço apresenta a maior importância dentre as variáveis, seguido pelo Passo por Diâmetro e Rotação, coincidindo com a correlação de Pearson feita anteriormente.

Na Fig. 22, tomando como referência o REQM negativo, métrica padronizada do *scikit-learn*, observamos um aumento do erro de treino e redução do erro de validação a medida que mais instâncias são adicionadas ao treinamento, sendo assim um indicativo que novas observações podem garantir uma redução da parcela de variância do erro, mas não da de viés.

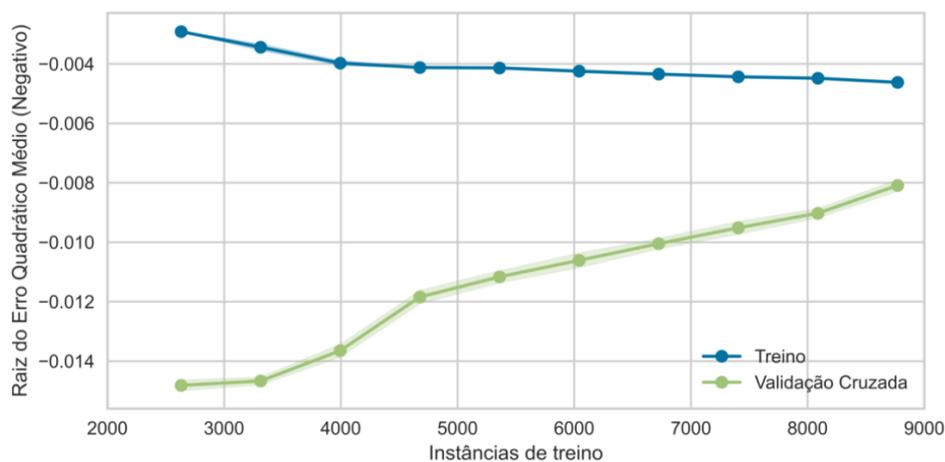
Na Fig. 23, vê-se uma distribuição residual normal com média próxima de zero, com maiores erros absolutos para maiores valores, referentes a valores de baixa Razão de Avanço, apresentando assim heterocedasticidade, o que torna o intervalo de predição menos eficiente. O modelo será mais preciso para maiores velocidades de voo e menores rotações. O erro de treino é 0,00751, menor que o de validação.

Figura 21 – Importância das variáveis independentes (Três Parâmetros).



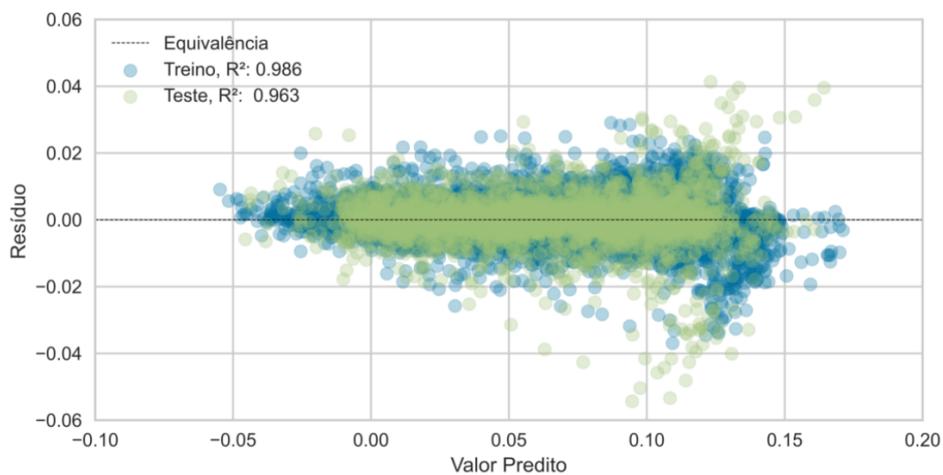
Fonte: Próprio autor.

Figura 22 – Curva de aprendizado do modelo (Três Parâmetros).



Fonte: Próprio autor.

Figura 23 – Resíduo das amostras de treino e teste (Três Parâmetros).



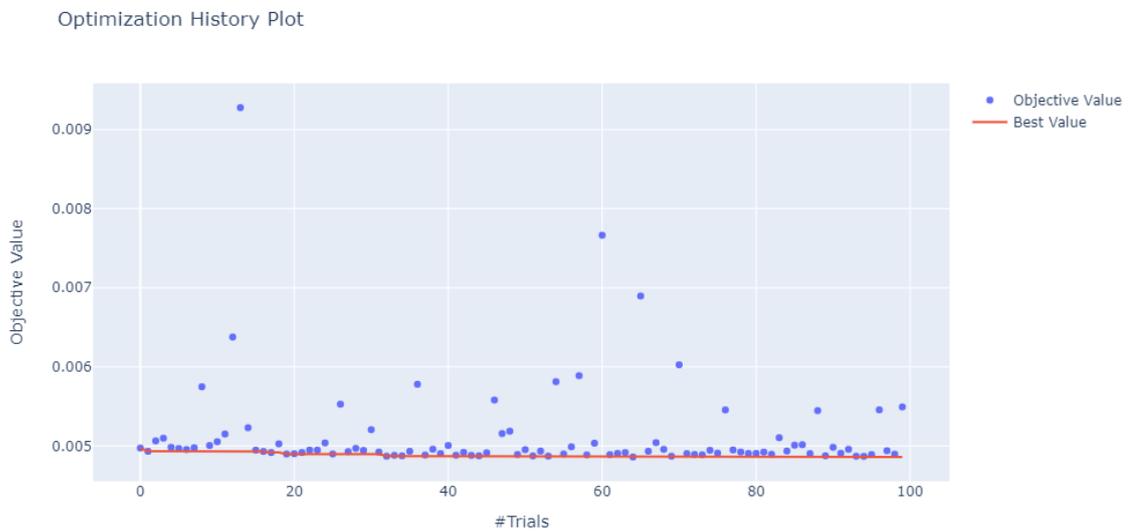
Fonte: Próprio autor.

5.4 Modelagem: Quatro Parâmetros Sem Imputação

A segunda modelagem leva em consideração a Solidez, que pode ser utilizada para projetos iniciais de hélices. No entanto, não leva em consideração a imputação aplicada, dessa forma, podemos avaliar o impacto desse procedimento no modelo.

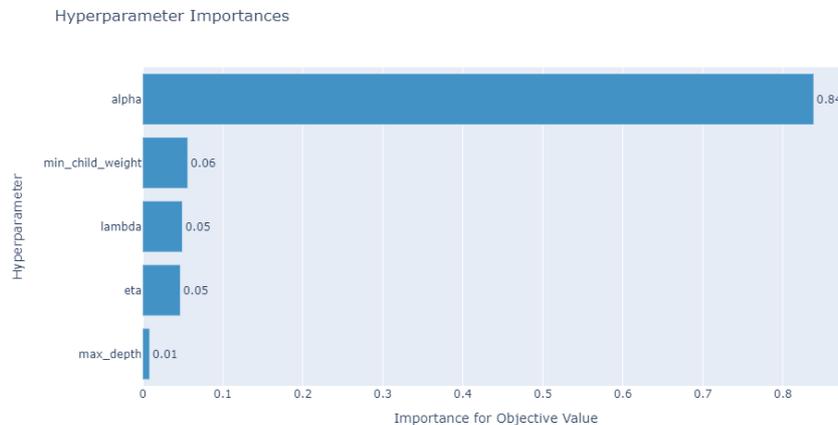
Na Fig. 24, observa-se quase nenhuma variação significativa do erro ao longo das 100 iterações, tendo como REQM de validação de 0,00487 a 32 iterações e 0,00486 a 64 iterações. O erro de treino final é de 0,00308.

Figura 24 – Histórico de otimização dos hiperparâmetros (Quatro Parâmetros Sem Imputação).



Fonte: Próprio autor.

Figura 25 – Importância dos hiperparâmetros para a otimização (Quatro Parâmetros Sem Imputação).

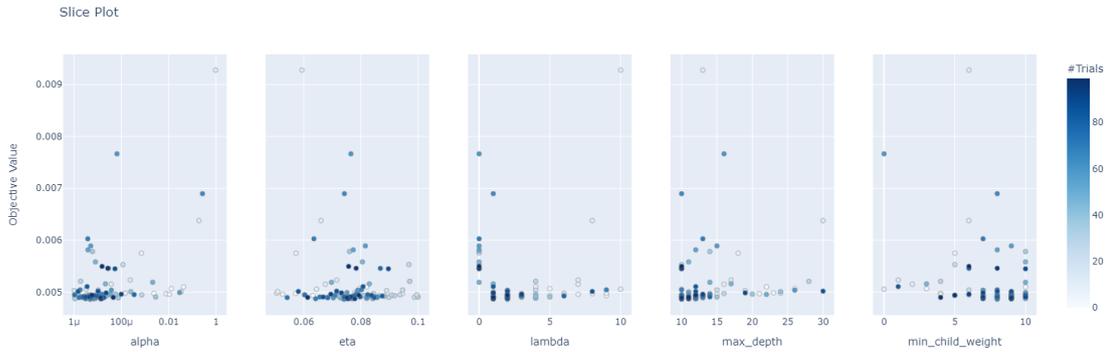


Fonte: Próprio autor.

A regularização L1 continua predominante no processo de otimização, conforme a

Fig. 25, seguido pela quantidade mínima de folhas em um nó terminal. Não se observa uma convergência a um valor ótimo como o modelo anterior, mas observa-se um aumento significativo a medida que o parâmetro se aproxima de 1, conforme visto na Fig. 26.

Figura 26 – Convergência dos hiperparâmetros por iteração (Quatro Parâmetros Sem Imputação).



Fonte: Próprio autor.

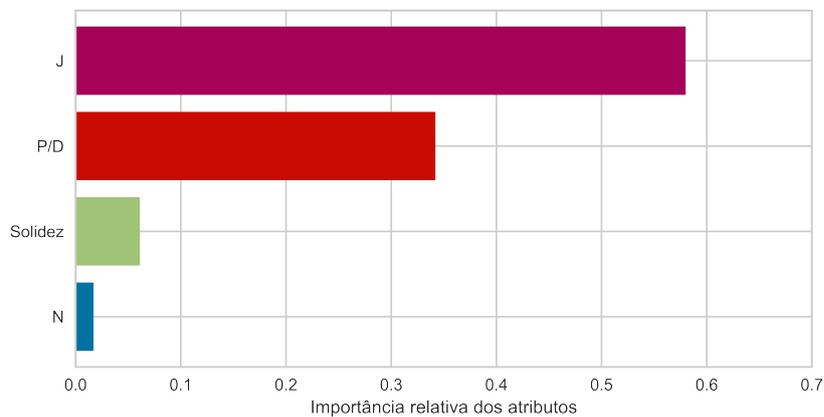
Tabela 7 – Hiperparâmetros Finais (Quatro Parâmetros Sem Imputação).

Hiperparâmetro	Valores
max_depth	10
min_child_weight	8
lambda	2
alpha	$4,8 \cdot 10^{-6}$
eta	0,074
n_estimators	510

Fonte: Próprio autor.

Conforme a Fig. 27, verifica-se que a Solidez desempenha um papel mais importante na redução do erro que a Rotação.

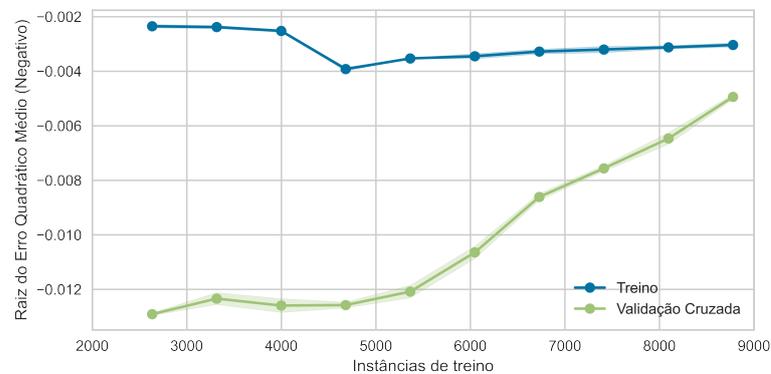
Figura 27 – Importância das variáveis independentes (Quatro Parâmetros Sem Imputação).



Fonte: Próprio autor.

Na Fig. 28, observa-se uma melhoria significativa que o modelo anterior, reduzindo seu erro mais que a metade do erro de validação e treino. A parcela de variância é baixa, a adição de novas instâncias pode não beneficiar tanto o modelo. Outro ponto importante para a análise é a redução do erro de treino a medida que o número de instâncias aumentam após 4800 instâncias de treino. Não foi controlada a proporção de dados faltantes a cada separação do conjunto de treino, portanto, essa redução pode estar relacionada a adição de instâncias ao treino que sem dados faltantes, instâncias essas mais informativas que instâncias com dados faltantes.

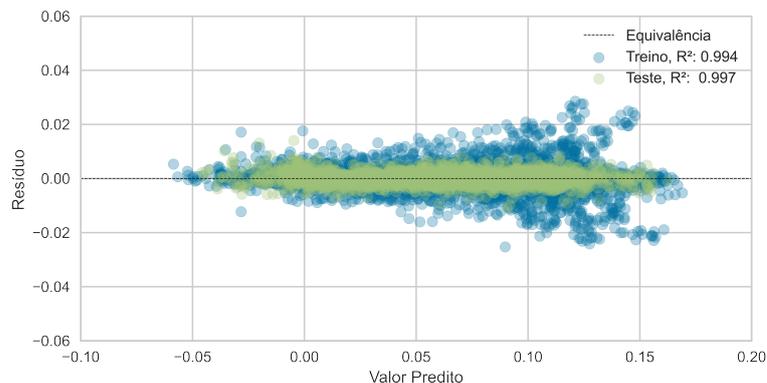
Figura 28 – Curva de aprendizado do modelo (Quatro Parâmetros Sem Imputação).



Fonte: Próprio autor.

Na Fig. 29, temos um fenômeno particular, o erro de teste ser menor que o treino, com valor de 0,00210. Ressalta-se que o conjunto de teste não possui dados faltantes, em outras palavras, todas as instâncias tem informações relevantes para a predição do Coeficiente de Tração que o Conjunto de Treino, que possui em sua composição maioria de instâncias com dados faltantes. A heterocedasticidade se mantém para os dados de treino, apesar reduzida.

Figura 29 – Resíduo das amostras de treino e teste (Quatro Parâmetros Sem Imputação).

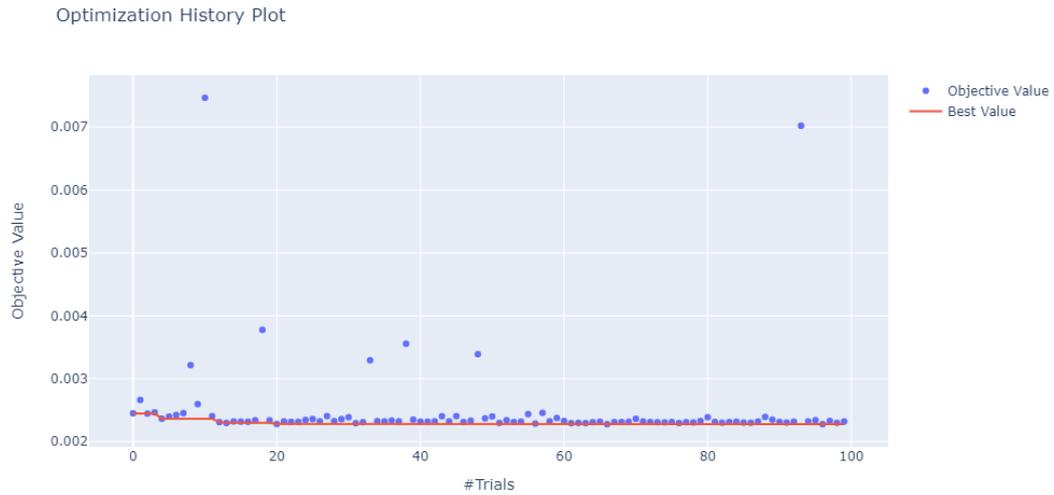


Fonte: Próprio autor.

5.5 Modelagem: Quatro Parâmetros Com Imputação

A terceira modelagem aplica o processo de imputação descrito anteriormente. Esta última tem como objetivo o desenvolvimento de um modelo substituto para um projeto preliminar de uma hélice.

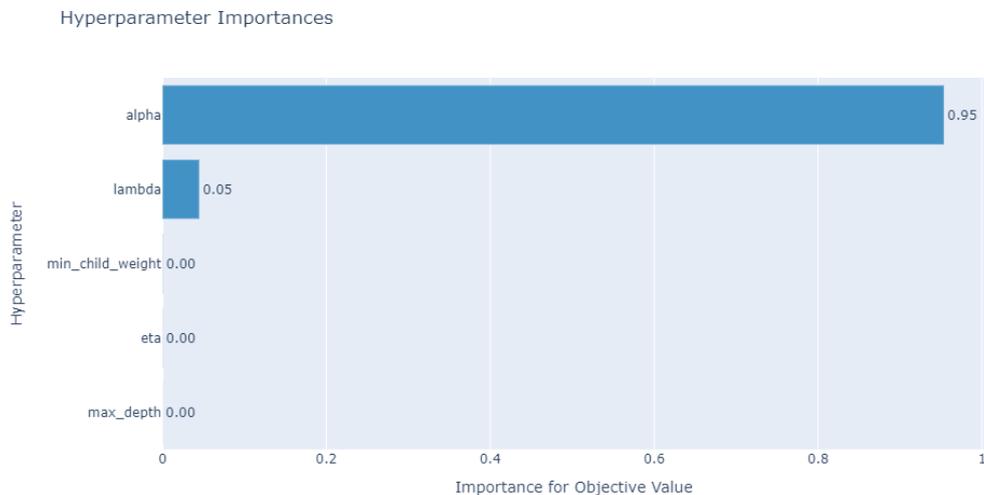
Figura 30 – Histórico de otimização dos hiperparâmetros (Quatro Parâmetros Com Imputação).



Fonte: Próprio autor.

Seguindo o exemplo dos modelos anteriores, o erro não reduziu significativamente, conforme visto na Fig. 30, com erro de validação 0,00228 após 20 iterações e 0,00227 após 66. O erro de teste final é 0,00078.

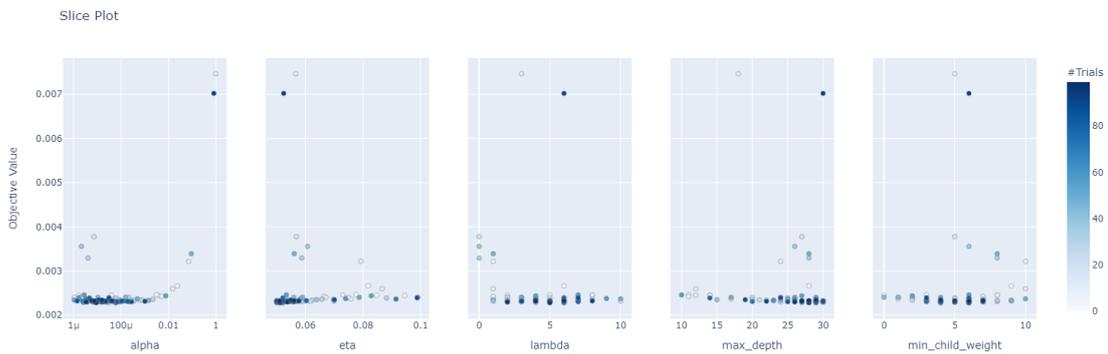
Figura 31 – Importância dos hiperparâmetros para a otimização (Quatro Parâmetros Com Imputação).



Fonte: Próprio autor.

A regularização L1 continua predominando como o mais influente dos hiperparâmetros, seguido da regularização L2. Ressalta-se que esses resultados não indicam que alguns são desnecessários, mas que, dentro da faixa em que foram limitados e das distribuições iniciais escolhidas, não contribuem significativamente para a redução do erro. Conforme observado na Fig. 32, O erro aumenta consideravelmente com o aumento da regularização L1 após um certo valor.

Figura 32 – Convergência dos hiperparâmetros por iteração (Quatro Parâmetros Com Imputação).



Fonte: Próprio autor.

Tabela 8 – Hiperparâmetros Finais (Quatro Parâmetros Com Imputação).

Hiperparâmetro	Valores
max_depth	29
min_child_weight	4
lambda	6
alpha	$5,97 \cdot 10^{-5}$
eta	0,053
n_estimators	295

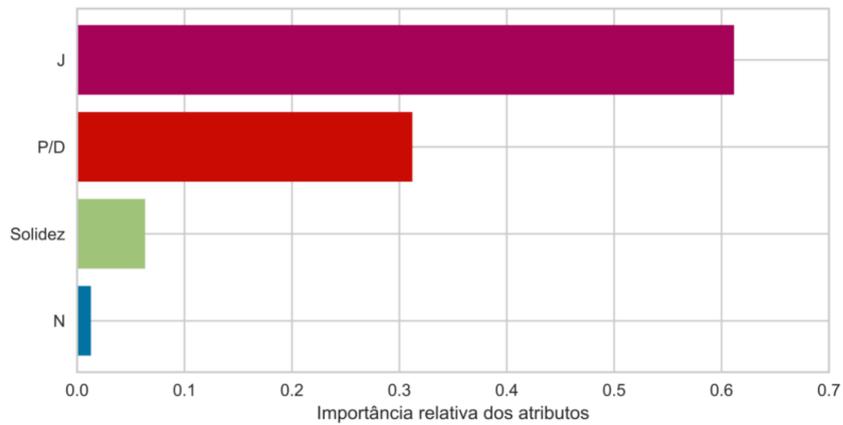
Fonte: Próprio autor.

Conforme a Fig. 33, A ordem de importância dos hiperparâmetros se mantém. verifica-se, no entanto, um crescimento da importância do hiperparâmetro de Razão de Avanço em relação ao modelo anterior.

É de se notar que, conforme a Fig. 34, o processo de imputação reduz consideravelmente o erro, tanto em variância como em viés considerando os conjuntos de treino e validação. Quase não se observa variação alguma no erro de treino ao se adicionar mais instâncias ao seu treinamento. As curvas quase se encontram se utilizando a totalidade dos dados, indicando baixa variância no modelo.

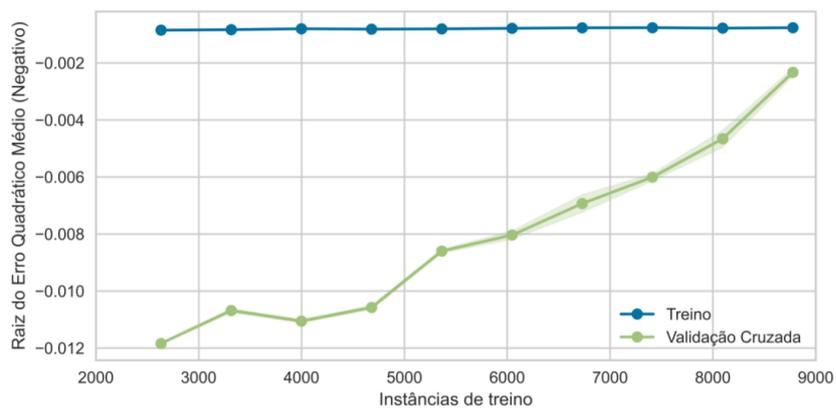
Pela Fig. 35, observa-se uma distribuição muito mais uniforme, sem dispersões consideráveis ao longo dos valores do Coeficiente de Tração, garantindo assim a homocedasticidade. O valor do erro de teste é 0,00199.

Figura 33 – Importância das variáveis independentes (Quatro Parâmetros Com Imputação).



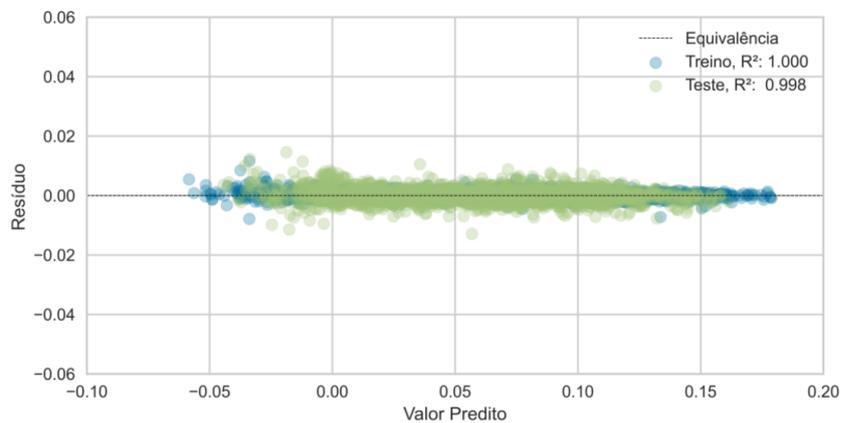
Fonte: Próprio autor.

Figura 34 – Curva de aprendizado do modelo (Quatro Parâmetros Com Imputação).



Fonte: Próprio autor.

Figura 35 – Resíduo das amostras de treino e teste (Quatro Parâmetros Com Imputação).



Fonte: Próprio autor.

Como forma de sumarizar, as REQM médios obtidas pelas três aproximações consideradas estão dispostas na Tab. 9, separados pelos conjuntos de treino, validação e teste e o Intervalo de Predição, assumindo uma confiabilidade de 95% aplicado ao erro de validação.

Tabela 9 – REQM obtidas pelas três aproximações consideradas

Modelagem	Treino	Validação	Teste	Intervalo (95%)
Três Parâmetros	0,00461	0,00808	0,00751	0,01589
Quatro Parâmetros Sem Imputação	0,00308	0,00486	0,00210	0,00952
Quatro Parâmetros Com Imputação	0,00078	0,00227	0,00199	0,00445

Tem-se resultados satisfatórios com as três aproximações. Para uma variação do coeficiente de tração de 0,00 a 0,18, a margem é relativamente pequena. Para todas as observações o erro de validação é superior que o erro de teste. A razão pode ser o tamanho do conjunto de teste em comparação com o de validação, menos instâncias implicam em menos variabilidade do conjunto, garantindo erro menor. No entanto, o conjunto de teste não foi escolhido de forma completamente aleatória, uma das condições era que o conjunto de teste fosse composto apenas de instâncias sem dados faltantes ou imputados. Dessa forma, o modelo Quatro Parâmetros Sem Imputação obteve um erro de teste consideravelmente menor que o seu erro de validação e treino, ambos que lidavam com dados faltantes.

6 Conclusão e Trabalhos Futuros

Ao fim deste trabalho, foi possível alcançar os objetivos estabelecidos no início, a criação de modelos substitutos confiáveis para a predição do coeficiente de tração com base na Razão de Avanço, Passo por Diâmetro, Rotação e Solidez. Desenvolveu-se duas aproximações que satisfazem duas condições que um projetista pode se encontrar, a necessidade da compra da hélice ou o desenvolvimento de um projeto preliminar de um sistema motopropulsivo. Referente a influência dos parâmetros, a Razão de Avanço mostrou-se predominante, seguido do passo por diâmetro. A rotação possui uma contribuição menor e, infelizmente, não foram disponibilizadas informações referentes às condições climáticas para a análise da influência do número de Reynolds, diretamente relacionado a Rotação. A adição do parâmetro de Solidez aumentou consideravelmente a precisão do modelo, reduzindo em 59% a raiz do erro quadrático médio sem imputação, e 26% considerando a imputação, para o conjunto de dados de validação. Em relação ao algoritmo *XGBoost*, verificou-se que o desempenho do modelo melhora com a aplicação do método de imputação por modelo para os conjuntos de treino e validação. Neste trabalho, utilizou-se um conjunto de teste sem dados imputados ou faltantes, o que, conforme observado pelos resultados, a imputação não melhora consideravelmente o modelo. No processo de otimização, o hiperparâmetro de regularização L1 foi predominante na redução do erro, nota-se portanto que regularização no *Gradient Boosting* impacta em sua performance.

Para trabalhos futuros, o autor propõe:

- Utilização de conjunto de teste com dados faltantes, a fim de se verificar o impacto da imputação para essas condições.
- Desenvolvimento de modelos preditivos ao Coeficiente de Potência.
- Adição de novas instâncias para o aprimoramento do modelo de três parâmetros.
- Desenvolvimento de uma API ao modelo para sua ampla utilização e divulgação.
- Utilização de outros métodos de imputação, bem como diferentes critérios de decisão para a escolha do método.
- Aplicação de outros algoritmos de regressão, como Máquinas de Vetor Suporte, Redes Neurais, Regressão Polinomial, entre outros.

Referências

- Adonai Topografia. *VANT*. 2017. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/3cRGVwD>>.
- AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2019.
- Amro. *What is a Learning Curve in machine learning?* 2011. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/3CM8L89>>.
- ANDERSON, J. D.; BOWDEN, M. L. Introduction to flight. McGraw-Hill Higher Education, 2005.
- ANZAI, Y. *Pattern recognition and machine learning*. [S.l.]: Elsevier, 2012.
- ARCHETTI, F.; CANDELIERI, A. *Bayesian optimization and data science*. [S.l.]: Springer, 2019.
- ASSON, K. M.; DUNN, P. F. Compact dynamometer system that can accurately determine propeller performance. *Journal of Aircraft*, v. 29, n. 1, p. 8–9, 1992.
- BAILEY, J. Mini-rpv engine-propeller wind tunnel tests. In: *National Free Flight Society Annual Symposium Proceedings*. [S.l.: s.n.], 1978.
- BASS, R. Small scale wind tunnel testing of model propellers. In: *24th aerospace sciences meeting*. [S.l.: s.n.], 1986. p. 392.
- BENGFORT, B. et al. *Yellowbrick*. 2018. Disponível em: <<http://www.scikit-yb.org/en/latest/>>.
- BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, v. 24, 2011.
- BILOGUR, A. Missingno: a missing data visualization suite. *Journal of Open Source Software*, The Open Journal, v. 3, n. 22, p. 547, 2018. Disponível em: <<https://doi.org/10.21105/joss.00547>>.
- Bonfim. *O que é bias-variance tradeoff*. 2020. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/3BN2XKk>>.
- BRANDT, J.; SELIG, M. Propeller performance data at low reynolds numbers. In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. [S.l.: s.n.], 2011. p. 1255.
- BUITINCK, L. et al. API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. [S.l.: s.n.], 2013. p. 108–122.
- BUSCEMA, M.; TERZI, S.; TASTLE, W. A new meta-classifier. In: *2010 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, Toronto, ON, Canada. [S.l.: s.n.], 2010. p. 1–7.

- BUUREN, S. V. *Flexible imputation of missing data*. [S.l.]: CRC press, 2018.
- CARDOSO; QUEIROS, J. de S.; SANTOS, W. A utilização de veículos aéreos não tripulados (vants) como ferramenta na conservação e no monitoramento ambiental da amazônia brasileira. *Anais dos Encontros Nacionais de Engenharia e Desenvolvimento Social-ISSN 2594-7060*, v. 15, n. 1, 2018.
- CHEN. *What is the difference between the R gbm (gradient boosting machine) and xgboost (extreme gradient boosting)?* 2015. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/3q9LA4f>>.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794.
- CHOLLET, F. *Deep learning with Python*. [S.l.]: Simon and Schuster, 2021.
- DOHERTY, P.; RUDOL, P. A uav search and rescue scenario with human body detection and geolocalization. In: SPRINGER. *Australasian Joint Conference on Artificial Intelligence*. [S.l.], 2007. p. 1–13.
- DRELA, M. Qprop: Propeller/windmill analysis and design. 2007.
- DRELA, M.; YOUNGREN, H. Xrotor user guide. *Massachusetts Institute of Technology*, 2003.
- DUQUETTE, M. M.; VISSER, K. D. Numerical implications of solidity and blade number on rotor performance of horizontal-axis wind turbines. *J. Sol. Energy Eng.*, v. 125, n. 4, p. 425–432, 2003.
- DURAND, W. F.; LESLEY, E. P. Experimental research on air propellers v. 1923.
- FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001.
- FRIEDMAN, J. H. *The elements of statistical learning: Data mining, inference, and prediction*. [S.l.]: springer open, 2017.
- FURTADO, V. H. et al. Aspectos de segurança na integração de veículos aéreos não tripulados (vant) no espaço aéreo brasileiro. In: SN. *Anais do VII Simposio de Transporte aereo-Sitraer7*. [S.l.], 2008. p. 506–517.
- GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: O’Reilly Media, 2019.
- GLAUERT, H. *The elements of aerofoil and airscrew theory*. [S.l.]: Cambridge university press, 1983.
- GYLES, B. R. Propselector help. *Gyles AeroDesign*, v. 46, 1999.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>.

- KEANE, A.; FORRESTER, A.; SOBESTER, A. *Engineering design via surrogate modelling: a practical guide*. [S.l.]: American Institute of Aeronautics and Astronautics, Inc., 2008.
- LESLEY, E. Propeller tests to determine the effect of number of blades at two typical solidities. 1939.
- MATIAS, G. R. d. M.; GUZATTO, M. P.; SILVEIRA, P. G. Mapeamento topográfico cadastral por integração de imagens adquiridas com vant a técnicas tradicionais. 2014.
- MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56.
- MERCHANT, M.; MILLER, L. S. Propeller performance measurement for low reynolds number uav applications. In: *44th AIAA Aerospace Sciences Meeting and Exhibit*. [S.l.: s.n.], 2006. p. 1127.
- MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012.
- OL, M.; ZEUNE, C.; LOGAN, M. Analytical/experimental comparison for small electric unmanned air vehicle propellers. In: *26th AIAA Applied Aerodynamics Conference*. [S.l.: s.n.], 2008. p. 7345.
- ROSA, E. da; TOPOROSKI, J. *Introdução ao Projeto Aeronáutico: uma contribuição à competição SAE AeroDesign*. [S.l.]: UFSC, Centro Tecnológico, 2006.
- ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.
- SANTOS, M. C. D. *Analytical Model for the Performance Curves of a Family of Propellers Based on Wind Tunnel Tests*. Tese (Doutorado) — Universidade da Beira Interior (Portugal), 2018.
- Scaccia. *Validação Cruzada Aninhada com Scikit-learn*. 2020. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/3IC6w9I>>.
- SILVA, J. d. A. *Substituição de valores ausentes: uma abordagem baseada em um algoritmo evolutivo para agrupamento de dados*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- SILVESTRE, M. A.; MORGADO, J. P.; PASCOA, J. Jblade: a propeller design and analysis code. In: *2013 International Powered Lift Conference*. [S.l.: s.n.], 2013. p. 4220.
- STACK, J. et al. Investigation of the naca 4-(3)(8)-045 two-blade propellers at forward mach numbers to 0.725 to determine the effects of compressibility and solidity on performance. 1950.
- WASKOM, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, The Open Journal, v. 6, n. 60, p. 3021, 2021. Disponível em: <<https://doi.org/10.21105/joss.03021>>.
- Wikipedia. *Rotor Solidity*. 2021. Online, Data de Acesso: 5 de novembro de 2021. Disponível em: <<https://bit.ly/2ZJYrih>>.

YEH, I.-C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, Elsevier, v. 28, n. 12, p. 1797–1808, 1998.