

Aplicação de Métodos de Aprendizado Supervisionado para Estimativa da Vida Útil Restante em Motores Turbofan

Arthur Souto Lima¹

¹Departamento de Ciência da Computação (DCC)

Universidade Federal de Minas Gerais (UFMG)

Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901

arthursl@ufmg.br

Abstract. *Following the transformations brought by Industry 4.0, predictive maintenance is a field of study aiming to improve industrial maintenance. One parameter frequently studied is the Remaining Useful Life (RUL) of a system or a machine. With this purpose, supervised learning models were trained using NASA's turbofan engine dataset in order to predict RUL from sensor data. Tree-based models stood out due to their good performance across all scenarios. In addition, in order to improve model explainability, analyses were made using the SHAP library.*

Resumo. *Na esteira das transformações da Indústria 4.0, a manutenção preditiva é um dos ramos de estudo para melhoria dos processos de manutenção industrial e um dos parâmetros recorrentemente estudados é a Vida Útil Restante (RUL, Remaining Useful Life) de um sistema ou equipamento. Com esse objetivo, foram treinados modelos de aprendizado supervisionado a partir do dataset de motores turbofan da NASA, a fim de se estimar a RUL com base nos dados de vários sensores. Os modelos baseados em árvores tiveram destaque devido a sua boa performance em todos os cenários. Além disso, visando aumentar a explicabilidade dos modelos treinados, análises foram feitas a partir da biblioteca SHAP.*

1. Introdução

Os recentes avanços industriais trazidos pelo contexto da Indústria 4.0 abarcam uma série de mudanças advindas da implantação de novas tecnologias. Essa nova fase trouxe benefícios, mas também novos desafios a serem investigados e superados. Com integração de outras áreas do conhecimento, como a Ciência da Computação, principalmente às Engenharias de Produção e Automação, são possibilitadas novas abordagens no contexto industrial, por exemplo no âmbito da comunicação, com a Internet das Coisas (IoT, Internet of Things), ou no setor de projetos, com sistemas cyberfísicos [Peres et al. 2020].

Nesse contexto, a inteligência artificial (IA) também se mostra como um catalisador de soluções. As técnicas e métodos de IA garantem aos sistemas e aos processos uma capacidade de interagir com o ambiente em que se encontram, adquirindo dados e resolvendo problemas complexos, juntamente com a possibilidade de aprenderem com a experiência e se tornarem melhores em suas respectivas tarefas [Peres et al. 2020]. Nesse momento de recente introdução dessas abordagens ao setor industrial, no âmbito da Indústria 4.0, devido à grande produção de dados nos mais diversos sistemas integrados,

são necessárias soluções focadas neles [Angelopoulos et al. 2020], as quais usualmente advêm do aprendizado de máquina.

O aprendizado de máquina (ML, machine learning) é uma sub-área da inteligência artificial, centrada nos dados, os quais servem de base para criação de modelos, a fim de se realizar análises e previsões [Russell and Norvig 2020]. Todos os dias surgem novas estratégias e formas de se refinar esses modelos, bem como novos campos de aplicação, inclusive o industrial, mais recentemente [Peres et al. 2020]. As técnicas e métodos desse campo, juntamente com a maior capacidade de processamento possibilitado pela tecnologia atual, permitem abordagens via aprendizado supervisionado, semi-supervisionado, não-supervisionado, aprendizado por reforço e, ainda, aprendizado profundo [Jagatheesaperumal et al. 2021].

No aprendizado supervisionado, há diversas abordagens que usam algoritmos como Support Vector Machine (SVM), k-Nearest-Neighbors (kNN) e Random Forests (RF) [Peres et al. 2020, Silvestrin et al. 2019]. O aprendizado profundo, apesar de, por vezes, ainda ser incipiente nesse ramo industrial, também tem suas soluções, sejam elas baseadas em métodos já consagrados ou mesmo estratégias especializadas para a indústria. Contudo, devido a uma escassez, muitas vezes, de uma vastidão de dados necessária para treinar tais arquiteturas, seu uso ainda é preambular [Silvestrin et al. 2019].

Outra abordagem recente advinda da aplicação de outros conhecimentos na indústria está relacionada à manutenção. Na literatura, pode-se elencar três grandes tipos de manutenção [Carvalho et al. 2019]: a manutenção corretiva ou Run-to-Failure (R2F), a manutenção preventiva (PvM, preventive maintenance) e a manutenção preditiva (PdM, predictive maintenance). A primeira é a manutenção mais simples, pois as correções só são aplicadas quando há alguma falha. No outro lado do espectro, a manutenção preventiva utiliza um raciocínio que programa as revisões periodicamente para evitar as quebras ou as falhas.

O terceiro tipo se vale de estratégias baseadas na análise de dados. Usando ferramentas, estima-se quando será necessária uma manutenção ou mesmo quanto de vida útil aquele equipamento ainda possui antes de ser preciso uma substituição ou manutenção. Desse modo, técnicas estatísticas e de aprendizado de máquina permitem uma abordagem para alcançar essas soluções, no intuito de maximizar o tempo de operação das linhas produtivas [Silvestrin et al. 2019, Zonta et al. 2020].

A adoção, paulatina e gradual, dessas técnicas nesse contexto industrial traz uma série de benefícios a diversos agentes envolvidos direta ou indiretamente, nos vários setores em que essa integração pode ser observada [Peres et al. 2020, Jagatheesaperumal et al. 2021]. No âmbito de uma produção preditiva, pode-se prever as necessidades dos clientes e do mercado a priori e adaptar a produção a isso, retornando mais lucro à empresa. Por outro lado, com a manutenção preditiva, há uma melhora na confiabilidade dos equipamentos e dos sistemas, o que otimiza os custos com reparos e garante mais segurança a todos os envolvidos com a produção [Jagatheesaperumal et al. 2021].

1.1. Problema e Objetivos

No âmbito da manutenção preditiva, tem-se a área de Prognósticos e Gerência de Saúde (PHM, Prognostics and Health Management), a qual se dedica a estimar o estado do equi-

pamento ou do sistema num momento posterior [Leser 2017, Wang et al. 2020]. Mais precisamente, é necessário saber em que momento o sistema não estará mais em um estado de funcionamento, ou quanto tempo até que isso ocorra, e essa informação é essencial para o planejamento das manutenções [Berghout and Benbouzid 2022]. A estimativa, portanto, da Remaining Useful Life (RUL, vida útil restante) pode ser feita a partir de modelos físicos, a partir de dados ou com base numa estratégia híbrida [Wang et al. 2020].

Na abordagem a partir dos dados, ML apresenta diversos algoritmos, com diferentes vantagens e desvantagens. Por isso, é importante investigar e compreender alguns modelos de aprendizado supervisionado para a estimativa da RUL.

Além disso, como ressaltam [Peres et al. 2020, Ferreira and Gonçalves 2022], é relevante que os modelos usados para previsões na indústria não sejam aplicados apenas como uma "caixa-preta", isto é, seus resultados devem ter uma interpretabilidade e explicabilidade. Essas exigências podem advir de medidas regulatórias ou mesmo pela necessidade de uma maior transparência desses métodos e, portanto, devem ser uma questão de importância no projeto também.

2. Trabalhos Relacionados

A manutenção preditiva é um assunto bastante estudado atualmente, com novas abordagens e métodos sendo propostos recorrentemente. Artigos recentes usaram aprendizado de máquina para avaliar a estabilidade de sistemas hidráulicos a partir de árvores de decisão [Silvestrin et al. 2019], manutenção em veículos usando Random Forests [Prytz et al. 2015] ou mesmo estratégias usando vários algoritmos [Susto et al. 2015].

Na área de PHM, um parâmetro muito comum de ser estimado é a RUL e, em abordagens dirigidas por dados, métodos de aprendizado são bastante utilizados, como por exemplo os baseados em SVM, tal qual visto em [Nuhic et al. 2013]. Além dos algoritmos convencionais, também são analisados métodos de aprendizado profundo para a estimativa desse parâmetro em [Wang et al. 2020].

Dentre os conjuntos de dados usados normalmente para estudos da área, o dataset de motores turbofan da NASA [Saxena et al. 2008] é bastante utilizado em vários deles. Com ele, usualmente a tarefa é de estimar a RUL a partir dos dados dos sensores, seja, por exemplo, por métodos baseados em SVM [Tang et al. 2021], por métodos mais convencionais de aprendizado de máquina como k-Nearest-Neighbors (kNN) e Random Forest, vistos em [Mathew et al. 2017], ou mesmo por aprendizado profundo, como comentado por [Berghout and Benbouzid 2022].

3. Metodologia

3.1. Visão Geral

O projeto segue o fluxo de trabalho de aprendizado de máquina. Inicialmente, separa-se os dados em treino e teste e usa-se apenas o primeiro grupo para elaborar e refinar os modelos. As primeiras investigações são centradas numa análise mais exploratória das informações, para, em seguida, focar-se na implementação dos modelos de aprendizado supervisionado em si. Em todos eles, buscou-se realizar uma sintonia de seus hiperparâmetros visando melhores resultados e, além disso, fez-se a validação cruzada a fim de

aperfeiçoar seu poder de generalização. Com os eles já finalizados, conduz-se uma etapa de análise dos mesmos, visando incrementar sua interpretabilidade e explicabilidade.

Essa sintonia dos hiperparâmetros se deu via *halving search*, estratégia semelhante a uma *grid search*, mas sem fazer uma busca exaustiva em todos os hiperparâmetros com todo o treino, isto é, faz-se uma avaliação dos candidatos iterativamente, refinando os modelos com cada vez mais dados na busca pela melhor configuração de parâmetros. Essa estratégia agiliza significativamente essa etapa de aperfeiçoamento dos modelos, especialmente quando há muitas amostras no conjunto de treino.

Tendo em vista que as tarefas de aprendizado são de regressão, escolheu-se como métricas de avaliação dos modelos o coeficiente de determinação, ou R^2 , e também a raiz do erro quadrado médio (RMSE, Root Mean Squared Error). A primeira é calculada diretamente pelas ferramentas usadas. A segunda normalmente é disponibilizada como erro quadrado médio e, por isso, exigiu uma adaptação para que fosse usada a raiz dele.

Os modelos escolhidos para a investigação no projeto foram: kNN, Linear Regressor, SGD Regressor, Linear Support Vector Regressor (SVR), Decision Tree (DT) e RF. Além deles, também utilizou-se o Dummy Regressor para se ter um desempenho-base de referência. Em todos eles, testou-se também a adição de features polinomiais (Polynomial Features).

Em termos de ferramentas, o código foi escrito em Python e executado em forma de notebooks Jupyter no Google Colab. Para a estruturação dos dados, usou-se a biblioteca Pandas [Wes McKinney 2010] e a NumPy [Harris et al. 2020]. Para os modelos, foi utilizada a biblioteca Scikit-Learn [Pedregosa et al. 2011]. Para as análises de explicabilidade, a biblioteca SHAP (SHapley Additive exPlanations) [Lundberg and Lee 2017]. Por fim, em relação aos gráficos e artefatos visuais, a biblioteca Matplotlib [Hunter 2007].

3.2. Dataset

A primeira etapa do projeto foi a escolha do conjunto de dados a ser utilizado. Foi selecionado o dataset de motores Turbofan, montado pela NASA [Saxena et al. 2008]. Um esquema desse equipamento pode ser visto na figura 1, juntamente com um diagrama dos módulos internos e suas conexões na figura 2.

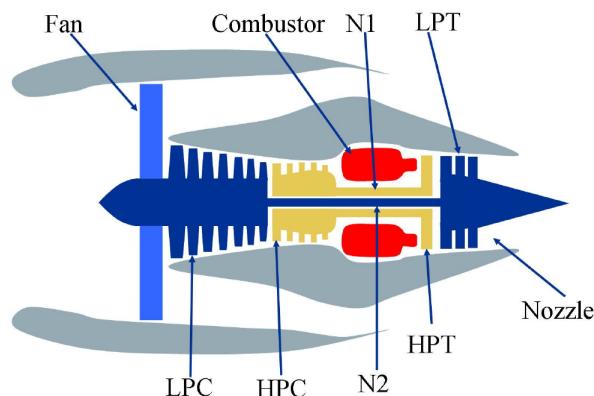


Figura 1. Diagrama simplificado do motor turbofan [Saxena et al. 2008]

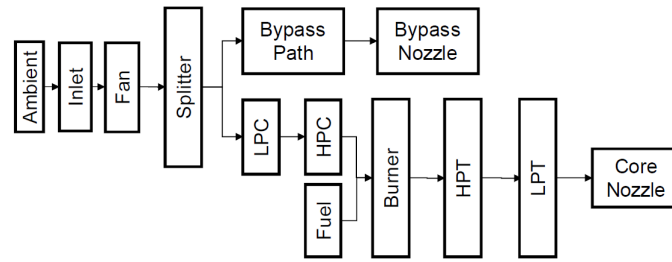


Figura 2. Esquema dos módulos do motor turbofan [Saxena et al. 2008]

Esses dados foram gerados a partir de simulações do *Commercial Modular Aero-Propulsion System Simulation* (C-MAPSS), uma ferramenta para a simulação realística de grandes motores turbofan comerciais, em que pode-se definir uma série de parâmetros de entrada para balizar os experimentos e obter vários valores como saída. No dataset, as saídas disponibilizadas são a de 21 sensores, listados na tabela 1, juntamente com três parâmetros que indicam a configuração daquela simulação com respeito à altitude de voo, número de Mach e temperatura.

Tabela 1. Descrição dos sensores [Saxena et al. 2008]

#	Symbol	Description	Unit
0	T2	Total temperature at fan inlet	°R
1	T24	Total temperature at LPC outlet	°R
2	T30	Total temperature at HPC outlet	°R
3	T50	Total temperature at LPT outlet	°R
4	P2	Pressure at fan inlet	psia
5	P15	Total pressure in bypass-duct	psia
6	P30	Total pressure at HPC outlet	psia
7	Nf	Physical fan speed	rpm
8	Nc	Physical core speed	rpm
9	epr	Engine pressure ratio (P50/P2)	—
10	Ps30	Static pressure at HPC outlet	psia
11	phi	Ratio of fuel flow to Ps30	pps/psi
12	NRf	Corrected fan speed	rpm
13	NRc	Corrected core speed	rpm
14	BPR	Bypass Ratio	—
15	farB	Burner fuel-air ratio	—
16	htBleed	Bleed Enthalpy	—
17	Nf_dmd	Demanded fan speed	rpm
18	PCNfR_dmd	Demanded corrected fan speed	rpm
19	W31	HPT coolant bleed	lbm/s
20	W32	LPT coolant bleed	lbm/s

Cada simulação é composta por séries temporais, uma para cada sensor, com uma medição por ciclo de operação. Cada motor começa num estado funcional, com algum grau de desgaste, mas que não afeta significativamente sua operação normal, até que, em

algum momento da simulação, ocorre algum defeito ou uma falha, que aumenta até a falha completa do sistema, isto é, tem-se o run-to-failure. O objetivo é, portanto, estimar quantos ciclos de operação ainda restam para aquele motor, dados aquelas leituras dos sensores. Em outras palavras, a meta é estimar a RUL do motor, dadas as informações de seus sensores.

Tabela 2. Especificações dos grupos do dataset

Grupo de Dados	FD001	FD002	FD003	FD004
Simulações Treino	100	260	100	248
Simulações Teste	100	259	100	249
Condições de Simulação	1	6	1	6
Tipos de Falhas	1	1	2	2

O dataset é dividido em quatro grupos, com diferentes graus de complexidade para essa tarefa. Os dados vêm previamente separados em treino e em teste para a elaboração dos modelos. Como é possível ver na tabela 2, o grupo FD001 é o mais simples, ao passo que o FD004 é o mais complexo. No FD001 e FD003, a única condição de simulação é no nível do mar. Nos outros dois, são seis diferentes condições. No FD001 e FD002, há apenas um tipo de defeito, causado por degradação no Compressor de Alta Pressão (HPC, High-Pressure Compressor). Nos outros dois, há, além desse possível defeito, também degradação na ventoinha (fan) do motor.

3.3. Cálculo da RUL

No dataset, somente há anotações de RUL corretas no conjunto de teste. Por isso, para o treino, é preciso fazer um processamento inicial das leituras dos sensores. Inicialmente, o intuito era calcular a RUL de forma linear, ou seja, a cada ciclo, ela diminui em uma unidade. Tendo em vista que as simulações do treino são run-to-failure, pode-se definir que o último ciclo de cada observação tem RUL 0 e incrementar os antecessores linearmente, como descrito.

Porém, como ressaltam [Berghout and Benbouzid 2022, Saxena et al. 2008], nessas simulações de treino, os motores apresentam leituras de forma que eles tendem a manter o seu funcionamento até que, em algum momento, ocorre uma falha, ou um princípio de falha, e, a partir desse instante, se nada for feito o sistema apresentará instabilidades graves. Assim, há uma não-linearidade nesse valor, ou seja, de início, não é possível saber quando haverá um problema, apenas quando surgirem princípios de um defeito pode-se estimar um valor.

Por exemplo, na figura 3, nota-se a leitura de um sensor para uma simulação do grupo FD001. Do início das leituras até o aproximadamente o ciclo 270, seu valor fica num patamar em torno de 521.5. Após esse instante, com o surgimento do defeito, é possível estimar quantos ciclos até a falha. Porém, antes disso, uma RUL totalmente linear não capturaria esse comportamento de continuidade de funcionamento até o aparecimento da irregularidade.

Em termos de modelagem, essa não-linearidade foi implementada como mais um parâmetro dos modelos, a ser sintonizado com o refino, por meio da função *clip* da biblioteca Numpy. Ela era feita logo antes do treino do modelo, sendo aplicada no vetor de

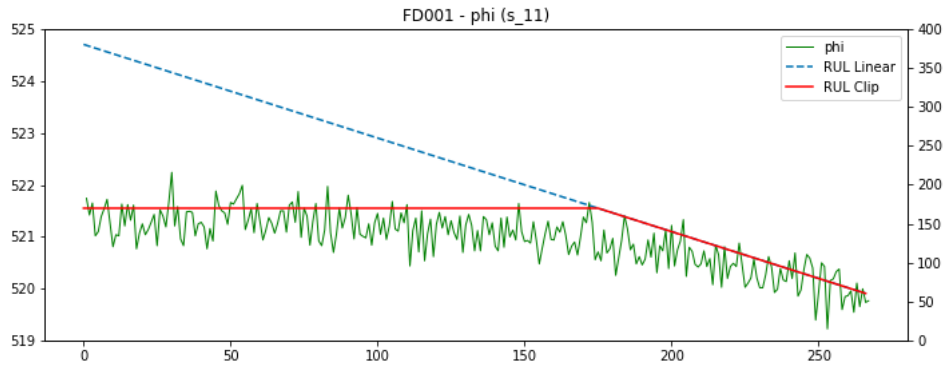


Figura 3. Ilustração da não-linearidade

resposta (y). Assim, por exemplo, um vetor com o comportamento da linha tracejada do gráfico 3 era transformado na linha vermelha contínua através da função *clip*.

4. Resultados e Discussão

O desempenho de cada modelo nos conjuntos de teste pode ser observado nas tabelas a seguir. Na tabela 3, vê-se a raiz do erro quadrado médio, enquanto na 4, é possível ver o coeficiente de determinação R^2 . Estão dispostos apenas os dados dos modelos que usaram a RUL não-linear, tendo em vista que os com RUL linear tiveram um desempenho significativamente pior em todos os casos. Ainda assim, os resultados com esses outros modelos podem ser obtidos do apêndice A.

Tabela 3. RMSE, usando RUL não-linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	45,115	53,834	44,158	60,690
	kNN	28,477	15,957	16,215	14,690
	Linear Regressor	25,526	17,697	20,287	20,646
	SGD Regressor	23,645	18,286	17,036	-27,183
	Linear SVR	25,807	16,424*	18,062	21,377*
	DT	14,227	15,532	15,577	17,293
	RF	12,761	12,769	13,237	13,767
Polynomial Features	Dummy Regressor	44,864	53,879	43,915	60,381
	kNN	21,998	11,595	11,266	16,763
	Linear Regressor	26,415	16,445	17,117	21,234
	SGD Regressor	22,171	13,885	19,049	21,902
	Linear SVR	14,641	15,647*	14,513	-98,189**
	DT	12,322	12,161	14,454	15,868
	RF	13,609	10,122	15,112	10,943

*Teve problemas de convergência

** Não convergiu

Essa diferença quanto ao resultado usando uma RUL linear reitera uma máxima do aprendizado de máquina e da computação em geral: "garbage in, garbage out", em

Tabela 4. R^2 , usando RUL não-linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	-0,179	-0,002	-0,138	-0,239
	kNN	0,530	0,742	0,761	0,770
	Linear Regressor	0,623	0,767	0,720	0,687
	SGD Regressor	0,676	0,764	0,755	0,647
	Linear SVR	0,614	0,766*	0,744	0,682*
	DT	0,819	0,800	0,756	0,742
	RF	0,839	0,839	0,812	0,803
Polynomial Features	Dummy Regressor	-0,166	-0,004	-0,125	-0,226
	kNN	0,720	0,812	0,780	0,793
	Linear Regressor	0,596	0,766	0,753	0,686
	SGD Regressor	0,715	0,761	0,733	0,618
	Linear SVR	0,789	0,774*	0,764	-4,017**
	DT	0,711	0,794	0,776	0,768
	RF	0,838	0,857	0,776	0,820

*Teve problemas de convergência

** Não convergiu

que dados mal tratados levam a resultados imprecisos, ou mesmo inconsistentes. No caso de ML, nem mesmo bons algoritmos, com uma excelente sintonia de hiperparâmetros conseguirão superar uma entrada com problemas.

Outra observação importante é a necessidade do uso e compreensão de métricas para a avaliação do desempenho dos métodos. Se considerarmos apenas a RMSE, vê-se que até mesmo o Dummy Regressor apresenta valores aceitáveis, principalmente no treino. Contudo, ele não generaliza bem, o que é denotado por seu coeficiente de determinação bastante baixo, isto é, a variância dos dados de entrada é pouco explicada pelo modelo. Assim, nas análises é importante conjugar as várias métricas a fim de se obter conclusões mais abalizadas.

É interessante também ressaltar a diferença de horizonte de tempo necessária para se treinar alguns modelos aqui retratados. Espera-se, naturalmente, que, com um aumento de complexidade, seja do problema, seja dos dados de entrada (com mais atributos, via adição de fatores polinomiais, por exemplo), que o tempo de treinamento e sintonia aumentem. Contudo, alguns modelos já partiam de altos tempos mesmo nas configurações mais simples. Por exemplo, o kNN, por ser baseado em distância e, frente a uma quantidade alta de amostras (no dataset, cada ciclo de cada simulação era uma amostra), pode apresentar uma complexidade maior. Além disso, o Linear SVR também apresentou essas dificuldades. Ademais, algumas vezes, esse modelo exibia problemas de convergência durante o treinamento, com o aparecimento de alguns avisos advindos da biblioteca. Em um único caso, no dataset mais difícil, FD004, e com os fatores polinomiais, ele não foi capaz de convergir, nem mesmo com mais iterações.

Um fato recorrente, durante o treinamento entre os grupos de dados, era que alguns modelos se repetiam, mesmo após a sintonia dos hiperparâmetros. A título de exemplo, o

modelo da DT usando RUL não-linear e com as features básicas foi exatamente o mesmo para todos os quatro datasets. Além dele, os modelos da RF usando RUL não-linear do FD004 também foram os mesmos dos seus correspondentes no FD002. Embora um acontecimento inesperado, é possível explicar essa similaridade devido ao fato de que os grupos de dados são parecidos entre si, sobretudo o FD001 e FD003, tal qual o FD002 e FD004.

Avaliando o eventual benefício das features polinomiais, nota-se que, em geral, ela auxiliou os modelos, principalmente os lineares. Percebe-se uma redução do erro médio quadrático e um aumento do coeficiente de determinação. Em alguns casos, como algumas configurações dos baseados em árvore, o uso desses atributos diminui o R^2 , mas, no entanto, melhora o RMSE. Vê-se, nesses casos, portanto, um tradeoff a ser avaliado também entre complexidade, tempo de treino e desempenho.

Em linhas gerais, nota-se que os modelos baseados em árvore tiveram os melhores desempenhos em todos os grupos de dados. Dentre os lineares, há de se destacar também a performance do regressor via gradiente descendente (SGD Regressor) e do Linear SVR, que, apesar de alguns poucos problemas de convergência, apresentou bons resultados.

Por fim, na comparação com outros estudos com o mesmo conjunto de dados, vê-se que resultados são semelhantes. Nos modelos baseados em SVM de [Tang et al. 2021], obteve-se uma RMSE de aproximadamente 25 no FD001, próximo ao obtido aqui sem a utilização de features polinomiais pelo LinearSVR, na tabela 3. Além disso, contraste com o obtido por [Mathew et al. 2017], nota-se uma semelhança nos modelos lineares, mas, sobretudo nos modelos baseados em árvore, há melhores resultados aqui apresentados, seja pelo uso de features polinomiais ou mesmo por um possível diferente processo de sintonia dos hiperparâmetros.

4.1. Análise de Valores SHAP

Visando aumentar a explicabilidade e a transparência das estimativas feitas, serão realizadas análises dos valores SHAP para os modelos treinados, por meio da biblioteca homônima. Dentre os vários propostos pelo projeto, usaremos aqui apenas os RF, pois eles tiveram os melhores resultados e, além disso, a biblioteca possui otimizações para o cômputo dos valores SHAP a partir desse tipo de modelo.

Ademais, serão analisados apenas os modelos sem as features polinomiais, visto que surge um outro tradeoff no seu uso nesse momento. Com mais atributos no modelo, os valores SHAP ficam mais granulares e muitas features ficam com pouca importância, como é possível observar na figura 4. Assim, o tradeoff é entre ganhar desempenho ou obter uma maior explicabilidade dos modelos.

No FD001, na figura 5, nota-se que os atributos relacionados ao compressor HPC são os mais importantes, como o Ps30 e P30, juntamente com os de módulos após ele no diagrama funcional, figura 2, como o T50, do LPT. Esses componentes à frente são afetados diretamente por alguma alteração no funcionamento do HPC, que é exatamente o defeito que ocorre nas simulações do FD001. Assim, o modelo prioriza as informações do HPC e de módulos afetados para fazer suas estimativas.

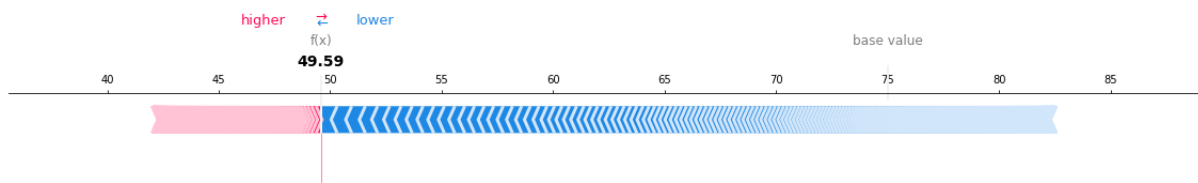


Figura 4. Valores SHAP de uma amostra com Polynomial Features

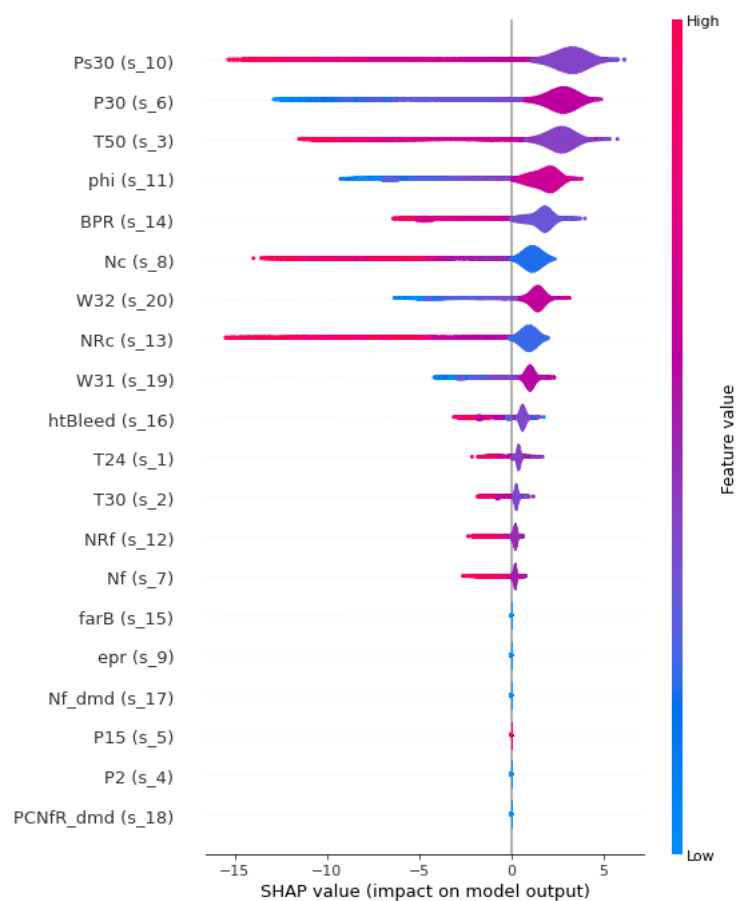


Figura 5. Valores SHAP para o grupo FD001

Além disso, a figura 5 denota que o Ps30 mais alto tende a diminuir a RUL de uma observação, em até 15 unidades, enquanto o P30 tem um efeito menor na diminuição, mas um fator positivo para o aumento da RUL.

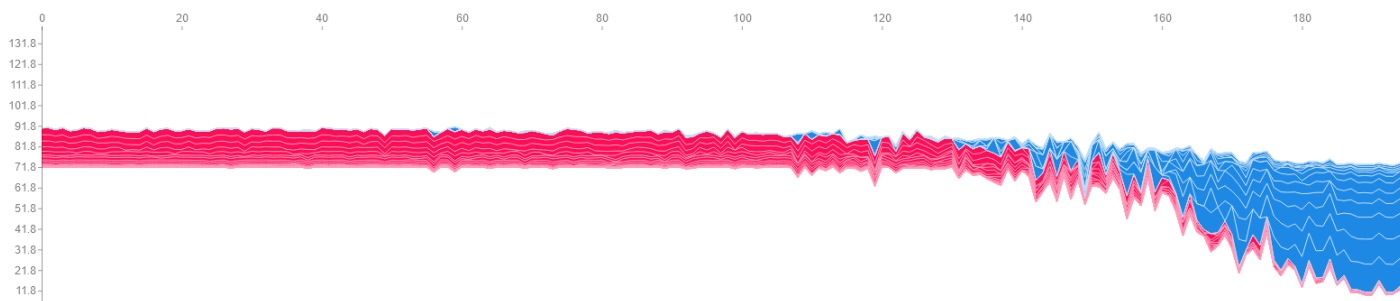


Figura 6. Evolução de valores SHAP da simulação 42 do FD001

Pode-se exemplificar esse comportamento com uma simulação feita, retratada na figura 6. Ao longo da simulação, é possível ver os parâmetros que contribuem para o aumento da estimativa da RUL, coloridos em tom avermelhado, a partir do valor médio, bem como os que contribuem para sua diminuição, coloridos em azul. O valor da estimativa está no limiar entre a área azul e a área avermelhada, de forma que sempre a azul fica por cima e a vermelha por baixo. Com isso, é possível perceber que, aproximadamente no ciclo 140, o modelo passa a detectar o defeito e o motor apresenta instabilidades cada vez maiores (área azul), até o fim da simulação.



Figura 7. Valores SHAP da simulação 42 do FD001

Fazendo um recorte nesse gráfico, é possível descobrir quais atributos contribuem em cada situação, na figura 7. Nota-se que os principais parâmetros elencados na figura 5 são exemplificados aqui, sendo que esses atributos, muito relacionados ao HPC e seus efeitos, inicialmente ajudam a denotar a estabilidade do sistema, mas, posteriormente, indicam o defeito e, por isso, a diminuição da RUL.

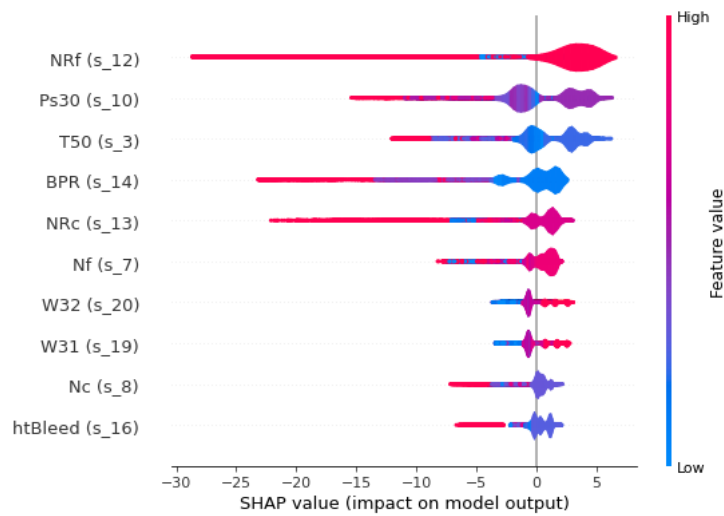


Figura 8. Dez maiores valores SHAP para o grupo FD002

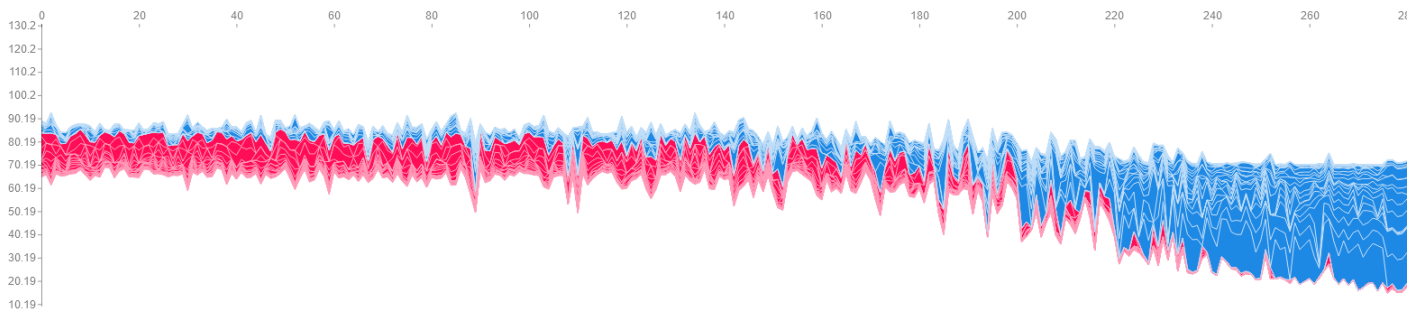


Figura 9. Evolução de valores SHAP da simulação 32 do FD002

Análises semelhantes podem ser feitas para os outros grupos de dados. No grupo FD002, retratado na figura 8, em que também há apenas um defeito, mas são feitas diferentes configurações de simulação, é esperado que os atributos ligados ao HPC, como Ps30 e T50 continuem importantes, o que de fato ocorreu. Contudo, nota-se alguns outros parâmetros ganhando importância como o NRf, relacionado à rotação da fan. E, além disso, os atributos de configuração da simulação não tiveram tanta relevância, não aparecendo, portanto, entre as dez features mais importantes.

Ao juntar essas observações com a evolução dos valores SHAP de uma simulação, na figura 9, nota-se uma maior instabilidade das estimativas, mesmo durante a fase inicial da experiência. Isso pode ser devido à alta importância de alguns atributos não ligados ao defeito em questão, mas que são influenciados pelas diferentes configurações, como salientado anteriormente. Apesar disso, ainda é possível perceber a mudança de tendência a partir dos defeitos, de forma semelhante ao que ocorre no FD001.

Nos dois últimos grupos, é possível fazer uma outra análise. Como agora são dois tipos de defeitos que podem ocorrer nos motores durante a simulação, e, nos dois primeiros datasets, o modelo usou os atributos ligados ao defeito para fazer a estimativa, espera-se que ele use, agora, também parâmetros ligados ao segundo tipo de falha para suas previsões. Como ressaltado previamente em [Saxena et al. 2008], o outro tipo de

problema possível está ligado a uma degradação da ventoinha.

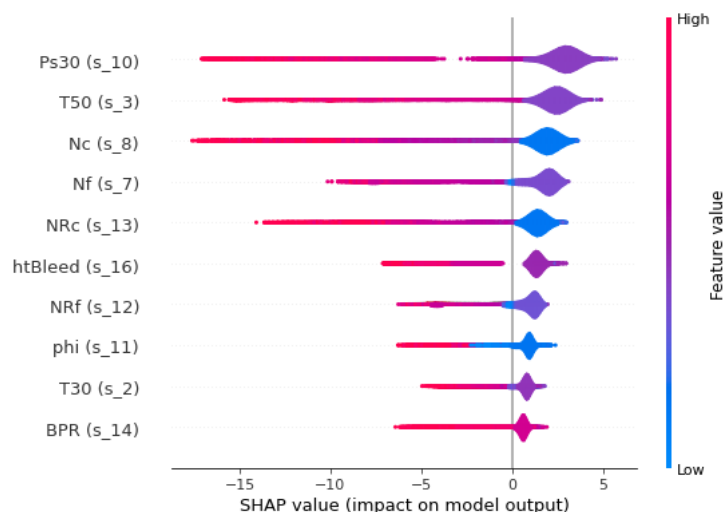


Figura 10. Dez maiores valores SHAP para o grupo FD003

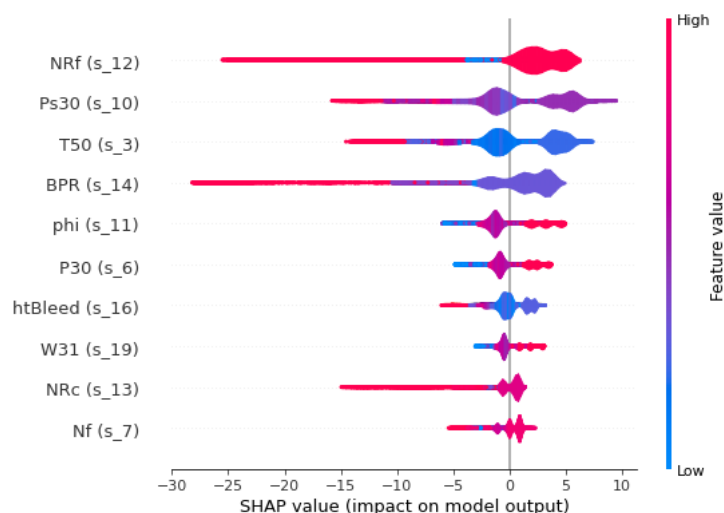


Figura 11. Dez maiores valores SHAP para o grupo FD004

Conforme as expectativas, atributos relacionados à ventoinha, como Nc, Nf, NRc e Nrf, ganham destaque e, juntamente com eles, os já conhecidos ligados ao HPC continuam importantes para os modelos, como denotam as figuras 10 e 11. Dessa forma, é possível encontrar tanto simulações que apresentaram o defeito no HPC, com um comportamento semelhante à figura 7, como também aquelas com defeito na ventoinha, como visto na figura 12. Nesse caso, na simulação 42 do FD004, o parâmetro NRf foi bastante importante para a diminuição da RUL.

Desse modo, por meio da análise dos valores SHAP dos modelos, é possível evidenciar a importância dada aos diversos atributos, relacionando-as com os efeitos que geraram a base dados. Isso permite, por exemplo, identificar o momento em que o motor começou a apresentar o defeito, bem como quais sensores contribuíram para a estimativa de RUL feita pelo modelo. Analogamente, pode-se descobrir quais são os parâmetros

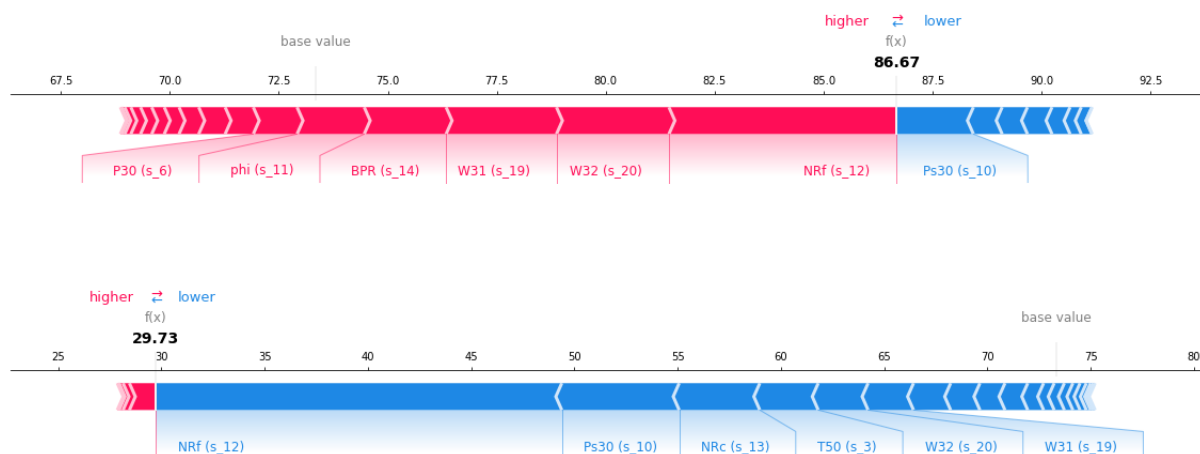


Figura 12. Valores SHAP da simulação 42 do FD004

que não tem relevância para essa tarefa. Além disso, os valores SHAP permitem quantificar essa importância para as amostras de entrada, com mais profundidade que apenas os vetores de coeficientes ou de importância de features próprios dos modelos em si.

Essa análise, na esteira das recentes discussões de ML acerca da explicabilidade de modelos, permite que eles não sejam tanto um método "caixa-preta". Em outras palavras, portanto, garante-se mais transparência e interpretabilidade aos resultados e previsões feitas.

5. Conclusão

Ao final deste projeto, foi possível averiguar o desempenho de diversos modelos de aprendizado supervisionado para a tarefa da previsão da RUL a partir de dados de sensores de motores turbofan. Após a análise de diferentes estratégias e configurações, constatou-se que os modelos baseados em árvore apresentaram um melhor desempenho em todos os cenários estudados, com um satisfatório grau de generalização. Além disso, por meio da análise de valores SHAP, pode-se interpretar e explicar as estimativas feitas pelos modelos nos vários cenários, o que garante mais transparência e explicabilidade para os resultados.

Como possibilidades de investigação futura, pode-se elencar a verificação do desempenho de arquiteturas de redes neurais ou mesmo de aprendizado profundo, que apresentam estratégias diferentes para sobrepujar essa tarefa. Além disso, pode-se selecionar uma outra base de dados para averiguar esses e outros modelos, a fim de se verificar as conclusões aqui salientadas.

Finalmente, é importante destacar que o projeto permitiu a aplicação de conhecimentos de ML, Ciência dos Dados, bem como diversas outras áreas da Ciência da Computação, com as quais houve contato ao longo da graduação. Além disso, foi possível vivenciar um fluxo de trabalho de aprendizado de máquina, incluindo a criação de modelos, com suas inerentes complexidades e particularidades, bem como implementar soluções para o tratamento inicial dos dados, para, ao final, com os modelos já treinados, ponderar acerca dos resultados de forma abalizada.

Referências

- [Angelopoulos et al. 2020] Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., and Zahariadis, T. (2020). Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors*, 20(1).
- [Berghout and Benbouzid 2022] Berghout, T. and Benbouzid, M. (2022). A systematic guide for predicting remaining useful life with machine learning. *Electronics*, 11(7).
- [Carvalho et al. 2019] Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P., and Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024.
- [Ferreira and Gonçalves 2022] Ferreira, C. and Gonçalves, G. (2022). Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of Manufacturing Systems*, 63:550–562.
- [Harris et al. 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- [Hunter 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- [Jagatheesaperumal et al. 2021] Jagatheesaperumal, S. K., Rahouti, M., Ahmad, K., Al-Fuqaha, A., and Guizani, M. (2021). The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions. *IEEE Internet of Things Journal*, pages 1–1.
- [Leser 2017] Leser, P. E. (2017). *Probabilistic Prognostics and Health Management for Fatigue-Critical Components using High-Fidelity Models*. PhD thesis, North Carolina State University.
- [Lundberg and Lee 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.
- [Mathew et al. 2017] Mathew, V., Toby, T., Singh, V., Rao, B. M., and Kumar, M. G. (2017). Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning. In *2017 IEEE International Conference on Circuits and Systems (ICCS)*, pages 306–311.
- [Nuhic et al. 2013] Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., and Dietmayer, K. (2013). Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources*, 239:680–688.
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

- [Peres et al. 2020] Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., and Barata, J. (2020). Industrial artificial intelligence in industry 4.0 - systematic review, challenges and outlook. *IEEE Access*, 8:220121–220139.
- [Prytz et al. 2015] Prytz, R., Nowaczyk, S., Rögnvaldsson, T., and Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering Applications of Artificial Intelligence*, 41:139–150.
- [Russell and Norvig 2020] Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, Englewood Cliffs, N.J.
- [Saxena et al. 2008] Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9.
- [Silvestrin et al. 2019] Silvestrin, L. P., Hoogendoorn, M., and Koole, G. (2019). A comparative study of state-of-the-art machine learning algorithms for predictive maintenance. In *SSCI*, pages 760–767.
- [Susto et al. 2015] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., and Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820.
- [Tang et al. 2021] Tang, R., Fang, G., Liu, G., and Wang, H. (2021). Propulsion life prediction based on support vector machine. *IOP Conference Series: Earth and Environmental Science*, 687(1):012082.
- [Wang et al. 2020] Wang, Y., Zhao, Y., and Addepalli, S. (2020). Remaining useful life prediction using deep learning approaches: A review. *Procedia Manufacturing*, 49:81–88. Proceedings of the 8th International Conference on Through-Life Engineering Services – TESConf 2019.
- [Wes McKinney 2010] Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.
- [Zonta et al. 2020] Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., and Li, G. P. (2020). Predictive maintenance in the industry 4.0: A systematic literature review. *Computers Industrial Engineering*, 150:106889.

A. Resultados Completos

Tabela 5. R^2 , usando RUL não-linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	-0,179	-0,002	-0,138	-0,239
	kNN	0,530	0,742	0,761	0,770
	Linear Regressor	0,623	0,767	0,720	0,687
	SGD Regressor	0,676	0,764	0,755	0,647
	Linear SVR	0,614	0,766*	0,744	0,682*
	DT	0,819	0,800	0,756	0,742
	RF	0,839	0,839	0,812	0,803
Polynomial Features	Dummy Regressor	-0,166	-0,004	-0,125	-0,226
	kNN	0,720	0,812	0,780	0,793
	Linear Regressor	0,596	0,766	0,753	0,686
	SGD Regressor	0,715	0,761	0,733	0,618
	Linear SVR	0,789	0,774*	0,764	-4,017**
	DT	0,711	0,794	0,776	0,768
	RF	0,838	0,857	0,776	0,820

*Teve problemas de convergência

** Não convergiu

Tabela 6. RMSE, usando RUL não-linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	45,115	53,834	44,158	60,690
	kNN	28,477	15,957	16,215	14,690
	Linear Regressor	25,526	17,697	20,287	20,646
	SGD Regressor	23,645	18,286	17,036	-27,183
	Linear SVR	25,807	16,424*	18,062	21,377*
	DT	14,227	15,532	15,577	17,293
	RF	12,761	12,769	13,237	13,767
Polynomial Features	Dummy Regressor	44,864	53,879	43,915	60,381
	kNN	21,998	11,595	11,266	16,763
	Linear Regressor	26,415	16,445	17,117	21,234
	SGD Regressor	22,171	13,885	19,049	21,902
	Linear SVR	14,641	15,647*	14,513	-98,189**
	DT	12,322	12,161	14,454	15,868
	RF	13,609	10,122	15,112	10,943

*Teve problemas de convergência

** Não convergiu

Tabela 7. R^2 , usando RUL linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	-0,604	-0,251	-2,298	-0,735
	kNN	0,177	0,474	-0,770	0,293
	Linear Regressor	0,409	0,602	-0,900	0,233
	SGD Regressor	0,472	0,632	-0,792	0,197
	Linear SVR	0,409	0,602*	-0,899	0,235*
	DT	0,298	0,589	-0,630	0,346
	RF	0,462	0,695	-0,311	0,424
Polynomial Features	Dummy Regressor	-0,604	-0,251	-2,298	-0,735
	kNN	0,322	0,592	-0,672	0,274
	Linear Regressor	0,409	0,602	-0,900	0,233
	SGD Regressor	0,432	0,612	-0,776	0,262
	Linear SVR	0,413*	0,604*	-0,685*	0,261*
	DT	0,311	0,633	-0,504	0,170
	RF	0,422	0,692	-0,314	0,446

*Teve problemas de convergência

** Não convergiu

Tabela 8. RMSE, usando RUL linear

Configuração	Modelo	FD001	FD002	FD003	FD004
Base Features	Dummy Regressor	52,625	60,162	75,180	71,828
	kNN	37,702	39,010	55,080	45,862
	Linear Regressor	31,950	33,943	57,057	47,758
	SGD Regressor	30,203	32,617	55,419	48,854
	Linear SVR	31,951	33,933*	57,039	47,691*
	DT	34,821	34,462	52,853	44,095
	RF	30,481	29,683	47,403	41,397
Polynomial Features	Dummy Regressor	52,625	60,162	75,180	71,828
	kNN	34,215	34,347	53,529	46,461
	Linear Regressor	31,950	33,943	57,057	47,758
	SGD Regressor	31,319	33,519	55,173	46,840
	Linear SVR	31,831*	33,823*	53,734*	46,887*
	DT	34,484	32,560	50,765	49,677
	RF	31,601	29,844	47,460	40,577

*Teve problemas de convergência

** Não convergiu