

MODELO DE *MACHINE LEARNING* PARA VALIDAÇÃO DE DADOS DE DEFEITO DAS AERONAVES T-27

MACHINE LEARNING MODEL FOR T-27 AIRCRAFTS FAILURE DATA VALIDATION

Lucas Antônio Silva¹
Rachel Andrade Ballardin²
Jardel Figueira da Silva³

RESUMO

Este trabalho analisa o desempenho de modelos de *machine learning* para validação de dados de defeito das aeronaves T-27. Tal abordagem se justifica pela grande quantidade de entradas inadequadas desses dados, diminuindo a confiança nos indicadores de confiabilidade que deles dependem e exigindo em suas análises uma conferência individual que possui potencial de ser automatizada. O objetivo deste estudo é levantar técnicas de machine learning aplicáveis à classificação dos textos descritivos das possíveis falhas e analisar suas performances na filtragem desses dados. Este propósito foi alcançado mediante pesquisa bibliográfica na área de *machine learning* aplicada à classificação de textos, seguida de um estudo de caso para os dados das aeronaves T-27, que passou por validar manualmente mais de 7000 Fichas de Coleta de Dados de Defeito. A análise evidenciou que o melhor modelo obtido, uma combinação de modelos *Support Vector Classifier*, Regressão Logística e Naive Bayes Multinomial, possui F1-macro de 0,9454 e acurácia de 96,33%, sendo esta performance superior à das técnicas tradicionais que são empregadas atualmente. O modelo também se mostrou promissor na correção de indicadores gerenciais de confiabilidade.

Palavras-chave: Dados de Defeito. Machine Learning. Confiabilidade.

ABSTRACT

This paper analyses the performance of machine learning models for T-27 aircrafts failure data validation. Such an approach is justified by the large number of inadequate inputs of these data, reducing the confidence in the reliability indicators that depend on them and demanding in its analysis an individual checking that could be automated. The object of this study is to survey machine learning techniques which can be applied to text classification of the possible failures and to analyze their performances on the data filtering. This purpose was achieved through bibliographic research in the machine learning field study applied to text classification, followed by a case study to the T-27 aircraft, which included a manual validation of more than 7000 data

¹ Pós-graduando em Logística Empresarial pelo Centro Universitário do Sul de Minas (UNISMG). Graduação em Engenharia Mecânica-Aeronáutica pelo Instituto Tecnológico de Aeronáutica (2017). E-mail: lcsilvap@gmail.com.

² Doutorado em Educação pela Universidade Metodista de Piracicaba – UNIMEP (2016). Mestrado em Engenharia Civil na área de concentração: Infraestrutura e gerência viária com ênfase em transportes pela Universidade Federal de Santa Catarina – UFSC (2009). E-mail: ballardinchel@gmail.com.

³ Graduação em Engenharia Aeronáutica pelo Instituto Tecnológico de Aeronáutica (2003) e Mestrado em Ciências e Tecnologias Espaciais pelo Instituto Tecnológico de Aeronáutica (2018). E-mail: jardeljfs@fab.mil.br

failure cards. The analysis enlightened that the best obtained model, a combination of Support Vector Classifier, Logistic Regression and Multinomial Naive Bayes models, showed a F1-macro score of 0,9454 and an accuracy score of 96,33% - superior performance when compared to traditional approaches. The model also has shown being promising for reliability indicators correction.

Keywords: Failure Data. Machine Learning. Reliability.

1 INTRODUÇÃO

Desde o início da Revolução Industrial, um dos grandes desafios do homem é gerir o suporte e a segurança dos equipamentos que constrói. Administrar a cadeia relacionada a essas máquinas exige alguma previsibilidade do tempo e das condições em que cada componente funcionará, processo cada vez mais facilitado pela crescente disponibilidade de dados e de recursos computacionais.

No mundo aeronáutico, após a Segunda Guerra Mundial, novos sistemas foram desenvolvidos e utilizados nas frotas comerciais, exigindo a evolução dos métodos de manutenção de modo a se obter o maior desempenho a menor custo possível.

Na Força Aérea Brasileira, a institucionalização da Manutenção Centrada na Confiabilidade se deu na ICA 400-21 e no MCA 400-15, ambos de 2006, definindo uma estrutura sistêmica dentro da Força e determinando responsabilidades para diferentes órgãos, descrevendo o processo prático da aplicação da metodologia. Apesar da aplicação de análises de confiabilidade no grande cenário serem associadas nos regulamentos à definição de programas de manutenção, elas são aplicadas rotineiramente em problemas mais específicos nos Parques de Material Aeronáutico, como análises de segurança, decisão de prorrogação de uso de componentes, aferição de efetividade de substituição de equipamentos, definição de tamanho de giro, previsão de necessidade de itens para gestão de obsolescência, entre outras.

A principal fonte de dados para a aplicação das técnicas de modelagem previstas nessas normativas, são os dados de falha do SILOMS (Sistema Integrado de Logística de Material e Serviços), sistema informatizado onde as falhas de cada componente são introduzidas pelos operadores das aeronaves em uma FCDD (Ficha de Coleta de Dados de Defeito). Outras fontes de dados podem auxiliar no aprofundamento das análises de confiabilidade (Silva, 2018), entretanto, são mais difusas e de difícil aglutinação, o que mantém uma FCDD com preenchimento correto e completo o instrumento mais efetivo para uma análise de confiabilidade fidedigna.

Apesar do potencial de informação da FCDD, na prática o seu uso apresenta algumas dificuldades provenientes do seu preenchimento e da arquitetura do sistema. A falta de confiança nos dados exige do analista que avalie, uma a uma, cada discrepância descrita e julgue se se trata de fato de uma falha ou não, o que aumenta em grande escala o tempo dedicado nos estudos. No contexto das sete dimensões da qualidade da informação (e que impactam na qualidade das decisões) definidos por Laudon e Laudon (2021), alguns dos dados puros das FCDDs não possuem Acurácia, isto é, não representam a realidade de algumas ocorrências que não configuram falhas.

Sabendo-se que há padrões recorrentes em FCDDs não validáveis, como já ilustrado em Vieira (2016), Martins e Barbosa Filho (2016) e Silva (2018), seria de grande benefício se a classificação destes dados pudesse ser realizada de forma automatizada, com uso de *machine learning*. Nesse sentido, o intuito deste trabalho é levantar técnicas de *machine learning* aplicáveis à classificação dos textos descritivos das possíveis falhas e analisar suas performances na filtragem desses dados.

Esse propósito será alcançado inicialmente por meio de uma revisão bibliográfica acerca da temática de modelagem em *machine learning* e, posteriormente, por um estudo de caso sobre os dados de falha das aeronaves T-27 na Força Aérea Brasileira. Para a construção do modelo no estudo de caso, será trabalhada a classificação individual de cada entrada do conjunto de dados, o que, por si só, já traz benefícios em relação ao conhecimento das características desse instrumento.

2 CONCEITOS DE MACHINE LEARNING

Na atual era da informação, onde se torna cada vez mais viável o armazenamento de grande volume de informações, novas técnicas e recursos são continuamente criados e adaptados de forma a se extrair conhecimento desses dados. Neste cenário, Provost e Fawcett (2013) se referem a *Big Data* como conjuntos de dados que são muito grandes para os sistemas de processamento de dados tradicionais e que, portanto, necessitam de novas tecnologias para seu tratamento. Waller e Fawcett (2013), por sua vez, prosseguem no assunto, afirmando que, para aproveitar a revolução do *Big Data*, pesquisadores e gestores de cadeias de suprimentos precisam entender e abraçar o papel das técnicas relacionadas e suas implicações na tomada de decisão, já que os dados são amplamente considerados *drivers* de melhores decisões e lucratividade.

Um termo tão popular quanto *Big Data* é Inteligência Artificial (IA). Dhankar e Walia (2020) o resumem como uma área da ciência da computação pela qual se criam sistemas inteligentes que podem processar informações ou até mesmo agir como os humanos. Adicionalmente, esses autores definem *Machine Learning* como uma subárea da Inteligência Artificial que permite aos sistemas que aprendam por si mesmos, com base em dados a ele oferecidos, dos quais o próprio sistema extrai padrões. Sendo assim, por IA podem ser entendidos desde sistemas muito simples programados apenas com *ifs* e *elses*, até modelos de *Machine Learning* de reconhecimento de imagens que exigem milhões de imagens pré-classificadas como entrada.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

Segundo Xu e Kumar (2015), textos não estruturados compõem cerca de 75% das informações em uma organização, sendo, portanto, a mineração desses textos uma importante solução para aplicações comerciais e industriais. Apesar dos computadores não interpretarem dados textuais da mesma forma que os humanos, aplicações que extraem informações desses conteúdos já são vistas cotidianamente, a saber, em:

- a) Corretores ortográficos nos computadores e smartphones;
- b) Categorização de conteúdo, como agrupamento de notícias sobre determinado assunto;
- c) Análise de sentimento em redes sociais;
- d) Conversão de fala em texto;
- e) Ferramentas de tradução de textos;
- f) *Chatbots*, ou seja, os atendentes eletrônicos automatizados; e
- g) Detecção de *spam* e fraudes nas caixas de e-mail

As técnicas mais difundidas de processamento de texto para modelagem são as que transformam os textos em matrizes numéricas após a aplicação de diferentes transformações, que podem ser utilizadas individualmente em uma entrada ou adaptados para todo um conjunto delas, sendo algumas:

- a) *Stemming*: transformação da palavra em seu radical, ignorando conjugações e desinências, focando na semântica básica da palavra. Por exemplo, as palavras “vencimento”, “vencido”, “vencida” e “venceu” são todas, com esta técnica, transformadas em “venc”;
- b) *Vectorization*: transformação de palavras em vetores numéricos mantendo relações de similaridade relativas. Por exemplo, a relação entre vetores relativos às palavras Brasil e Brasília seria a mesma relação entre os vetores Espanha e Madri, cujas comparações se associam a país-capital. Geralmente são utilizados valores já pré-treinados em grandes volumes de texto, que são aplicados nos textos analisados;
- c) Remoção de *stopwords*: é a desconsideração, dentro do texto, de palavras que não acrescentam valor semântico ao seu conteúdo, como artigos (o, a, um etc.) e preposições (de, com, por etc.);
- d) *Count Vectorizer*: se trata da construção de vetores numéricos a partir de texto, simplesmente indicando o número de vezes que a palavra aparece no texto. Assim, o conjunto de dados exemplo do Quadro 1 seria transformado no Quadro 2.

Quadro 1 - Exemplos de descrições de FCDDs.

Entrada 1	Apresenta freio esquerdo travado / freio direito ineficiente
Entrada 2	Apresenta freio direito ineficiente
Entrada 3	Apresenta pneu direito furado

Fonte: O próprio autor (2021).

Quadro 2 - CountVectorizer das entradas do Quadro 1.

	<i>apresenta</i>	<i>freio</i>	<i>esquerdo</i>	<i>travado</i>	<i>direito</i>	<i>ineficiente</i>	<i>pneu</i>	<i>furado</i>
Entrada 1	1	2	1	1	1	1	0	0
Entrada 2	1	1	0	0	1	1	0	0
Entrada 3	1	0	0	0	1	0	1	1

Fonte: O próprio autor (2021).

e) *TF-IDF (Term Frequency – Inverse Document Frequency)*: Representa a multiplicação entre o TF (*Term Frequency*) pelo IDF (*Inverse Document Frequency*). O TF é a divisão do número de vezes que uma palavra aparece numa entrada dividido pelo número de palavras totais nessa mesma entrada. Já o IDF é o logaritmo da razão entre o número total de entradas pelo número de entradas em que a palavra aparece. O objetivo dessa transformação é diminuir a interferência de palavras muito comuns em todas as entradas e aumentar a importância de palavras muito comuns dentro de uma única entrada. Por exemplo, num conjunto de dados aeronáuticos, a palavra “voo” pode ser muito frequente em todas as entradas e não ser relevante na extração de informações, mas a frequência da palavra “vazamento” em poucas dessas entradas provavelmente é de interesse de um usuário que pesquisa sobre o tipo de falha. O TF-IDF tem por objetivo ressaltar essa relação de frequências. O Quadro 1 seria transformado, com o TF-IDF, no Quadro 3 a seguir.

Quadro 3 - TF-IDF das entradas do Quadro 1.

	<i>apresenta</i>	<i>freio</i>	<i>esquerdo</i>	<i>travado</i>	<i>direito</i>	<i>ineficiente</i>	<i>pneu</i>	<i>furado</i>
Entrada 1	0	0,05	0,07	0,07	0	0,03	0	0
Entrada 2	0	0,04	0	0	0	0,04	0	0
Entrada 3	0	0	0	0	0	0	0,12	0,12

Fonte: O próprio autor (2021).

É de se observar que os valores para as palavras “apresenta” e “direito” foram todos zerados, uma vez que a presença delas em todos os itens faz com que não indiquem destaque interpretativo. Já “pneu” e “furado”, por aparecerem em apenas uma entrada, têm seus valores aumentados na Linha 3 em relação às palavras que aparecem em mais de uma entrada.

2.2 MODELOS E MÉTRICAS DE CLASSIFICAÇÃO EM MACHINE LEARNING

As estratégias de predição em *Machine Learning* se baseiam em encontrar uma função que se aproxime dos valores alvo de seu conjunto de dados em função de suas variáveis. O processo de aproximação é basicamente um problema de otimização, eventualmente sujeito a restrições, cuja função é a soma de resíduos quadráticos ou outra que se faça conveniente. Tradicionalmente, esse problema de otimização é realizado sobre um conjunto de *treino*, com seus resultados avaliados sobre um conjunto de *validação/teste*, conforme será detalhado à frente.

Particularmente nos casos do Aprendizado Supervisionado, isto é, aquele que possui uma variável de saída rotulada que guia o processo (HASTIE, TIBSHIRANI e FRIEDMAN, 2009) há uma grande variedade de modelos que variam em complexidade, que vão de modelos lineares e lineares generalizados, passando por modelos baseados em árvores de decisão e chegando a modelos como redes neurais, que mimetizam o processo biológico de sinapse dos neurônios. Estes modelos incluem técnicas como Regressão Logística (RL), *Support Vector Classifier* (SVC), *Naive Bayes Multinomial Classifier* (MultinomialNB), *Random Forest Classifier* (RFC), implementados e discutidos em Pedregosa *et al* (2011) e LightGBM, por Ke *et al* (2017). Eventualmente, tais modelos são combinados em um *ensemble*, que configura uma média da saída dos modelos ou uma “votação” entre eles.

Por outro lado, a escolha das métricas de avaliação nos treinamentos de modelo de machine learning tem papel relevante na eficiência dos modelos quando colocados em uso e é quase uma questão primária da pesquisa (BEKKAR, DJEMAA e ALITOUICHE, 2013). Essas métricas são responsáveis pelo guiamento do ajuste de parâmetros do modelo no problema de otimização do treinamento e na comparação da performance de diferentes modelos.

Nos problemas de classificação, os classificadores são costumeiramente analisados inicialmente pela sua *matriz de confusão* (BEKKAR, DJEMAA e ALITOUICHE, op. cit.) sobre os dados de teste/validação. No caso da classificação binária, ou seja, onde existem apenas duas classes a serem atribuídas a uma entrada, trata-se de uma matriz 2x2, que cruza as classes reais das entradas às classes as quais elas foram atribuídas, conforme Quadro 4, indicando em cada quadrante o número de seus elementos.

Quadro 4 - Matriz de Confusão para problema de classificação de duas classes.

		Classe Prevista	
		Positiva	Negativa
Classe Real	Positiva	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
	Negativa	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Fonte: O próprio autor (2021).

Naturalmente, o objetivo de um bom modelo é que haja grandes valores em TP e TN, ou seja, corretamente atribuídos à sua real classe. A métrica mais imediata para avaliar essa adequação do modelo é a acurácia, que é simplesmente a fração de amostras corretamente classificadas, conforme indicado na Equação 1.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Apesar da simplicidade, a acurácia nem sempre é uma boa métrica. De acordo com Bekkar, Djemaa e Alitouche (op. cit.), classes desbalanceadas causam uma leitura equivocada das métricas comuns, o que pode levar a uma classificação viesada.

Os autores trazem o seguinte exemplo: considere um caso de detecção de fraudes onde apenas 1% dos exemplos sejam fraudulentos e 99% sejam confiáveis. Um modelo simples que classifica todos os exemplos como confiáveis sempre terá uma acurácia de 99% - uma métrica aparentemente ótima, mas que não traz avanço nenhum para sua aplicação, já que não filtra nenhuma entrada fraudulenta.

Uma das métricas sugeridas pelos autores para contornar este problema é o *F1-Score* (ou *F-measure*), que se trata da média harmônica entre *Recall* ($\frac{TP}{TP+FN}$) e *Precision* ($\frac{TP}{TP+FP}$). Com essa configuração, garante-se que o valor da métrica só será alto se o modelo performar bem sobre a classe menos prevalente no conjunto de dados. O cálculo dessa métrica é apresentado na Equação 2. No caso do exemplo acima, o F1-score do modelo ilustrado seria de apenas 0,02.

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \quad (2)$$

2.3 O PROBLEMA DO *OVERFITTING*

Segundo Hastie, Tibshirani e Friedman (2009), a generalização da performance de um modelo se relaciona à sua capacidade de predição em um conjunto de dados de teste independente, sendo a avaliação desta performance de extrema importância, já que guia a escolha do modelo e nos dá uma medida da qualidade do modelo selecionado.

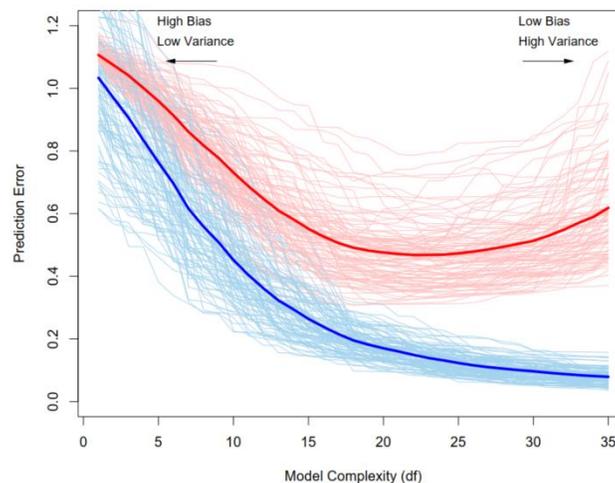
Neste sentido, segundo os autores, o método mais indicado (num contexto de abundância de dados) de divisão do conjunto de dados para treinamento do modelo é dividir esse conjunto em

três: o de treino, o de validação e o teste. Sobre o conjunto de treino são otimizados os parâmetros do modelo, sobre o conjunto de validação são testados os erros de predição de diferentes modelos para seleção do melhor e sobre o conjunto de teste, avalia-se o erro geral do modelo selecionado. Não há uma regra geral para o tamanho destes conjuntos, mas uma divisão comum é de 50%, 25% e 25% para os conjuntos de treino, validação e teste, respectivamente. Essa divisão é necessária, pois, se fosse utilizado o erro sobre o conjunto de treino como métrica, haveria uma subestimação de seu valor. O contexto de poucos dados pode exigir outras abordagens, uma delas sendo utilizar apenas os conjuntos de treino e de validação. Os métodos baseados nessa divisão são comumente chamados de *Cross-Validation* e constituem o padrão dos métodos de treinamentos em *machine learning*.

Ainda de acordo com Hastie, Tibshirani e Friedman (op.cit.), uma das opções quando não se possui dados numerosos é o *K-Fold Cross Validation*. Trata-se de dividir o conjunto de dados em K partes (a princípio, de igual tamanho). Para a K-ésima parte, selecionada temporariamente como conjunto de validação, treina-se o modelo sobre as K-1 partes restantes, obtendo-se a métrica escolhida do erro sobre a validação. Fazendo isso para as K partes, finalmente combinamos seus K resultados, geralmente calculando a média. Os autores colocam 5 e 10 como bons valores de K, conforme pesquisas bibliográficas de seu feito.

Os métodos descritos acima têm objetivo de evitar a superestimação da performance dos modelos por pura aleatoriedade ou por sobreajuste, ou *overfitting*, dos modelos. Para entender esse problema, a Figura 1 mostra um gráfico teórico de erros sobre os conjuntos de treino e teste, variando conforme a crescente complexidade (capacidade de ajuste) do modelo. A curva em azul se refere ao erro sobre o conjunto de treino e a curva em vermelho, sobre o conjunto de validação/teste. Percebe-se que o erro sobre o conjunto de treino diminui constantemente, enquanto o erro sobre o conjunto de teste, independente, diminui até um valor ótimo, a partir de onde cresce. Mas o primeiro é consistentemente menor que o segundo, já que se trata de uma amostra “independente”, que não influenciou na otimização de parâmetros do modelo.

Figura 1 - Gráfico usual de erro de treinamento sobre conjunto de treino e conjunto de validação/teste para diferentes complexidades do modelo.

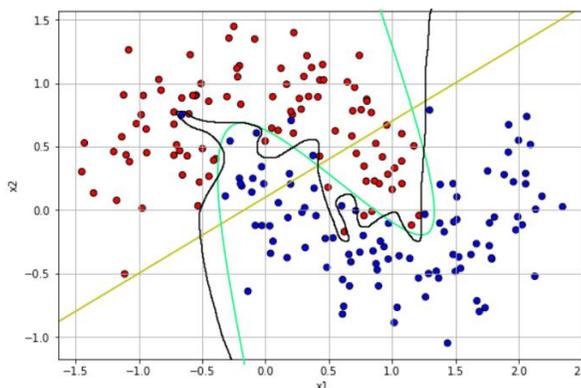


Fonte: (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Ainda nesse sentido, cada modelo possui diferentes formas de ajuste de sua complexidade, isto é, sua capacidade de se ajustar aos dados de entrada. Por exemplo, um modelo polinomial possui maior complexidade conforme aumenta-se seu grau pois aumenta o número de parâmetros, assim como um modelo de *Random Forests* aumenta sua complexidade quando se aumenta seu número de componentes (árvores de decisão). Essa maior capacidade de ajuste poderia parecer benéfica, mas, a partir de certa complexidade, o modelo começa a se ajustar inclusive ao ruído do conjunto de treino, diminuindo o erro sobre ele, mas aumentando o erro sobre o conjunto de validação/teste, tornando o modelo cada vez menos generalizável, o que também pode ser visualizado na Figura 1.

Já na Figura 2 é exemplificado um problema de classificação fictício com 2 variáveis de entrada, x_1 e x_2 , com classes de resposta “azul” e “vermelho”. Em amarelo, vê-se uma curva em *underfit*, ou seja, com subajuste em relação à fronteira das classes. Em verde, há uma fronteira com bom ajuste, com um certo erro proveniente de um ruído natural que não se pode remover. Já em preto, há uma fronteira de um modelo com *overfit*, ou seja, que se ajusta a entradas de dados onde há grande influência de ruído. Esse modelo terá um desempenho deficiente em novas amostras que sejam por ele classificadas quando comparado ao modelo de cor verde, por exemplo.

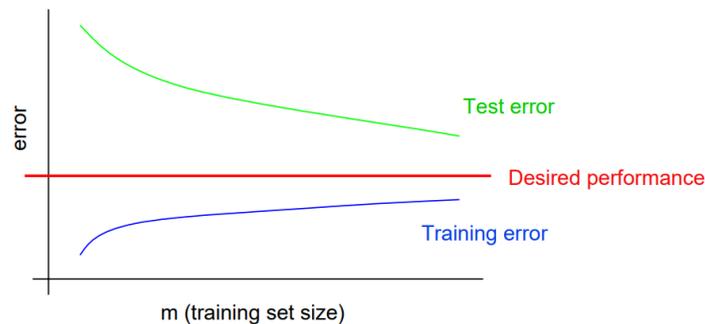
Figura 2 - Exemplo visual de overfitting em problema de classificação.



Fonte: O próprio autor (2021).

Para a prevenção do problema do *overfitting*, além das técnicas de *Cross Validation* já discutidas, Ng (2011) sugere aquisição de maior número de dados, diminuição do número de parâmetros/variáveis do modelo e regularização. Para a avaliação do *overfitting*, o mesmo autor sugere a observação da *learning curve*, que é a avaliação do desempenho do modelo (no conjunto de dados de treino e de teste) em função do tamanho da amostra utilizada como conjunto de dados de treino. Se houver grande diferença nos desempenhos, é provável que haja *overfitting*, mas não há regra geral para esta diferença. Um modelo com *overfitting* provavelmente terá a *learning curve* no formato da Figura 3, com um gap de desempenho entre os desempenhos de treino e teste. Um modelo mais adequado terá as curvas de erro de treino e de teste assintoticamente próximas.

Figura 3 - Comportamento típico de uma *learning curve* para modelo com *overfitting*, com gap entre resultados nos conjuntos de treino e teste.



Fonte: (NG, 2011).

Cabe ressaltar, entretanto, que nem sempre é possível evitar o *overfitting*, em especial em contextos com dados insuficientes. Ainda assim, modelos com bons resultados calculados no conjunto de validação terão resultados úteis.

3 MATERIAL E MÉTODO

Para alcançar o objetivo proposto neste trabalho de analisar a performance de aplicações de *machine learning* na validação de FCDDs, será lançada mão de revisão bibliográfica acerca da temática, já discutida na seção anterior deste estudo, seguida de um estudo de caso sobre os dados de falha das aeronaves T-27 na Força Aérea Brasileira (FAB). Sendo assim, este trabalho configura uma pesquisa aplicada, quantitativa, descritiva, bibliográfica e de estudo de caso.

Sobre a revisão de literatura, depreendeu-se que os modelos Regressão Logística, *Support Vector Classifier* (SVC), *Naive Bayes Multinomial Classifier* (MultinomialNB), *Random Forest Classifier* (RFC) e *LightGBM* são aplicáveis ao estudo realizado. Entende-se, também, que uma combinação destes modelos em um *ensemble* pode trazer um desempenho ainda melhor. Caso se perceba que o conjunto de dados analisado é desbalanceado, será utilizada uma métrica derivada do F1-score.

Já quanto ao estudo de caso, serão classificadas manualmente um grande número de FCDDs da aeronave T-27 consultadas no SILOMS, classificando cada uma como válida como falha ou não válida como falha. Esses dados serão as entradas para os modelos de classificação, que serão implementados em linguagem Python. Por fim, os resultados dos modelos terão seus desempenhos comparados com as técnicas de filtragem atualmente utilizadas e serão analisados o impacto da utilização do modelo sobre os indicadores de confiabilidade utilizados atualmente.

Os detalhamentos da seleção dos dados analisados, sua pré-limpeza, tratamentos nos textos, melhoria dos modelos, seleção do *ensemble* e seleção da métrica são discutidos nas subseções a seguir.

3.1 AQUISIÇÃO E LIMPEZA DE DADOS

Os dados de falha foram recolhidos do SILOMS e passaram por uma limpeza inicial no software Microsoft Excel. Foram selecionados para as análises os anos de 2018 e 2019. A DIRMAB (Diretoria de Material Aeronáutico e Bélico) já possuía as FCDDs analisadas para o ano de 2018 para um trabalho interno. Sendo assim, optou-se por adicionar os dados relativos a 2019

para assegurar uma maior conformidade dos dados dos dois anos e garantir um maior conjunto de dados para fortalecer o modelo construído.

Desses dados, foram removidas as seguintes entradas:

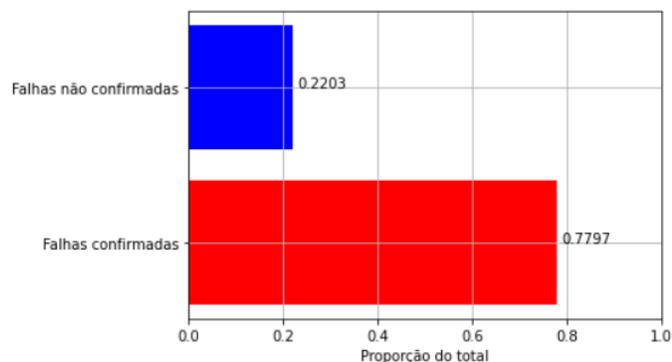
- Entradas duplicadas, como as provenientes de entrada única de PNs que possuem mais de um tipo de controle (horas de voo, ciclos, partidas, etc.);
- Entradas duplicadas de descrição distinta de um mesmo SN de um mesmo PN, no mesmo dia já que, provavelmente, tratavam da mesma ocorrência. Foi mantida a primeira entrada;
- Entradas de descrição vazia, que não podem ser interpretadas quando à sua validade.

Após a aplicação deste filtro, restaram 7299 entradas de dados, sendo 3907 referentes ao ano de 2018 e 3392 referentes ao ano de 2019.

Os dados remanescentes desse filtro tiveram sua validade, ou não, julgada inicialmente pelo campo “DISCREPÂNCIA”. Foi criada uma coluna “Validação”, que associava o valor “Sim” às entradas consideradas, de fato, provenientes de falhas, e “Não” às que se relacionavam a outras ocorrências. Quando o campo “DISCREPÂNCIA” se fazia insuficiente na análise, outros campos eram analisados, como “COMENTÁRIO DA OFICINA”, “WDC” (*When Discovered Code*), etc. Cabe ressaltar que essa avaliação possui inevitavelmente um cunho subjetivo, já que se baseia em uma informação limitada e, portanto, pode variar de acordo com os conhecimentos e experiências do analista. Entretanto, este método é o mesmo que se utilizaria num grande trabalho relacionado à confiabilidade, já que a investigação de todas as entradas de dados muitas vezes se faz inviável, como já foi discutido.

Realizada a classificação, foi verificado que cerca de 78% das entradas correspondiam, de fato, a falhas, e 22%, a entradas de ruído que respondiam a outros tipos de ocorrências, como é representado graficamente na Figura 4. É um nível considerável de dados não confiáveis a serem detectados.

Figura 4 - Proporção de dados de falhas validados e não validados do Projeto T-27 para os anos de 2018 e 2019.



Fonte: O próprio autor (2021).

3.2 TRATAMENTO DE TEXTO

No escopo deste trabalho, como ponto de partida, a previsão é feita somente sobre o campo “DISCREPÂNCIA” do conjunto de dados, já que este é, em geral, o campo mais expressivo na avaliação da validade da FCDD.

A partir deste ponto, a manipulação dos dados textuais e a consequente modelagem foram realizadas na linguagem Python, versão 3.7.10.

Para a transformação dos dados textuais, não estruturados, em dados numéricos, os seguintes passos são tomados, na ordem enunciada, a saber:

- 1) Números seguidos de letras são substituídos pelo termo “*measure*”, já que, da observação, a grande maioria desses casos se referem a medidas, cujo valor numérico *per se* não traz significado semântico de falha. Acreditou-se que a presença de uma medida na descrição poderia ser fator importante na avaliação da FCDD, e por isso a indicação de sua presença seria importante. Além disso, “*measure*” é um termo que não estava presente em outras entradas, não gerando confusão com termos rotineiramente utilizados;
- 2) Palavras que misturavam letras e números foram substituídas pelo termo genérico “SNPNPUB”, que indica a presença de escrita de SN, PN ou Publicação Técnica. O ideal seria a diferenciação desses três elementos, mas não se identificou padrão que pudesse separá-los. Similar ao item anterior, o código individual desses SN, PN ou Publicação Técnica em si também não traz valor semântico, mas sua presença pode trazer;
- 3) Foram removidos caracteres especiais como “%”, “\$”, etc., e todas os caracteres foram transformados em caracteres minúsculos;
- 4) Foi aplicado um *Stemming* termo a termo, de forma a reduzir as palavras aos seus radicais. O *stemmer* utilizado foi o da biblioteca NLTK da linguagem Python;
- 5) Finalmente, os textos são transformados em colunas numéricas pelo método TF-IDF. São aproveitadas no TF-IDF apenas os radicais que tenham aparecido ao menos 4 vezes, de modo a evitar que o modelo se ajuste a palavras pouco frequentes. Além disso, não são contabilizadas no modelo as *stopwords*, sendo usada a relação da língua portuguesa disponibilizada também pela biblioteca NLTK.

Cabe ressaltar que as etapas 1 a 4 acima foram aplicadas indiscriminadamente em todos os dados, mas o TF-IDF da etapa 5 é ajustado ao conjunto de treino e, em seguida, usado para transformar o conjunto de validação. Essa separação de ajuste e transformação é necessária para que os modelos não “vejam” os dados de validação, que são considerados independentes antes de serem treinados. Esse processo de separação de teste e validação é clarificado na subseção seguinte.

3.3 SELEÇÃO DE MODELOS, MÉTRICAS, PARÂMETROS DE *TUNING* E *CROSS-VALIDATION*

Para avaliação, foram selecionados os modelos Regressão Logística (RL), *Support Vector Classifier* (SVC), *Naive Bayes Multinomial Classifier* (MultinomialNB), *Random Forest Classifier* (RFC) e LightGBM. Todos estes modelos possuem estruturas diferentes de classificação, com exceção do RFC e do LightGBM que são ambos baseados em árvores de decisão. Optou-se por checar se o LightGBM traz, no presente caso, melhorias em relação ao RFC, por ser um modelo muito popular na comunidade de *machine learning*. Os modelos são todos implementados em linguagem Python 3.7.10, com auxílio das bibliotecas Scikit-Learn e lgbm.

Cada um destes modelos possui hiperparâmetros a serem selecionados, processo este chamado de *tuning*. Para cada modelo, estes hiperparâmetros foram selecionados de modo a permitir o maior valor da métrica selecionada. Serão testados, para cada um dos modelos:

- 1) Regressão Logística: C (inverso do fator de regularização) entre os valores 0,01; 0,1; 1; 10; 100;
- 2) SVC: C (inverso do fator de regularização) entre os valores 0,01; 0,1; 1; 10; 100;

- 3) MultinomialNB: *alpha* (parâmetro de suavização) entre os valores 0,01; 0,1; 1; 10; 100;
- 4) RFC: *n_estimators* (número de estimadores, ou seja, número de árvores de decisão) entre os valores 10, 25, 50, 75, 100, 200 e *max_depth* (máxima profundidade das árvores de decisão) entre os valores 10, 20, 30;
- 5) LightGBM: *n_estimators* (número de estimadores, ou seja, número de árvores de decisão) entre os valores 10, 50, 100, 200.

Foi analisado se o uso de *ensembles* de estimadores colaboraria com a melhoria da performance, escolhendo dentre os *ensembles* com os *n* melhores modelos obtidos, com *n* de 1 a 5.

A métrica selecionada para a comparação dos modelos será a F1-macro, uma variação do F1-score. Consiste em calcular a métrica F1-score considerando alternadamente as falhas confirmadas e não confirmadas como a classe positiva (classe referência para o cálculo de *Precision* e *Recall*, conforme Quadro 4) e computar a média aritmética entre os dois valores. Dessa forma:

- a) com o uso de uma métrica derivada da F1-score, evita-se que os modelos tenham viés em prever a classe mais presente (Bekkar, Djemaa e Alitouche, 2013), no caso, falhas confirmadas; e
- b) com um resultado derivado da métrica F1-score calculado sobre as duas classes possíveis, priorizam-se igualmente as duas classes (falha validada e falha não validada) quanto ao diagnóstico do modelo.

Por fim, tanto para a seleção de hiperparâmetros quanto para a seleção do modelo final, foi utilizado um *K-Fold Cross Validation* com 5 *Folds*, conforme números de divisões sugeridos por Hastie, Tibshirani e Friedman (2009).

3.3 BENCHMARKING COM TÉCNICA ATUAL

É interessante comparar o método aqui desenvolvido com a aplicação de abordagens tradicionais, que se resumem a verificar se alguns termos comuns em FCDDs não validadas estão presente em cada entrada. Se uma das palavras desse filtro estiver presente na entrada, considera-se esta FCDD não confirmada como falha; caso contrário, mantém-se a entrada como uma falha validada. Este modelo será chamado aqui por “Filtro Binário”. Chaves (2020) utiliza como filtro os termos “TBO”, “REVISAO GERAL”, “REV GERAL”, “INSPEÇÃO CRÍTICA”, “OVERHAUL”, “HSI”, “T.B.O.”, “TLV” e “PEDIDO”. Além destes termos, foram adicionados “VENCÍ”, “INSPE” e “CANIB”. Será calculado o desempenho desse modelo no banco de dados validado sob a mesma métrica F1-macro de forma a poder ser visualizada a comparação com o modelo selecionado.

4 RESULTADO E DISCUSSÃO

4.1 SELEÇÃO E ANÁLISE DO MODELO

Após a aplicação dos treinamentos dos modelos, foram selecionados pelo tuning os modelos apresentados na Tabela 1, com os respectivos hiperparâmetros, do melhor desempenho para o pior. A melhor seleção de componentes do *ensemble* foi a combinação dos modelos SVC, RL e MultinomialNB da Tabela 1 aqui chamado de “Voting”. Esse estimador retorna a média das

probabilidades estimadas de confirmação da falha para cada um dos 3 modelos, trazendo uma melhoria pouco relevante, também apresentada na Tabela 1.

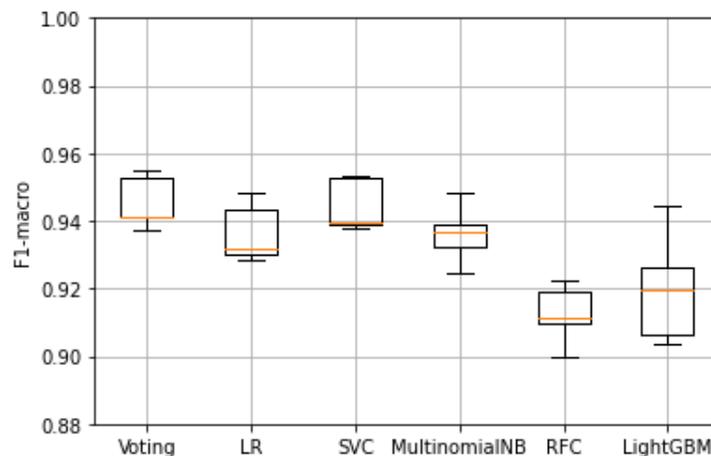
Tabela 1 - Resultados dos melhores conjuntos de hiperparâmetros por modelo em relação à métrica F1-macro.

Modelo	Hiperparâmetros	Média do F1-macro no CV-5
Voting-RL-SVC-MNB	-	0,9454
SVC	C=100	0,9443
RL	C=10	0,9364
MultinomialNB	alpha=0,1	0,9361
LightGBM	n_estimators=200	0,9200
RFC	n_estimators=75, max_depth=30	0,9125

Fonte: O próprio autor (2021).

Uma forma melhor de comparação é a visualização dos boxplots dos F1-macro retornados em cada Fold para cada modelo acima. Isto é feito na Figura 5. Dos modelos simples, percebe-se que, de modo geral, o RFC e LightGBM performam um pouco pior que os demais. Percebe-se também que não há grande diferença nos resultados de RL e SVC, bem como há uma melhoria muito sutil ao se utilizar o modelo Voting. Ainda assim, será este o modelo selecionado para as análises seguintes. O desempenho próximo de vários dos modelos não surpreende, já que possuir boa quantidade de dados de boa qualidade é mais importante que a arquitetura do modelo utilizado, no geral. Em particular, a Acurácia no cross-validation para o modelo Voting foi de 96,33%.

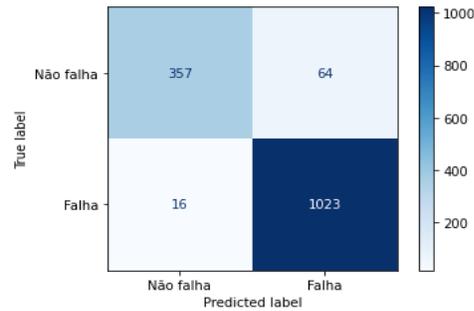
Figura 5 - Boxplots dos valores de F1-macro calculados para cada modelo.



Fonte: O próprio autor (2021).

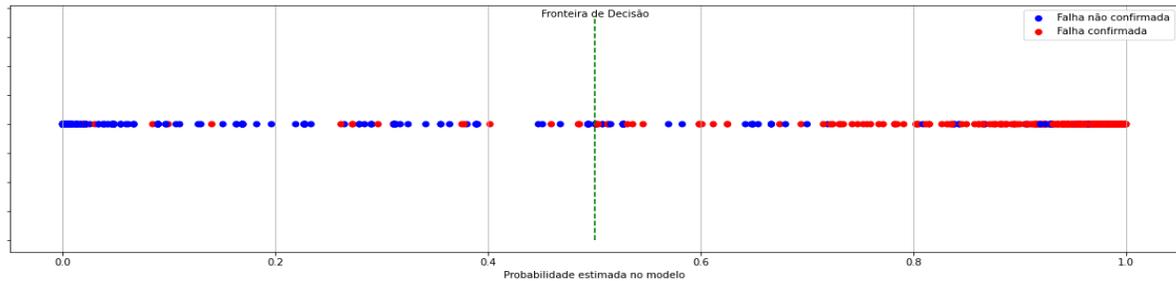
Em relação especificamente ao modelo selecionado, Voting, sua matriz de confusão (utilizando o último quinto do conjunto de dados como dados de validação) é apresentada no Quadro 5. Já a Figura 6 mostra a relação entre as probabilidades previstas para cada entrada de dado da validação e a sua real confirmação quanto a se tratar de falha ou não.

Quadro 5 - Matriz de Confusão para o classificador Voting.



Fonte: O próprio autor (2021).

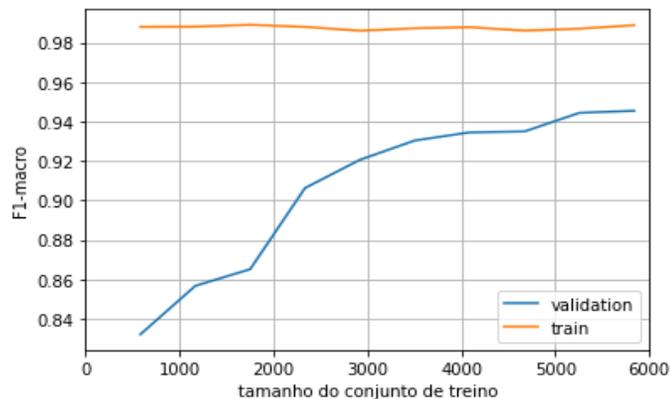
Figura 6 - Visualização das probabilidades previstas e o rótulo real de cada dado de validação para o classificador Voting.



Fonte: O próprio autor (2021).

Em relação ao *overfitting*, vemos o comportamento da *learning curve* na Figura 7 abaixo. Comparativamente ao comportamento da Figura 3, os desempenhos nos conjuntos de treino e validação têm comportamento invertido, já que a métrica F1-macro é tão melhor quanto maior, enquanto o “erro” da Figura 3 é tão pior quanto maior. Percebe-se que existe um gap razoável entre as curvas no ponto de maior tamanho do conjunto de treino. Entretanto, o comportamento crescente da curva relativa à validação indica que o modelo deve melhorar com aporte de maior conjunto de dados. De todo modo, o alto desempenho sobre o conjunto de validação aponta que o modelo possui utilidade.

Figura 7 - Learning Curve para o classificador Voting.



Fonte: O próprio autor (2021).

É interessante também avaliar os radicais que mais auxiliam no desempenho do modelo, ou seja, os mais importantes. Para tal, utilizou-se da métrica *Permutation Feature Importance*, que retreina o modelo várias vezes para permutações aleatórias de cada variável. Dessa forma, a relação da variável com o rótulo de classificação se perde e se mensura o impacto na métrica estudada (Breiman, 2001). Os 15 radicais mais importantes sobre o conjunto de validação (dentro os 976 gerados no treinamento), junto aos seus impactos no F1-macro, são mostrados na Tabela 2. São mostrados também os termos mais comuns com os radicais exibidos. Destaca-se que a importância calculada é uma medida do impacto da existência da variável na métrica escolhida, e não necessariamente indica diretamente aumento ou diminuição da probabilidade calculada.

Tabela 2 - Radicais mais importantes no classificador Voting em relação à métrica F1-macro no conjunto de validação, sua *Permutation Importance* e os principais termos que contêm esses radicais.

Radical	PI	Principais termos
geral	0.1085	geral
tbo	0.0884	tbo
venc	0.0288	vencido(a)(s), vencimento
test	0.0144	test, teste, testado(a)
estoqu	0.0123	estoque
retorn	0.0044	retorno (de áudio, à posição), retornado da empresa
mater	0.0039	material (pedido)
empr	0.0038	empresa
rep	0.0035	reparo, reparado(a)
anv	0.0032	anv (comum em canibalizações)
inoper	0.0032	inoperante
retir	0.0029	retirado(a), retirar
vaz	0.0028	vazamento
ped	0.0027	pedido (de material)
measur	0.0023	measure (medidas em qualquer unidade)

Fonte: O próprio autor (2021).

4.2 BENCHMARKING COM TÉCNICAS APLICADAS ATUALMENTE

A comparação entre os desempenhos do modelo Voting e do modelo Filtro Binário, adaptado de Chaves (2020), é ilustrada na Tabela 3 abaixo.

Tabela 3 - Comparação de desempenho entre os modelos Voting e Binário.

Métrica	Voting	Filtro Binário
F1-macro	0,9454	0,7623
Acurácia	96,33%	86,31%

Fonte: O próprio autor (2021).

Portanto, percebe-se que o classificador gerado neste trabalho traz benefícios em relação ao método utilizado atualmente. Provavelmente, isso se deve à interação de palavras, que não é capturada pelo método Filtro Binário, e pela não necessidade de identificar visualmente os termos mais importantes para filtragem dos dados, o que é feito manualmente no método atual.

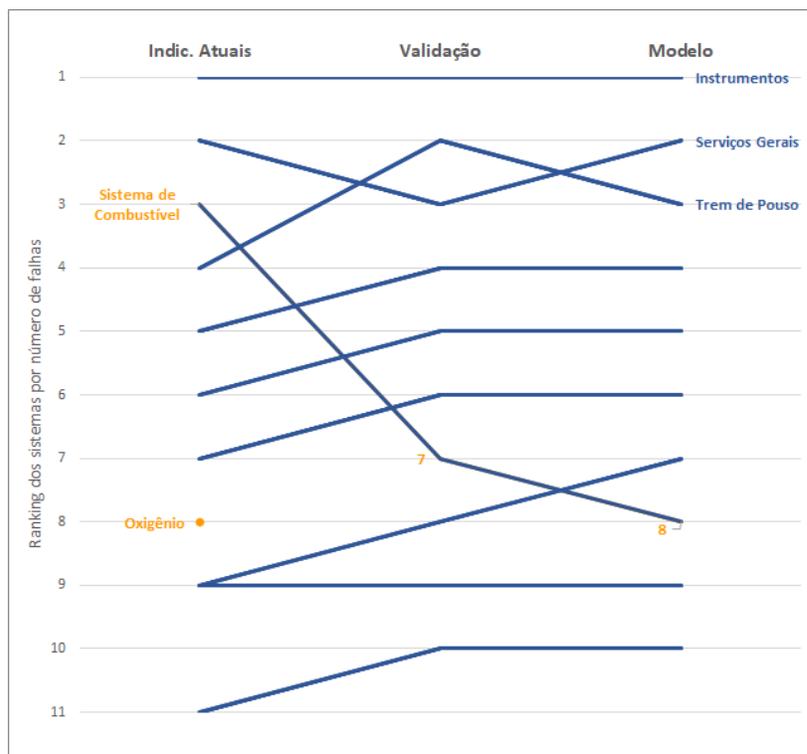
4.3 IMPACTOS EM INDICADORES LOGÍSTICOS

Considerando que o modelo poderia ser utilizado para refinar os dados dos indicadores logísticos de confiabilidade hoje utilizados, a análise de seus benefícios é feita nos slope graphs da

Figura 8 e da Figura 9. Os Quadros com a quantidade de falhas por sistema e WUC para a construção dos rankings são mostradas no Apêndice A. A Figura 8 mostra os rankings de sistemas com o maior número de falhas em 3 cenários: Indicadores Atuais, com entrada de FCDDs sem limpeza dos dados; Validação, que se refere à validação manual realizada sobre os dados; e Modelo, que representa o ranking com os dados refinados pela metodologia aqui apresentada. A Figura se refere aos últimos 20% dos dados (modelo treinado nos 80% iniciais), que temporalmente respondem às FCDDs geradas de 24/07/2019 a 31/12/2019.

Percebe-se que os indicadores atuais superestimam a criticidade de dois sistemas, o de Combustível e o de Oxigênio em relação a validação, enquanto o ranking gerado com refinamento do modelo possui grande fidelidade ao ranking validade, com algumas permutações entre posições adjacentes.

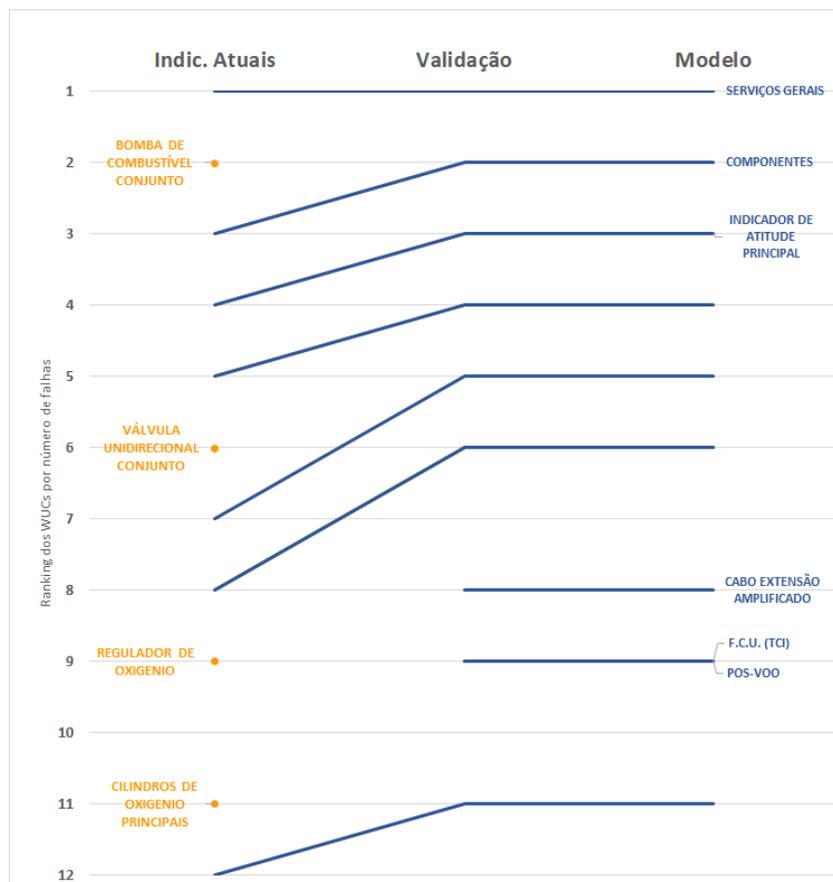
Figura 8 - Ranking de Sistemas com maior número de FCDDs nos indicadores atuais, após validação manual e com uso do modelo proposto.



Fonte: O próprio autor (2021).

Já a Figura 9 mostra a mesma análise para os WUCs (*Work Unit Codes*) do projeto, no mesmo período. Percebe-se que sistemas como Bomba de Combustível, Válvula Unidirecional, Regulador de Oxigênio e Cilindros de Oxigênio Principais aparecem nos indicadores atuais, mas não fazem parte do ranking após validação manual. Por outro lado, o WUC Cabo Extensão Amplificado está no Top 10 dos códigos com mais falhas, mas não aparece nos indicadores atuais. O ranking refinado com o modelo, mais uma vez, possui bons resultados, mantendo as mesmas posições, até o ranking 9, havendo trocas de posições subsequentes a partir da posição 11, apenas.

Figura 9 - Ranking de WUCs com maior número de FCDDs nos indicadores atuais, após validação manual e com uso do modelo proposto.



Fonte: O próprio autor (2021).

5 CONSIDERAÇÕES FINAIS

O objetivo deste estudo foi levantar técnicas de *machine learning* aplicáveis à classificação dos textos descritivos das possíveis falhas e analisar suas performances na filtragem desses dados. Neste âmbito, verificou-se que os modelos de *machine learning* apresentaram boas performances, sendo a melhor a da combinação em *ensemble* dos modelos SVC, Regressão Logística e MultinomialNB, que apresentou score F1-macro de 0,9454 e acurácia de 96,33%. Essa performance é superior à do modelo tradicional, que depende do levantamento humano de palavras importantes para a filtragem. O modelo ainda apresentou boa aplicabilidade na correção dos rankings dos indicadores logísticos de confiabilidade, mantendo praticamente a mesma ordem de sistemas e WUCs quando ordenados por quantidades de falhas.

O processo trouxe ainda conhecimentos sobre a características dos dados de falha das aeronaves T-27, como a porcentagem de dados que não configuram falhas, as palavras mais importantes para o modelo de classificação e os sistemas e WUCs que hoje têm suas criticidades super ou subestimadas em relação ao número de falhas.

Vale dizer que a aplicação de modelos como o proposto não substitui a conscientização do pessoal da FAB em relação ao correto preenchimento dos campos de uma FCDD no SILOMS nem a de se buscar a melhoria da arquitetura do sistema de dados, mas traz um método sistemático da limpeza dos dados já produzidos até hoje. Esse método se faz mais adequado para a composição de dados que exigiriam trabalho extenso de validação, como indicadores e dimensionamento de contratos, e não para estudos específicos como estudos de confiabilidade de um PN único, que dependem de mais fontes de dados.

Percebe-se também que a criação desses modelos exige uma validação de um grande conjunto de dados para treinamento e teste. Nesse sentido, cabe ressaltar que a validação desses dados por uma única pessoa, como foi o caso deste estudo, pode trazer vieses pessoais na classificação das entradas e, por consequência, no treinamento do modelo. Essa é uma limitação da análise realizada, sendo uma validação de dados conduzida por diferentes analistas um contorno ao problema.

Por fim, este estudo dá margem a uma gama de trabalhos futuros envolvendo aplicação de *machine learning* em dados de confiabilidade. A validação de mais dados de falha da aeronave T-27 certamente aprimoraria o modelo aqui desenvolvido, mas ainda o limitaria a esta plataforma. Sendo assim, a validação de dados de outro tipo de aeronave para combinação em um modelo único poderia trazer *insights* da sinergia dos dados de diferentes origens e motivar a aplicação de uma ferramenta única de limpeza de dados a ser utilizada nos indicadores de confiabilidade. Outra possibilidade é a utilização deste tipo de validação de dados para refinamento de previsão de demanda, especialmente buscando identificar a que PN cada entrada de dados se refere, uma vez que muitos têm sua falha cadastrada no PN da aeronave.

REFERÊNCIAS

BEKKAR, Mohamed; DJEMAA, Hassiba K.; ALITOCHE, Taklit A. Evaluation Measures for Models Assessment over Imbalanced Data Sets. **Journal of Information Engineering and Applications**, v. 3, n. 10, p. 27-28, 2013.

BREIMAN, Leo. Random Forests. **Machine Learning**, v.45, n. 1, p. 5-32, 2001.

CHAVES, Pedro L. Mineração de Dados de FCDD por meio de Linguagem Python. *In: ENCONTRO DE CONFIABILIDADE NA AVIAÇÃO*, 2., 2020, Guarulhos. **Seminário...** Guarulhos: ILA, 2020. p.15.

DHANKAR, Minal; WALIA, Nipun. An Introduction to Artificial Intelligence. **Emerging Trends in Big Data, IoT and Cyber Security**. Nova Délhi: Maharaja Srajmal Institute, 2020.

HASTIE Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning**. 2. ed. Stanford, Califórnia: Springer, 2009.

LAUDON Kenneth C.; LAUDON Jane P. **Management Information Systems. Managing the Digital Firm**. 17. ed. Global Edition: Pearson, 2021.

MARTINS, Lucas S; BARBOSA FILHO, Carlos M. Comando da Aeronáutica. **Análise de dados de defeito da frota de T-27**: relatório técnico. Lagoa Santa: Parque de Material Aeronáutico de Lagoa Santa, 2016.

NG, Andrew. Advice for applying machine learning. *In: Machine Learning*. 2011.

PROVOST, Foster; FAWCETT, Tom. Data Science and its relationship to big data and data-driven decision making. **Big data**, v. 1, n. 1, p. 51-59, 2013.

SILVA, Jardel F. **Tópicos de MCC 1**: Curso de Manutenção Centrada na Confiabilidade. Guarulhos: Instituto de Logística da Aeronáutica, 2018.

VIEIRA, Thales A. C. J. **Análise de confiabilidade dos itens da frota de aeronaves de C-105**: relatório técnico. Rio de Janeiro: Parque de Material Aeronáutica do Galeão, 2016.

WALLER, Matthew A.; FAWCETT Stanley E. Data Science predictive analytics, and big data: a revolution that will transform supply chain and management. **Journal of Business Logistics**, n. 34(2), p. 77-84, 2013.

XU, Brian; KUMAR, Sathish A. A text mining classification framework and its experiments using aviation datasets. **Proceedings of the Workshop of information technology and systems**, Dallas. 2015.

APÊNDICE A – Quadros de indicadores de confiabilidade

Quadro A.1 – Ranking de sistemas com maior número de falhas segundo indicadores atuais

Indic. Atuais			
#	Sistema	Qtde.	Rank
51	Instrumentos	271	1
1	Serviços Gerais	188	2
46	Sistema de Combustível	156	3
13	Trem de Pouso	139	4
42	Suprimento de Energia Elétrica	95	5
22	Motor Turbo Hélice	94	6
71	Sistema de Rádio Navegação	70	7
47	Oxigênio	65	8
14	Controles de Voo	61	9
64	Sistema de Interfone e Seletora de Áudio	61	9
62	Sistema de Comunicação VHF	51	11
4	Inspecções Especiais	46	12
11	Estrutura	36	13
41	Ar Condicionado	33	14
44	Sistema de Iluminação	23	15
3	Inspecções Programadas	23	15

Quadro A.2 – Ranking de sistemas com maior número de falhas segundo validação

Validação			
#	Sistema	Qtde	Rank
51	Instrumentos	247	1
13	Trem de Pouso	111	2
1	Serviços Gerais	94	3
42	Suprimento de Energia Elétrica	73	4
22	Motor Turbo Hélice	72	5
71	Sistema de Rádio Navegação	68	6
46	Sistema de Combustível	62	7
64	Sistema de Interfone e Seletora de Áudio	61	8
14	Controles de Voo	59	9
62	Sistema de Comunicação VHF	50	10
11	Estrutura	26	11
3	Inspecções Programadas	23	12
44	Sistema de Iluminação	22	13
41	Ar Condicionado	17	14
47	Oxigênio	15	15
12	Cabine	11	16

Quadro A.3 – Ranking de sistemas com maior número de falhas segundo modelo

Modelo			
#	Sistema	Qtde	Rank
51	Instrumentos	248	1
1	Serviços Gerais	120	2
13	Trem de Pouso	110	3
42	Suprimento de Energia Elétrica	75	4
22	Motor Turbo Hélice	74	5
71	Sistema de Rádio Navegação	69	6
64	Sistema de Interfone e Seletora de Áudio	61	7
46	Sistema de Combustível	60	8
14	Controles de Voo	53	9
62	Sistema de Comunicação VHF	50	10
11	Estrutura	29	11
44	Sistema de Iluminação	23	12
3	Inspecções Programadas	22	13
47	Oxigênio	17	14
41	Ar Condicionado	17	15
4	Inspecções Especiais	17	16

Quadro A.4 – Ranking de WUCs com maior número de falhas segundo indicadores atuais

Indic. Atuais			
#	Sistema	Qtde	Rank
51	Instrumentos	271	1
1	Serviços Gerais	188	2
46	Sistema de Combustível	156	3
13	Trem de Pouso	139	4
42	Suprimento de Energia Elétrica	95	5
22	Motor Turbo Hélice	94	6
71	Sistema de Rádio Navegação	70	7
47	Oxigênio	65	8
14	Controles de Voo	61	9
64	Sistema de Interfone e Seletora de Áudio	61	9
62	Sistema de Comunicação VHF	51	11
4	Inspecções Especiais	46	12
11	Estrutura	36	13
41	Ar Condicionado	33	14
44	Sistema de Iluminação	23	15
3	Inspecções Programadas	23	15

Quadro A.5 – Ranking de WUCs com maior número de falhas segundo validação

Validação			
#	Sistema	Qtde	Rank
51	Instrumentos	247	1
13	Trem de Pouso	111	2
1	Serviços Gerais	94	3
42	Suprimento de Energia Elétrica	73	4
22	Motor Turbo Hélice	72	5
71	Sistema de Rádio Navegação	68	6
46	Sistema de Combustível	62	7
64	Sistema de Interfone e Seletora de Áudio	61	8
14	Controles de Voo	59	9
62	Sistema de Comunicação VHF	50	10
11	Estrutura	26	11
3	Inspecções Programadas	23	12
44	Sistema de Iluminação	22	13
41	Ar Condicionado	17	14
47	Oxigênio	15	15
12	Cabine	11	16

Quadro A.6 – Ranking de WUCs com maior número de falhas segundo modelo

Modelo			
#	Sistema	Qtde	Rank
51	Instrumentos	248	1
1	Serviços Gerais	120	2
13	Trem de Pouso	110	3
42	Suprimento de Energia Elétrica	75	4
22	Motor Turbo Hélice	74	5
71	Sistema de Rádio Navegação	69	6
64	Sistema de Interfone e Seletora de Áudio	61	7
46	Sistema de Combustível	60	8
14	Controles de Voo	53	9
62	Sistema de Comunicação VHF	50	10
11	Estrutura	29	11
44	Sistema de Iluminação	23	12
3	Inspecções Programadas	22	13
47	Oxigênio	17	14
41	Ar Condicionado	17	15
4	Inspecções Especiais	17	16