APRENDIZAGEM DE MÁQUINA APLICADA NA PREVISÃO DE DEMANDA POR SETUP DO PROCESSO DE TRATAMENTO QUÍMICO DE TECIDO NA INDÚSTRIA TÊXTIL

Machine Learning Applied in the Forecast of Demand by Process Setup of Chemical Treatment of Fabric in the Textile Industry

Paulo Sergio Nascimento de Jesus
Orientador: Prof. Drº Oberdan Rocha Pinheiro

RESUMO

O uso cada vez maior do Machine Learning (ML) tem levado muitos processos a dependerem menos da ação humana. Neste artigo foi selecionado o processo de dipagem de tecido em uma indústria têxtil, onde a decisão pela parada da produção, avaliando a necessidade de ajustes nos parâmetros (setup) do equipamento, depende principalmente da ação humana. Trata-se de uma decisão que impacta os indicadores de capacidade produtiva e qualidade do produto. A partir desses dados, buscou-se criar e testar 7 modelos ML com técnicas distintas para decidir sobre a aplicação do setup, avaliando métricas de desempenho. Foram coletados dados históricos, base para o treinamento dos modelos, então seguiu-se por um processo de preparação, organização, estruturação e análise exploratória, produzindo um conjunto de dados que foi dividido em 2 grupos para treino e teste. Na fase de treino ocorreram simulações com diferentes combinações de hiperparâmetros para identificar os melhores estimadores. Por fim, os modelos foram testados com dados desconhecidos e o KNN foi o modelo vencedor, apresentando os melhores resultados.

PALAVRAS-CHAVE: machine learning, dipagem, métricas.

ABSTRACT

The increasing use of Machine Learning (ML) has led many processes to depend less on human action. In this article, the fabric dipping process in a textile industry was selected, where the decision to stop production, evaluating the need for adjustments in the parameters (setup) of the equipment, depends mainly on human action. This is a decision that impacts the indicators of production capacity and product quality. Based on these data, an attempt was made to create and test 7 ML models with different techniques to decide on the application of the setup and evaluate performance metrics. Historical data were collected, the basis for training the models, then followed by a process of preparation, organization, structuring, and exploratory analysis, producing a data set that was divided into 2 groups for training and testing. In the training phase, simulations were performed with different combinations of hyperparameters to identify the best estimators. Finally, the models were tested with unknown data and KNN was the winning model, showing the best results.

KEYWORDS: machine learning, dipping, metrics.

1. INTRODUÇÃO

Os avanços tecnológicos que surgem a cada dia estão provocando um aperfeiçoamento de processos e redução de custos, impactando em menos dependência de avaliação humana. Nesse contexto, surgem tecnologias disruptivas da chamada Indústria 4.0, dentre elas a Aprendizagem de Máquina, que possibilita, através de algoritmos treinados com dados históricos, uma máquina executar tarefas que até então dependiam de decisão humana. Segundo Paixão *et. al*. (2022) “O aprendizado de máquina – do inglês, *Machine Learning* (ML) – é o ramo da inteligência artificial (IA) que explora o estudo e a construção de algoritmos computacionais a partir do aprendizado por dados [...]”. Para Ignacio (2021) “É importante ressaltar que o estudo do aprendizado de máquina envolve diferentes áreas do conhecimento tais como matemática, estatística, computação e otimização, sendo assim um campo multidisciplinar”. Como processo de negócio para aplicação do presente estudo, foi selecionado o sistema de dipagem de tecidos (Figura 1) utilizado no processo produtivo de uma fábrica têxtil de lonas para reforço de pneus, localizada no Polo Industrial de Camaçari na Bahia. O processo de dipagem dessas lonas precisa ser interrompido para que o equipamento tenha suas configurações ajustadas (*setup*) quando as diferenças de especificações entre o produto fabricado e o próximo na linha de produção indicarem esta necessidade, a partir da avaliação de um operador do chão de fábrica. Considerando a importância dessa tomada de decisão, o fato de ter um processo repetitivo de análise, que exige bastante atenção pelas especificidades existentes entre os produtos, dependendo exclusivamente do fator humano, representa um fator de risco pelo fato de sabidamente existir uma margem de erro intrínseca.

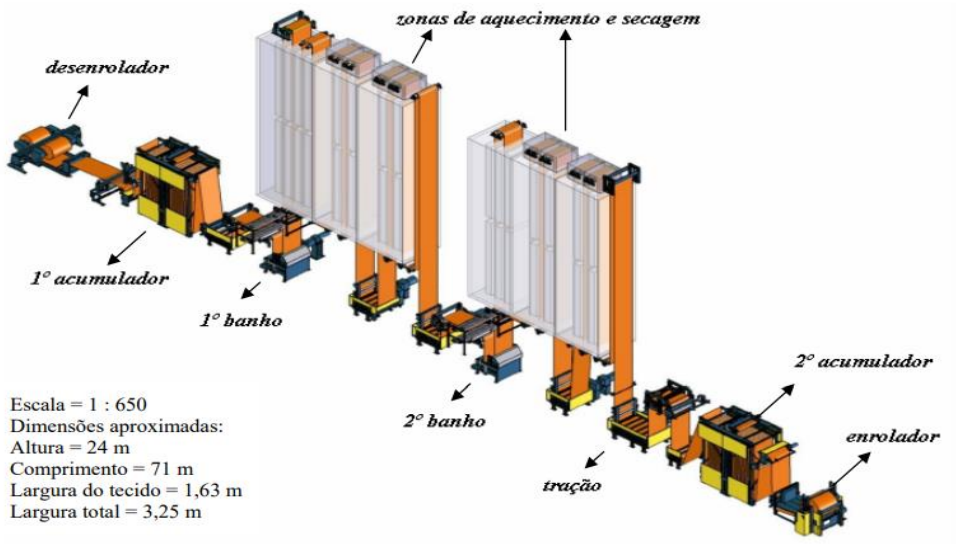
> Segundo Mendes (2014), alguns erros são desvios ou lapsos, muitas vezes, “ações inconscientes ou não planejadas”. Eles ocorrem durante uma tarefa familiar e incluem deslizes (por exemplo, pressionando o botão errado ou a leitura errada de um medidor) e lapsos (por exemplo, esquecendo-se de realizar uma etapa de um processo). Estes tipos de erros ocorrem comumente em procedimentos altamente treinados, onde a pessoa ao realizá-las não precisa se concentrar no que eles estão fazendo. Estes não podem ser eliminados por formação, mas melhorar o projeto pode reduzir a sua probabilidade e fornecer um sistema menos tolerante a erro.

A precisão do momento em que se decide pelo *setup* é muito importante porque a parada do equipamento por equívoco, reduzirá a capacidade produtiva da planta industrial, e por outro lado, se o equívoco for porque não houve parada do

equipamento, gerará risco de qualidade para o próximo produto da linha de produção, um diferencial competitivo desta indústria.

Buscando garantir redução na margem de erro provocada pelo fator humano na decisão de escolha pelo *setup*, este artigo tem como objetivo criar e testar um modelo ML que assuma esta tarefa com desempenho satisfatório, validado por métricas de avaliação, o que tornará o sistema de controle do processo produtivo menos tolerante a falhas.

Figura 1 – Sistema de dipagem para tecidos.



Fonte: CARDOSO, S., 2009

2. DIPAGEM DE TECIDO

> Godinho (2003) *apud* Cardoso, S. (2009) explica o termo dipagem como sendo “processo termoquímico da indústria têxtil pelo qual um tecido cru é submetido a um ou mais banhos de soluções químicas à base de água e látex [...] zonas de aquecimento e secagem para cura [...]”.

A dipagem de tecido é um processo complexo onde um tecido é imerso em uma solução química para atender as propriedades requeridas. A fórmula para preparar esta solução difere conforme cada empresa fabricante ou cliente do produto, sendo considerado parte do segredo industrial, impactando na redução de custos e melhoria de propriedades, sendo a etapa que gera maior valor agregado ao produto. Este processo deve ser feito em determinada velocidade, tensão e temperatura para que o tecido atinja as características físicas e químicas especificadas pelo cliente.

Ao iniciar o processo, o produto é referenciado como “tecido cru”, e após passar por banhos químicos, secagem e cura nos fornos, recebe a referência de “tecido

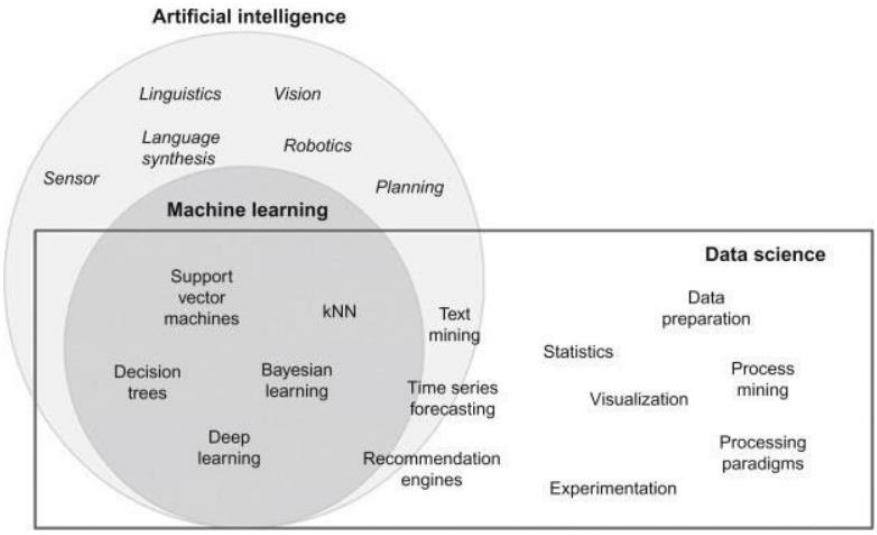
dipado”. As variações nas especificações, além das diferenças nas fórmulas de preparação da solução, possibilitam a produção de diferentes padrões de produtos para cada cliente, e contribuem para sua complexidade. Dependendo do grau destas variações ocorrerá o *setup*, mediante a passagem de um tecido especial pelo equipamento, chamado *liner*, tecido este que desempenha também a função de limpeza dos resíduos acumulados de processos anteriores.

3. APRENDIZAGEM DE MÁQUINA

> A aprendizagem de máquina (ou, em inglês, *Machine Learning*) é uma área da Ciência da Computação que lida com algoritmos que aprendem por experiência e melhoram suas performances com o decorrer do tempo. Essa abordagem é normalmente utilizada para a detecção de padrões em dados, visando à automatização de tarefas complexas ou fazer predições, e vêm se tornando um diferencial em diversas áreas [...] (INAZAWA *et al*., 2019, p. 20).

O ML possui relação com duas outras áreas do conhecimento, a Inteligência Artificial (IA) e a Ciência de Dados, que por estarem ligadas muitas vezes são tratadas como sinônimos quando na realidade são áreas complementares. A IA tem o objetivo principal de fazer uma máquina se comportar como um humano, imitando o funcionamento do cérebro. A Ciência de Dados é um campo do conhecimento que recorre as áreas da matemática, estatística, computação e IA para buscar *insights* de informações sobre um conjunto de dados, sendo o ML uma das formas da Ciência de dados atuar através de diferentes técnicas e algoritmos.

Figura 2 – Relação entre Data Science, AI e ML.



Fonte: ILUMEO, 2020

O ML possui quatro tipos de aprendizagem, são eles, supervisionado, não supervisionado, semi-supervisionado e por reforço. O aprendizado supervisionado se caracteriza pelo treinamento do modelo com os dados de entrada e de saída (resultado) conhecidos, enquanto o aprendizado não supervisionado se caracteriza pelo treinamento do modelo buscando extrair padrões de um conjunto de dados, sem ter conhecimento da relação entre estes dados. A utilização dos aprendizados supervisionado e não supervisionado em um mesmo modelo caracteriza o aprendizado semi-supervisionado, possibilitando criar um modelo que permita o conhecimento apenas parcial das saídas, o que eventualmente por ocorrer. Finalmente, o aprendizado por reforço é aquele onde o algoritmo aprende o melhor caminho por tentativa e erro, baseado em recompensas por suas decisões.

Neste estudo, como a variável de saída (classe) é conhecida em todo o conjunto de dados, o tipo de aprendizagem supervisionado se apresenta como ideal para solucionar o problema apresentado.

A teoria de ML é implementada através de algoritmos, conjunto de instruções e regras a serem executadas por um computador. O grupo destes algoritmos que atendem aos requisitos da aprendizagem supervisionada recebem a denominação “algoritmos de classificação”. Estes algoritmos de classificação por sua vez, quando estão diante de um problema onde a variável de saída (classe) possui apenas 2 resultados possíveis, são chamados de classificadores binários, neste caso, teremos a decisão pelo *setup* que poderá ser “sim” (1) ou “não” (0).

Neste artigo, 7 modelos ML serão avaliados entre si, a partir de métricas que contribuirão para determinar aquele com maior efetividade de generalização, capacidade de acertar a previsão quando exposto a novos casos, uma vez treinado. A seguir são apresentados os respectivos algoritmos de classificação binária selecionados para o estudo:

- **Algoritmo Árvore de Decisão ou Árvore de Classificação:** Como o nome sugere, este algoritmo refere-se a uma estrutura com característica semelhante a uma árvore, composta por “nós”, onde cada “nó” representa uma decisão lógica a ser tomada. O primeiro “nó” é chamado de raiz, então a partir daí ocorrem ramificações com regras de classificação gerando outros nós chamados de folha, até que se obtenha um resultado, o rótulo de classe na extremidade. Para se obter uma

árvore de classificação eficiente com melhores decisões costuma-se utilizar uma medida que calcula “pureza” de amostras de dados para se obter melhores escolhas. Assim, adota-se a medida Entropia, cálculo de ganho de informação originado da teoria da informação, ou o Índice Gini, critério que mede o grau de heterogeneidade dos dados, obtendo amostras de dados consideradas mais puras.

Vantagens: Requer menos dados para treinamento que outros modelos, além de ser de fácil interpretação e tolerante a valores ausentes.

Desvantagens: É instável a modificações, podendo alterar suas previsões a partir de uma leve mudança nos dados, e não são indicadas individualmente para predições muito robustas, por isso é muito comum se combinar modelos de árvores de decisão formando uma floresta (*Random Forest*) para suprir este ponto fraco;

- **Algoritmo *Random Forest***: Trata-se de um algoritmo resultante de um método chamado *Ensemble*, onde o resultado de vários modelos ML, neste caso o algoritmo Árvore de Decisão, é apresentado de maneira única, daí o nome Floresta Aleatória. Portanto, a classificação neste modelo será o resultado do processamento de múltiplas Árvores de Decisão.

  Vantagens: Possui alto grau de precisão e é menos suscetível a sobreajuste (*overfitting*), quando o modelo trabalha muito bem com os dados de treino, mas não consegue generalizar com dados novos.

  Desvantagens: Tem baixa performance, principalmente para fazer predição em tempo real devido ao grande número de árvores necessárias, exigindo recurso computacional.

- **Algoritmo *Naive Bayes*:** Classifica dados a partir de uma tabela de probabilidades, tendo como característica não levar em conta a correlação entre variáveis, tratando cada uma de forma independente, o que levou ao seu nome “ingênuo” (*Naive*). Já o termo *Bayes* vem da sua origem, o Teorema de *Bayes* publicado em 1812 por Pierre-Simon Laplace.

  Vantagens: Tem boa performance porque é baseado em uma equação matemática simples, necessita de poucos dados de treinamento e é altamente escalável.

  Desvantagens: Desconsidera a correlação entre as variáveis e não é reconhecido como um bom estimador porque suas probabilidades não são consideradas precisas. Também não consegue fazer uma previsão se uma categoria de uma

variável não existir no conjunto de treinamento, pois o modelo atribuirá uma probabilidade zero conhecida como “Frequência Zero”, a menos que seja aplicada uma técnica de suavização chamada estimativa de *Laplace*.

- **Algoritmo KNN (K-*Nearest Neighbors*)**: Classifica amostras de um conjunto de dados conforme distância em relação aos K vizinhos mais próximos, onde K é definido como parâmetro. Define uma amostra como sendo de uma classe se os vizinhos mais próximos forem desta classe em sua maioria.
  Vantagens: É simples na teoria, implementação e interpretação, e trata-se de um algoritmo do tipo *lazy-learning* que faz uma aprendizagem por analogia, não havendo fase de treino para fazer a predição baseando-se na memorização dos dados de treino, o que o torna muito mais rápido do que outros algoritmos.
  Desvantagens: Tem baixa performance para conjunto de treinamento maiores e para um número elevado de variáveis de entrada, exigindo recurso computacional.

- **Algoritmo Regressão Logística**: Classifica de maneira probabilística se um evento vai ocorrer, utilizando a função matemática logística também chamada de curva logística ou sigmóide, em formato de “S”.
  Vantagens: É de fácil implementação, simples de treinar, possui alto grau de confiabilidade e tem boa performance principalmente se os dados forem linearmente separáveis.
  Desvantagens: Não consegue resolver problemas não lineares e é vulnerável a sobreajuste (*overfitting*).

- **Algoritmo SVM (*Support Vector Machine*)**: Classifica os dados em grupos distintos representados por pontos (coordenadas) e separados por uma linha conhecida como hiperplano. Os vetores de suporte são os pontos das extremidades de cada grupo que representam a menor distância entre si, formando uma margem que definirá o melhor hiperplano de separação. Para dados não separáveis linearmente é utilizado o recurso do Kernel, onde transforma os dados de uma dimensão para duas dimensões, e nesta, passam a ser separáveis linearmente. Faz-se esse processo mapeando cada ponto em uma dimensão para um par ordenado correspondente nas duas dimensões.

Vantagens: Muito eficiente para problemas com alta dimensionalidade e quando o número de dimensões é maior que o número de amostras.

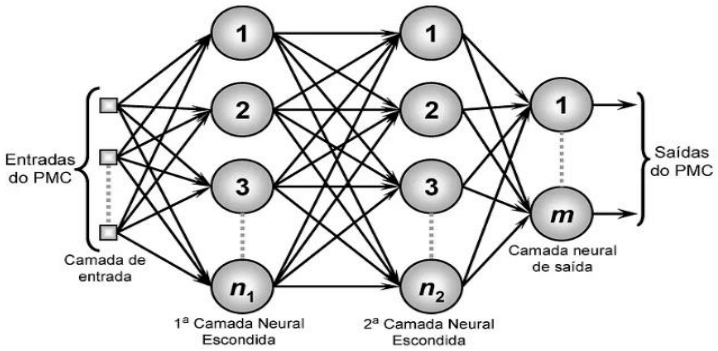
Desvantagens: Tem baixa performance para conjunto de treinamento maiores, exigindo recurso computacional, e tem complexa interpretação da estratégia de decisão, por isso considerado um classificador “caixa-preta”.

- **Algoritmo MLP (*Multilayer Perceptron*)**: Antes de tratar das características deste algoritmo é preciso destacar que “as redes neurais artificiais (RNAs) são sistemas computacionais adaptativos inspirados nas características de processamento de informação semelhante ao neurônio biológico de organismos inteligentes” (HAYKIN, 1999 *apud* NETO; BONINI). A arquitetura mais simples de uma RNA é chamada de perceptron, um modelo matemático composto de n entradas e uma saída binária, uma camada única com um nó mais conhecido como neurônio. Neste estudo será utilizado a RNA perceptron multicamadas ou MLP (*Multilayer Perceptron*), composta por camadas de neurônios, uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída (Figura 3). As camadas intermediárias são também conhecidas como camadas ocultas por não ter interações com agentes externos, recebendo e enviando dados de outros neurônios. A classificação dos dados se dá através do processamento de neurônios interligados com um sistema de tolerância a falhas e aprendizado através de padrões tendo como saída um valor que representa uma das classes esperadas.

  Vantagens: Ser eficiente em problemas com alto grau de não linearidade.

  Desvantagens: Tem baixa performance em tempo de treinamento, exigindo recurso computacional, e tem complexa interpretação da estratégia de decisão, por isso considerado um classificador “caixa-preta”.

Figura 3 – Diagrama do Perceptron Multicamadas.



Fonte: MOREIRA, 2018.

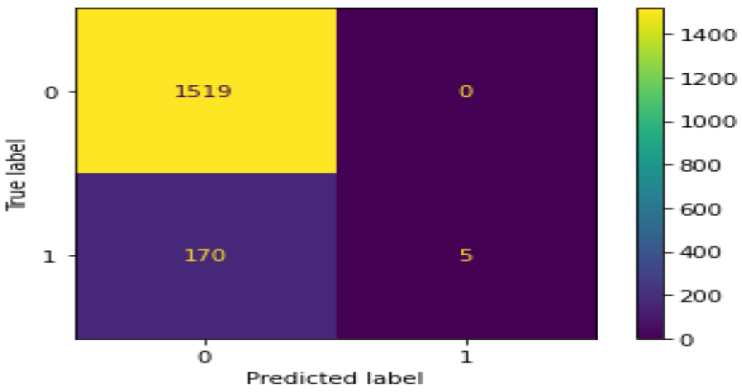
4. MÉTRICAS DE AVALIAÇÃO

> Há inúmeras métricas de avaliação para algoritmos de Aprendizado de Máquina, que possibilitam identificar se as técnicas de pré-processamento aplicadas resultam numa melhoria significativa para o projeto ou não. As métricas possibilitam avaliar, quantitativamente, os algoritmos de classificação utilizados nos experimentos. [...]" (SHALEV-SHWARTZ; BEN-DAVID, 2014 apud CARDOSO, I., 2022).

Um modelo de predição ML precisa ser avaliado para garantir sua qualidade, e isso ocorre através de métricas que indicarão sua efetividade. Os modelos de classificação possuem métricas específicas também chamadas de métricas de classificação, as quais indicarão o quão próximo o modelo se encontra da perfeição. Dentre essas métricas, existem as mais indicadas para o objetivo do modelo que está sendo criado, e no caso do presente estudo, o objetivo principal é identificar a classe positiva (1), ou seja, se deve parar o processo produtivo para aplicar o *setup* pois a não aplicação poderá contribuir para prejuízos de qualidade ou até mesmo de perda de material. Com base nesse princípio, foram selecionadas as seguintes métricas para avaliar os 7 modelos de classificação do estudo: acurácia, precisão, revocação, f1-score, mcc, log loss, auroc e auprc. Adicionalmente, como parte do processamento de uma rede neural, utilizou-se da métrica função de perda ou de custo para garantir a validade do modelo MLP.

Uma forma de visualizar a qualidade de um modelo de classificação binária é visualizar seus resultados em uma matriz quadrada (2x2) chamada matriz de confusão. Trata-se basicamente de uma tabela que cruza quantidades de acertos e erros entre os dados preditos e reais das classes.

Figura 4 – Matriz de Confusão.



Fonte: CARDOSO, I., 2022

Na Figura 4 pode-se ver um exemplo, onde 1.519 dados reais da classe negativa (0) tiveram 1.519 (100%) dados preditos corretamente, e 175 dados reais de classe positiva (1) tiveram 170 (97,14%) dados preditos corretamente e 5 (2,86%) dados preditos erroneamente. Pode-se ler também que os dados são distribuídos em 5 verdadeiros positivos (VP), 1.519 verdadeiros negativos (VN), 0 falso positivo (FP) e 170 falsos negativos (FN).

4.1 ACURÁCIA

A métrica acurácia indica o quanto o modelo conseguiu atingir seu objetivo de classificar corretamente, sendo considerada uma métrica importante, porém simples, e por isso indicada apenas para uma avaliação preliminar mostrando a tendência do modelo, não servindo para escolha do modelo ideal numa análise conjunta. Trata-se do percentual de acertos obtido pela razão da quantidade de acertos sobre a quantidade total de entradas.

Não se comporta bem com dados desbalanceados porque tenderá para o grupo de maior incidência, causando uma falsa indicação de bom desempenho. Por exemplo, um modelo que foi treinado sobre um conjunto de dados financeiros com identificação de fraude em apenas 5% das entradas, as predições de validação e teste tenderão para as entradas onde não indica fraude, simplesmente porque representaram 95% das entradas do treinamento. Sendo assim, a acurácia dará a impressão de que os acertos do modelo foram altos por conta do bom ajuste do modelo quando na realidade o modelo estará viciado.

4.2 PRECISÃO e REVOCAÇÃO (*RECALL*)

Precisão e revocação são métricas de classificação indicadas quando a classe positiva (1) é de maior interesse, como neste estudo. Enquanto a métrica precisão apresenta quantas classes positivas o modelo acertou dentre as classes positivas preditas, a métrica revocação apresenta quantas classes positivas o modelo detectou dentre todas as classes realmente positivas. Por exemplo, em um modelo criado para identificar se um e-mail é spam, foi predito que 300 e-mails eram spam, contudo somente 150 destes e-mails realmente eram spam, isso quer dizer que a precisão foi de 50%, por outro lado, considerando todo o conjunto de dados, 1.000 e-mails eram spam, então a revocação foi de 15%, já que o modelo acertou apenas

150 e-mails spam conforme já citado. Pode-se observar pela precisão que tiveram 150 e-mails falsos positivos (FP) e pela revocação que tiveram 850 e-mails falsos negativos (FN).

Segundo Marcondes (2021), “um aumento na precisão, em geral, pode trazer uma redução no número de positivos que é descoberto pelo modelo. [...] um modelo muito preciso raramente erra ao apontar um positivo, porém, como consequência, pode acabar apontando poucos positivos.”. A essa característica dar-se o nome de *trade-off,* e no caso, tem-se um *trade-off* entre precisão e revocação.

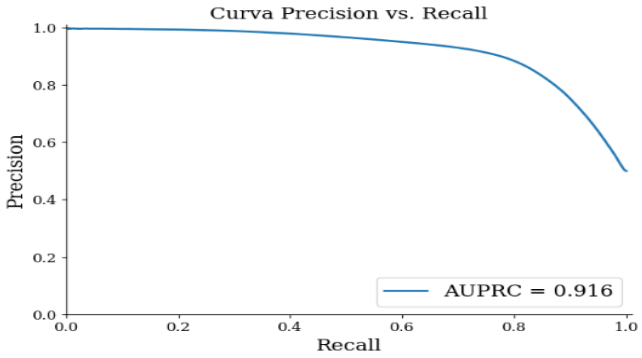
4.3 F1-SCORE

F1-*Score* é uma métrica que representa a média harmônica entre as métricas precisão e revocação, ou seja, observando-a estará avaliando essas duas métricas através de um valor único. O uso da média harmônica é importante porque é uma média que valoriza os valores baixos, e desta forma, não causa distorção quando existe um valor de precisão ou revocação muito baixo, o que não aconteceria se fosse uma média aritmética. Por exemplo, se um modelo estiver com uma precisão de 15% e uma revocação de 95%, a média aritmética será de 55%, o que poderá ser interpretado como um modelo de desempenho médio, e seria um equívoco já que o modelo demonstrou uma precisão fraca. Por outro lado, a média harmônica dará 26% estando mais conforme com a realidade do exemplo. Resumindo, um F1-*Score* alto quer dizer que ambas as métricas, precisão e revocação, estão com bom desempenho.

4.4 AUPRC (*Area Under the Precision-Recall Curve*)

A métrica **AUPRC** consiste no cálculo da área abaixo de uma curva, neste caso da curva de precisão e revocação, demonstrando a comparação entre essas 2 métricas. A curva corresponde a precisão e revocação ponderada de cada classe para limiares que variam entre 0 e 1, e quanto mais próximo de (1,1) estiver, melhor será o modelo. A possibilidade de analisar a relação entre os valores de precisão e revocação ao longo destes limiares, identificando o melhor ponto de corte, representa uma vantagem desta métrica na comparação entre classificadores.

Figura 5 – Curva AUPRC.



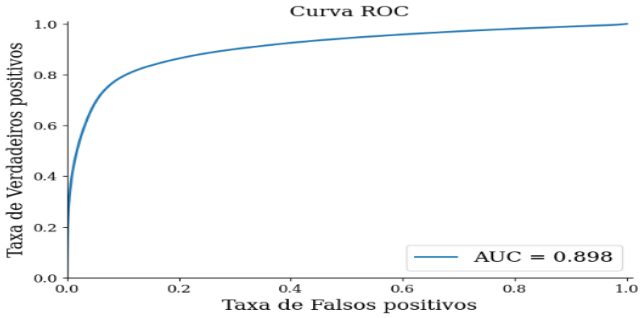
Fonte: BENTO, VASSALO; SAMATELO, 2021

No exemplo da Figura 5 observa-se um modelo com um bom desempenho, tanto pelo visual onde a curva se aproxima do canto superior direito quanto pelo próprio valor do indicador, 0.916.

4.5 AUROC (*Area Under the Receiver Operating Characteristic Curve*)

A métrica AUROC ou AUC consiste no cálculo da área abaixo da chamada curva ROC (curva característica de operação do receptor). A curva ROC (Figura 6) corresponde a taxa de verdadeiro positivo (TVP), classes positivas que o modelo previu corretamente, sobre a taxa de falso positivo (TFP), classes negativas que o modelo previu erroneamente como positiva. O indicador TVP é a própria métrica revocação também conhecida como sensibilidade, e o indicador TFP representa a diferença entre 1 e a métrica especificidade, representada pela taxa de verdadeiro negativo (TVN), classes negativas que o modelo previu corretamente. A curva ROC representa limiares que variam entre 0 e 1, e quanto mais próximo de (0,1) estiver, melhor será o modelo. Assim como na curva precisão-revocação, esta curva tem como vantagem a identificação do melhor ponto de corte, aquele que tem o valor de melhor desempenho, para os modelos avaliados.

Figura 6 – Curva ROC.



Fonte: BENTO, VASSALO; SAMATELO, 2021

No exemplo da Figura 6 observa-se um modelo com um bom desempenho, tanto pelo visual onde a curva se aproxima do canto superior esquerdo quanto pelo próprio valor do indicador, 0.898.

4.6 MCC (*Matthews Correlation Coefficient*)

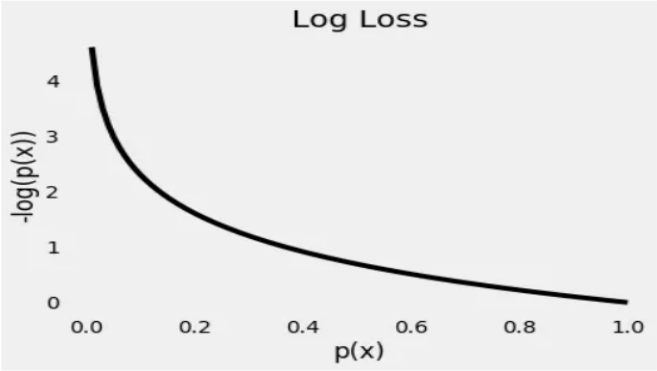
O MCC, conhecido na estatística como coeficiente Phi, é uma métrica de classificação robusta que contempla todas as possibilidades apresentadas na matriz de confusão, acertos e erros das classes positiva e negativa. Trata as classes reais e preditas como 2 variáveis, calculando o coeficiente de correlação entre elas, e quanto maior for essa correlação, melhor será a previsão. O coeficiente retorna um valor normalizado entre -1 e +1, onde mais próximo de -1 significa que o modelo está errando mais que acertando, mais próximo de +1 significa que o modelo acertando mais que errando, ou seja, ambas as classes são bem previstas, e mais próximo de 0 significa que o modelo está “chutando” a previsão de maior frequência, ou seja, as classes possuem correlação fraca.

4.7 LOG LOSS (*Logarithmic Loss*)

Log loss ou *Binary Cross Entropy* (entropia cruzada) é uma função de perda que representa uma métrica de avaliação para modelos de classificação. Faz uma avaliação das probabilidades geradas por um modelo de classificação, penalizando previsões com falhas grosseiras, desta forma avalia a confiabilidade do modelo. Por exemplo, se o classificador indicar erroneamente que uma classe binária é 0 com uma probabilidade de 99%, estará cometendo um erro grave, mas se indicar uma probabilidade de 55%, estará cometendo um erro pequeno por se tratar de uma probabilidade próxima ao limite de decisão de 50%. Trata-se de uma métrica considerada negativa porque quanto mais próximo de zero, melhor estará o modelo, e na outra direção, quanto maior seu valor mais a probabilidade prevista estará divergente da classe real.

Observa-se na Figura 7 um exemplo do gráfico log loss da probabilidade para acertar uma classe positiva (1), onde quanto mais próximo a zero estiver a probabilidade, maior será a penalidade, chegando a um custo exponencial.

Figura 7 – Função Log Loss.

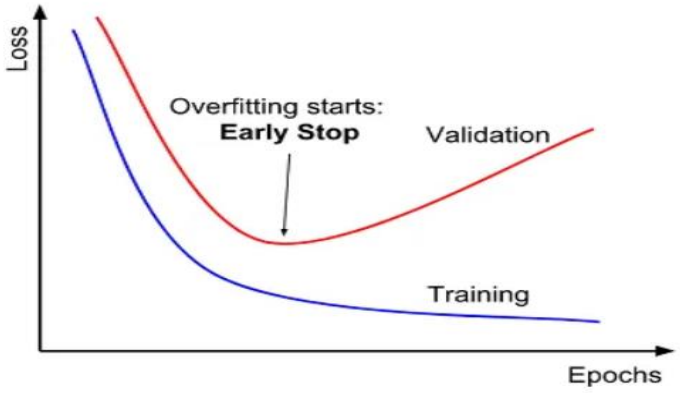


Fonte: GODOY, 2018

4.8 FUNÇÃO DE PERDA (LOSS)

A função de perda (*loss function*) ou função de custo (*cost function*) avalia o quão bem um modelo de redes neurais está fazendo suas predições. Trata-se de um dos principais responsáveis pelos ajustes de pesos na construção dos modelos de redes neurais, atuando durante a retropropagação destes pesos que são reinseridos na rede. Esse ciclo se repete até a conclusão das iterações (épocas) ou até que seja observado a saturação na melhoria do desempenho do modelo. O método chamado parada precoce ou parada antecipada (*early stopping*) pode ser usado para calcular a precisão da classificação ao final de cada iteração (época), interrompendo o treinamento quando a previsão parar de melhorar, evitando inclusive o sobreajuste (*overfitting*) do modelo, conforme apresentado na Figura 8.

Figura 8 – Função Loss.



Fonte: TITO, 2020

5. PROJETO *MACHINE LEARNING*

> A chave para entregar um bom projeto de *Machine Learning* é entender o problema, resolver este problema e produzir um resultado que atenda às suas necessidades. Então, é preciso ter noção do problema, dos dados envolvidos e do contexto. Além disso, é preciso ter em mente o objetivo da empresa, e se é possível atingi-lo utilizando técnicas de aprendizado de máquina. (SHASHANK, 2022 *apud* ILUMEO, 2022)

Em conformidade com Shashank, procurou-se aqui o conhecimento sobre o negócio dipagem de tecidos na indústria têxtil, entendendo como problema central do estudo, os riscos envolvidos na escolha de quando parar o processo produtivo para transição de parâmetros de especificação e limpeza de equipamento, o chamado *setup*, sem uma ferramenta inteligente de apoio, riscos estes que podem impactar os resultados da empresa. Como solução ao exposto, optou-se pela implementação de um modelo ML como a citada ferramenta inteligente dando mais segurança a tomada de decisão.

O Projeto ML foi desenvolvido em 5 etapas: (1) Coleta de dados; (2) Pré-processamento; (3) Análise Exploratória; (4) Treinamento e otimização dos modelos; e (5) Análise e avaliação de resultados, tendo como produto um modelo de classificação de dados que sinaliza quando aplicar o setup em uma máquina de dipagem de tecidos.

Figura 9 – Etapas de Projeto ML.

Fonte: Elaborado Própria

A implementação de um Projeto ML exige o uso de determinadas tecnologias capazes de atender aos requisitos exigidos pela ciência de dados, e neste caso foi escolhida a ferramenta *Google Colaboratory*, mais conhecida como Colab, uma plataforma que disponibiliza um ambiente de desenvolvimento em nuvem, ou seja, acessível *on-line* através da internet, com utilização da linguagem de programação *Python.*

5.1 COLETA DE DADOS

Foram levantados dados dos registros de *setup* com a identificação dos produtos envolvidos antes e depois do mesmo, apontados ao longo do tempo no pro-

cesso industrial “dipagem de tecidos” da fábrica têxtil selecionada para este estudo, abrangendo um período de 6 anos (2017-2022) ordenados cronologicamente, contabilizando 15.312 linhas.

Pode-se observar na Tabela 1, o dicionário de dados do referido conjunto de dados (*dataset*) distribuídos em 39 variáveis ou atributos, sendo 9 variáveis categóricas nominais, 6 variáveis numéricas discretas e 24 variáveis numéricas contínuas. Destas, 5 variáveis de identificação, 1 variável de série temporal, 32 variáveis de especificação de processo e produto e 1 variável de interesse representando a classe que se deseja predizer com o modelo de classificação binária.

Tabela 1 – Variáveis do conjunto de dados.

Variável	Descrição	Classificação
id	Nº de ordem do processo produtivo de dipagem de tecido	Numérica Discreta
nu_ano	Ano quando ocorreu o processo	Numérica Discreta
nu_produto_a	Nº de Produto A	Numérica Discreta
cd_artigo_a	Código do Artigo de Produto A	Categórica Nominal
cd_tipo_material_pa	Código do Tipo de Material de Produto A	Categórica Nominal
cd_solucao_1b_pa	Código de Solução para 1º Banho do Produto A	Categórica Nominal
cd_solucao_2b_pa	Código de Solução para 2º Banho do Produto A	Categórica Nominal
vl_temp_f1_pa	Temperatura Forno 1 para Secagem na Dipagem do Produto A	Numérica Contínua
vl_temp_f2a_pa	Temperatura Forno 2A para Secagem na Dipagem do Produto A	Numérica Contínua
vl_temp_f2b_pa	Temperatura Forno 2B para Secagem na Dipagem do Produto A	Numérica Contínua
vl_temp_f5_pa	Temperatura Forno 5 para Secagem na Dipagem do Produto A	Numérica Contínua
vl_temp_f3_pa	Temperatura Forno 3 para Processo na Dipagem do Produto A	Numérica Contínua
vl_temp_f4_pa	Temperatura Forno 4 para Processo na Dipagem do Produto A	Numérica Contínua
vl_temp_f6_pa	Temperatura Forno 6 para Processo na Dipagem do Produto A	Numérica Contínua
vl_temp_f7_pa	Temperatura Forno 7 para Processo na Dipagem do Produto A	Numérica Contínua
vl_tensao_z2_pa	Tensão em Zona 2 de Fornos na Dipagem do Produto A	Numérica Contínua
vl_tensao_z4_pa	Tensão em Zona 4 de Fornos na Dipagem do Produto A	Numérica Contínua
qt_fornos_usados_pa	Quantidade de Fornos usados na Dipagem do Produto A	Numérica Discreta
qt_largura_pa	Largura do Produto A	Numérica Contínua
vl_velocidade_pa	Velocidade de Processamento da Dipagem do Produto A	Numérica Contínua
nu_produto_b	Nº de Produto B	Numérica Discreta
cd_artigo_b	Código do Artigo de Produto B	Categórica Nominal
cd_tipo_material_pb	Código do Tipo de Material de Produto B	Categórica Nominal
cd_solucao_1b_pb	Código de Solução para 1º Banho do Produto B	Categórica Nominal
cd_solucao_2b_pb	Código de Solução para 2º Banho do Produto B	Categórica Nominal
vl_temp_f1_pb	Temperatura Forno 1 para Secagem na Dipagem do Produto B	Numérica Contínua
vl_temp_f2a_pb	Temperatura Forno 2A para Secagem na Dipagem do Produto B	Numérica Contínua
vl_temp_f2b_pb	Temperatura Forno 2B para Secagem na Dipagem do Produto B	Numérica Contínua
vl_temp_f5_pb	Temperatura Forno 5 para Secagem na Dipagem do Produto B	Numérica Contínua
vl_temp_f3_pb	Temperatura Forno 3 para Processo na Dipagem do Produto B	Numérica Contínua
vl_temp_f4_pb	Temperatura Forno 4 para Processo na Dipagem do Produto B	Numérica Contínua
vl_temp_f6_pb	Temperatura Forno 6 para Processo na Dipagem do Produto B	Numérica Contínua
vl_temp_f7_pb	Temperatura Forno 7 para Processo na Dipagem do Produto B	Numérica Contínua
vl_tensao_z2_pb	Tensão em Zona 2 de Fornos na Dipagem do Produto B	Numérica Contínua
vl_tensao_z4_pb	Tensão em Zona 4 de Fornos na Dipagem do Produto B	Numérica Contínua
qt_fornos_usados_pb	Quantidade de Fornos usados na Dipagem do Produto B	Numérica Discreta
qt_largura_pb	Largura do Produto B	Numérica Contínua
vl_velocidade_pb	Velocidade de Processamento da Dipagem do Produto B	Numérica Contínua
cd_passa_liner	Confirmação Setup - Sim (1) / Não (0)	Categórica Nominal

Fonte: Elaboração Própria

5.2 PRÉ-PROCESSAMENTO E ANÁLISE EXPLORATÓRIA

Antes da criação do modelo ML, tem-se como condição obrigatória a passagem pelas etapas de pré-processamento que envolve a preparação, organização e estruturação dos dados, e a análise exploratória dos dados, ou seja, a análise das características do conjunto de dados. Essas etapas são cíclicas, ou seja, são executadas uma após a outra quantas vezes forem necessárias, contribuindo para o desempenho satisfatório dos modelos estudados, uma vez que prepara os dados em conformidade com as orientações técnicas de cada algoritmo.

Segundo Silva (2014), as principais etapas envolvidas no pré-processamento de dados são a integração dos dados, limpeza dos dados, redução dos dados e transformação dos dados.

A integração dos dados se faz necessária quando os dados coletados se encontram em fontes diferentes. Neste caso, os dados encontravam-se em 2 fontes distintas, planilhas eletrônicas e banco de dados relacional, o que exigiu a integração dos mesmos em um conjunto de dados único através da análise de cada variável, tratando quaisquer redundâncias e conflitos eventuais de valores, ou seja, houve um alinhamento dos dados e o relacionamento entre os mesmos através de variáveis chaves para que a integração tivesse sucesso.

A limpeza de dados busca verificar existência de dados ausentes, ruidosos, discrepantes e inconsistentes que possam influenciar negativamente nos resultados do modelo ML.

Verificou-se ausência de dados em um conjunto de linhas de 6 variáveis, sendo 2 variáveis ligadas a largura de produto e 4 variáveis ligadas ao código de solução usada no banho de produto, conforme descrição abaixo.

- 2 linhas em largura de produto A.
- 2 linhas em largura de produto B.
- 6.667 linhas em código de solução para 1º banho de produto A.
- 6.672 linhas em código de solução para 1º banho de produto B.
- 1.674 linhas em código de solução para 2º banho de produto A.
- 1.674 linhas em código de solução para 2º banho de produto B.

As linhas com dados ausentes nas variáveis relativas à largura foram excluídas uma vez avaliado as seguintes considerações:

- Todas as linhas são relacionadas a um mesmo código de produto, produte este que não mais foi produzido durante o período trabalhado, o que não permitiu a localização de alguma referência desta largura.
- A quantidade de linhas possui baixa representatividade, uma vez que equivalem a 0,026% do total de linhas do conjunto de dados.
- O código se refere a um produto usado para teste a época, conforme atestado pelo responsável técnico do negócio.

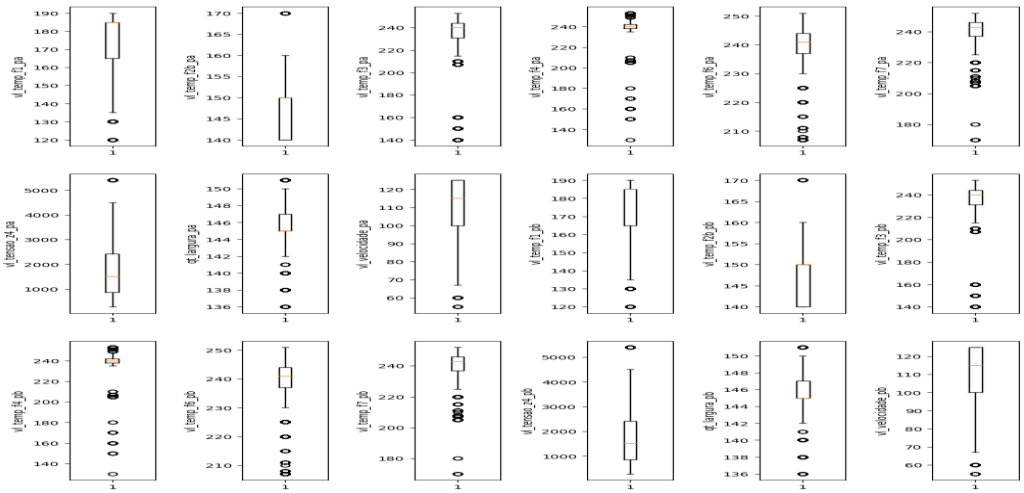
As linhas com dados ausentes das variáveis relativas a código de solução para banho, foram preservadas uma vez analisado as seguintes considerações:

- Foi verificado que não é obrigatório a aplicação de 2 banhos no processo produtivo “dipagem de tecidos”.
- Não foi detectada linha alguma com códigos de solução ausentes nos 2 banhos possíveis.

Verificou-se inconsistência nas variáveis de tensão do processo de dipagem, onde 30 linhas estavam com valores zerados nas 2 tensões possíveis da especificação do produto, o que provocou a exclusão das mesmas.

Para verificar existência de dados ruidosos discrepantes, os chamados *outliers,* optou-se por utilizar o método *Tukey*, o qual apresenta essa discrepância visualmente através do gráfico *Box Plot*. Neste gráfico, os dados são divididos em 4 partes iguais contendo 25% dos elementos, e conhecidas como *quartis*, que demonstram a dispersão dos mesmos. Para além dos quartis são calculados os limites inferior e superior que delimitam a fronteira para um valor não ser considerado *outlier*.

Figura 10 – *Box Plot* apresentando *Outliers*.



Fonte: Elaboração Própria

Nos gráficos *box plot* apresentados na Figura 10 pôde ser confirmado a presença de *outliers* no conjunto de dados deste estudo, sendo identificados dados discrepantes em 18 variáveis, especificamente em variáveis relacionadas a temperatura, tensão, largura e velocidade, destacando-se os índices de discrepância nos dados de temperatura do forno 3 (vl_temp_f3_pa e vl_temp_f3_pb) e largura de produtos (qt_largura_pa e qt_largura_pb) com valores acima de 20%. Contudo, ao apresentar esses valores de *outliers* ao Eng. de Tecnologia responsável técnico pelo processo, foi avaliado tratar-se de diferenças naturais inerentes ao negócio por conta das características específicas definidas por cada cliente, ou seja, não é indicado tratamento sobre os mesmos.

Dentro da etapa de pré-processamento relativa a redução de dados para otimizar o desempenho do modelo, observou-se a possibilidade de redução de dimensionalidade do conjunto de dados. Primeiro, foram identificadas e excluídas algumas variáveis sem relevância para a tarefa de classificação, 5 variáveis de identificação e 1 variável de tempo conforme descrito abaixo.

- **id**: Identificação do número de ordem de execução do processo de dipagem.
- **nu_produto_a**: Identificação do volume produzido passado pela dipagem.
- **nu_produto_b**: Identificação do volume produzido a passar pela dipagem.
- **cd_artigo_a**: Identificação do padrão técnico do produto passado pela dipagem.
- **cd_artigo_b**: Identificação do padrão técnico do produto a passar pela dipagem.
- **nu_ano**: ano quando ocorreu o processo de dipagem.

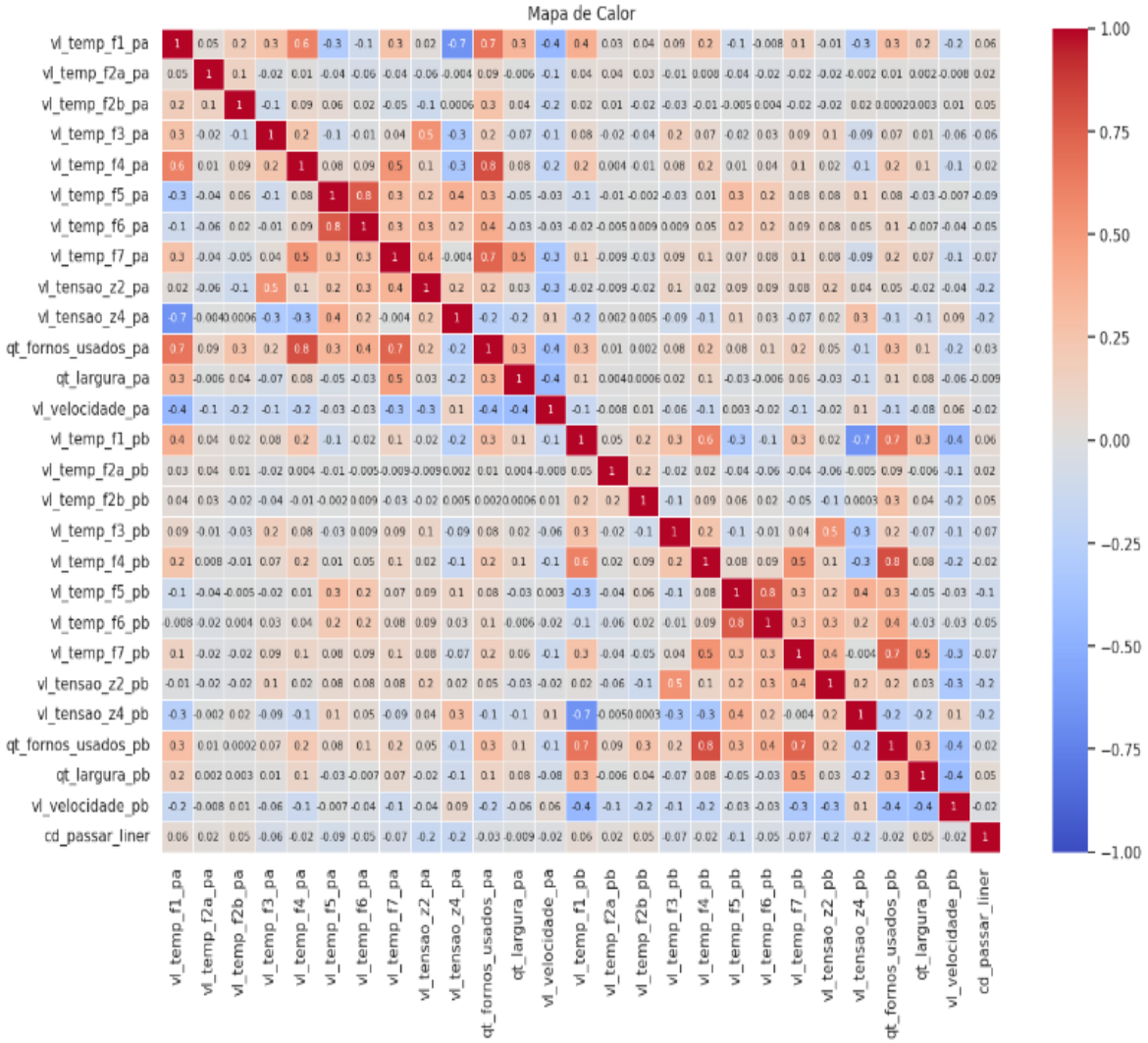
A seguir foram identificadas mais 2 características que foram tratadas, a presença de variáveis categóricas e redundantes, conforme mapa de calor (Figura 11). Assim, as ações a seguir se mostraram indicadas.

- Exclusão das variáveis quantidade de fornos usados para produtos A e B (qt_fornos_usados_pa e qt_fornos_usados_pb) já que essa informação será identificada pelo algoritmo com base na existência de registro das temperaturas;
- Criação de variáveis para registro das diferenças entre temperaturas, tensões, velocidades e larguras dos produtos A e B;
- Criação de variável para registro de mudança em alguma das soluções de banho aplicadas sobre o produto A e B;
- Criação de variável para registro de mudança no tipo de material entre o produto A e B;

- Exclusão de variáveis de temperaturas, tensões, velocidade, largura, soluções de banho e tipo de material do produto A e do produto B.

Em resumo, a redução de dimensionalidade sobre o conjunto de dados gerou uma redução de 24 variáveis, saindo de 39 para 15 variáveis (Tabela 2), contribuindo para simplificação do modelo e redução do tempo de treino.

Figura 11 – Mapa de Calor de Correlações.



Fonte: Elaboração Própria

Quanto a transformação de dados, observou-se a necessidade de aplicar a técnica de normalização, uma vez verificado que as variáveis do conjunto de dados possuem ordens de grandeza diferentes, situação não tratada pelos algoritmos KNN, Regressão Logística, SVM e MLP.

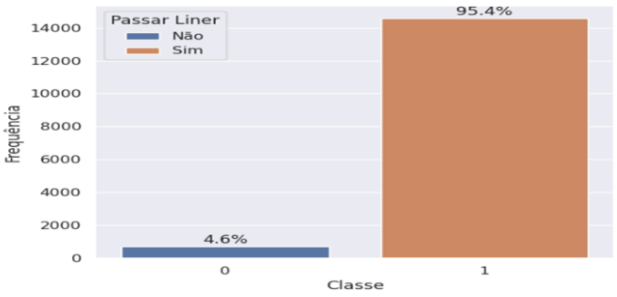
Tabela 2 – Variáveis após redução de dimensionalidade.

Variável	Descrição	Classificação
fl_tipo_material_alterado	Confirma alteração de tipo material	Categórico Nominal
fl_solucao_banho_alterado	Confirma alteração solução para banho	Categórico Nominal
vl_dif_temp_forno1	Diferença de temperatura forno 1	Numérico Discreto
vl_dif_temp_forno2a	Diferença de temperatura forno 2a	Numérico Discreto
vl_dif_temp_forno2b	Diferença de temperatura forno 2b	Numérico Discreto
vl_dif_temp_forno3	Diferença de temperatura forno 3	Numérico Discreto
vl_dif_temp_forno4	Diferença de temperatura forno 4	Numérico Discreto
vl_dif_temp_forno5	Diferença de temperatura forno 5	Numérico Discreto
vl_dif_temp_forno6	Diferença de temperatura forno 6	Numérico Discreto
vl_dif_temp_forno7	Diferença de temperatura forno 7	Numérico Discreto
vl_dif_tensao_z2	Diferença de tensão zona 2	Numérico Discreto
vl_dif_tensao_z4	Diferença de tensão zona 4	Numérico Discreto
vl_dif_velocidade	Diferença de velocidade	Numérico Discreto
qt_dif_largura	Diferença de largura	Númerico Contínuo
cd_passa_liner	Confirmação de Limpeza - Sim (1) / Não (0)	Categórico Nominal

Fonte: Elaboração Própria

O conjunto de dados deste estudo possui uma limitação de amostra, típica de modelos de classificação binária, onde ocorre desbalanceamento de dados entre as classes. Verificou-se conforme Figura 13, que o desbalanceamento é de 95,40% para classe 1 e 4,60% para classe 0, situação que muitos algoritmos de aprendizagem de máquina não conseguem tratar, provocando resultados não satisfatórios na realização de previsões pelo modelo. Para resolver situações de desbalanceamento tem-se a opção de aplicar a técnica de subamostragem do conjunto majoritário, sobreamostragem do conjunto minoritário ou uma combinação das duas técnicas. Neste caso, optou-se pelo método SMOTE, técnica de sobreamostragem (*oversampling*), onde se cria novos casos sintéticos associando cada caso da classe minoritária a um de seus k vizinhos mais próximos, escolhidos aleatoriamente, até equalizar os números de classificação. O fato dessa técnica simplesmente não duplicar os dados da classe minoritária, evita a possibilidade de sobreajuste porque muitos dados ficariam idênticos. A sobreamostragem é aplicada somente sobre o conjunto de dados de treinamento para garantir que os dados de teste mostrem o desempenho de como seria a aplicação real.

Figura 13 – Balanceamento de Classes.



Fonte: Elaboração Própria

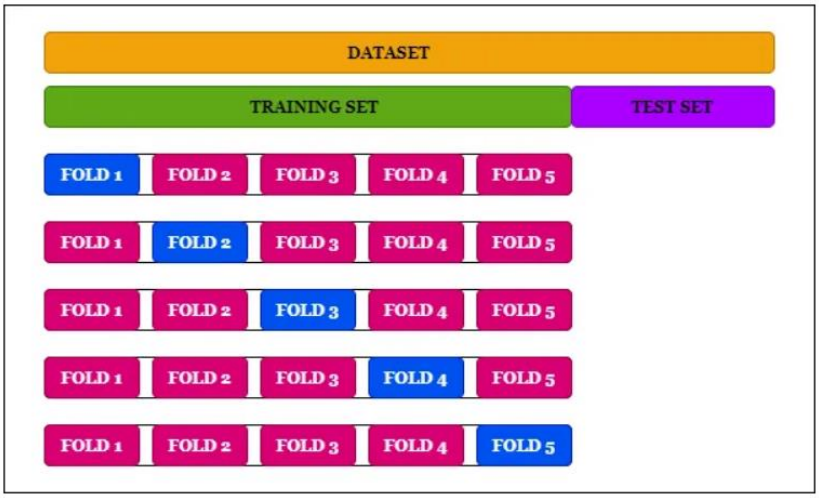
5.3 DIVISÃO DE DADOS

Uma vez que os dados foram preparados para servirem de base ao ML, através dos algoritmos selecionados neste artigo, torna-se necessário a divisão destes dados em 2 grupos de instâncias aleatórias, um para treinamento e outro para teste, este último usado somente após a geração do modelo para colocá-lo a prova. Estes grupos, treino e teste, costumam estar em uma relação percentual de 80-20 ou 70-30 respectivamente, conforme escolha adotada para o estudo, e neste caso, optou-se pela relação 80-20. Uma outra divisão importante, ocorre sobre o grupo treinamento, criando um grupo de validação usado para avaliação de diferentes modelos com diferentes hiperparâmetros ou parâmetros ajustáveis que permitem tunar (customizar) o modelo em busca do seu melhor desempenho.

Para ter uma validação otimizada do treinamento, foi decidido pela utilização da técnica de validação cruzada k-*fold* (Figura 14), o que significa dividir o grupo de treinamento em k partes iguais, e avaliar o melhor desempenho executando o treinamento k vezes, onde a cada execução o grupo de validação assumirá uma das partes e o grupo de treinamento, as demais.

Como a normalização do conjunto de dados foi aplicada apenas para os algoritmos KNN, MLP, Regressão Logística e SVM, passaram a existir 2 conjuntos de dados, um original com as diferentes ordens de grandezas em suas variáveis, e outro normalizado com uma única ordem de grandeza. Desta forma, cada um destes conjuntos de dados foi dividido, gerando seus respectivos grupos de treinamento, validação e teste.

Figura 14 – Validação Cruzada K-*Fold* (5).



Fonte: LEITE, 2020

5.4 TREINAMENTO E OTIMIZAÇÃO DOS MODELOS

Existem métodos que conseguem verificar a melhor performance de um modelo para diversas combinações de hiperparâmetros, funcionando como um ajuste fino do processo. No presente estudo, foi utilizado um método que seleciona e treina o modelo com um número de combinações aleatórias em uma grade de combinações possíveis, obtendo a combinação que apresentou melhor desempenho. A implementação deste método ocorreu através de algoritmo de busca aleatória, sendo definida a quantidade de combinações a serem treinadas através do parâmetro “número de iterações” (n_iter), e quantas partes o conjunto de dados será dividido para aplicação da validação cruzada (Figura 14) através do parâmetro “número de divisões*”* (n_splits). A aplicação da validação cruzada exigiu o emprego de um recurso implementado pela função *pipeline*, recurso este que evitou a utilização dos dados de treinamento com sobreamostragem pelas partes usadas para validação, ou seja, essas partes utilizaram dados de treinamento desbalanceados, seguindo o princípio de que testes e validações devem ser feitos sobre dados que refletem a situação real.

Tabela 3 – Aplicação Método de Algoritmo de Busca Aleatória.

ALGORITMO CLASSIFICADOR	GRADE DE HIPERPARÂMETROS	COMBINAÇÕES				MELHORES ESTIMADORES	TEMPO
		GRADE	ITERAÇÕES	K-FOLDS	AJUSTES		
Árvore de Decisão	"criterion": ['gini', 'entropy'], "splitter": ['best', 'random'], "min_samples_leaf": [1, 2, 4], "min_samples_split": [2, 5, 10], "max_features": ['sqrt'], "max_depth": [3, 5, 7, None]	144	72	10	720	{'decisiontreeclassifier__splitter': 'random', 'decisiontreeclassifier__min_samples_split': 5, 'decisiontreeclassifier__min_samples_leaf': 1, 'decisiontreeclassifier__max_features': 'sqrt', 'decisiontreeclassifier__max_depth': None, 'decisiontreeclassifier__criterion': 'gini'}	0,5 min
Random Forest	"criterion": ['gini','entropy'], "bootstrap": [True], "n_estimators": [50,100,200], "min_samples_leaf": [1, 2, 4], "min_samples_split": [2, 5, 10], "max_features": ['sqrt'], "max_depth": [3, 5, 7]	162	20	10	200	{'randomforestclassifier__n_estimators': 200, 'randomforestclassifier__min_samples_split': 5, 'randomforestclassifier__min_samples_leaf': 4, 'randomforestclassifier__max_features': 'sqrt', 'randomforestclassifier__max_depth': 7, 'randomforestclassifier__criterion': 'entropy', 'randomforestclassifier__bootstrap': True}	3 min
Naive Bayes	"var_smoothing": np.logspace(0,-9, num=100)	100	100	10	1.000	{'gaussiannb__var_smoothing': 1.873817422860383e-07}	0,5 min
K-Nearest Neighbors	list(range(1,31)) ['uniform', 'distance']	60	30	10	300	{'kneighborsclassifier__weights': 'distance', 'kneighborsclassifier__n_neighbors': 2}	1 min
Regressão Logística	"penalty": ['l1','l2'], "solver": ['lbfgs','liblinear'], "C": np.logspace(-5, 8, 15), "max_iter": [500]	120	25	10	250	{'logisticregression__solver': 'liblinear', 'logisticregression__penalty': 'l2', 'logisticregression__max_iter': 500, 'logisticregression__C': 1e-05}	2 min
Support Vector Machine	"C": [0.1, 1, 10, 100], "gamma": [0.1, 0.01, 0.001]	12	5	10	50	{'svc__gamma': 0.1, 'svc__C': 100}	77 min
Multilayer Perceptron	"hidden_layer_sizes": [(50,50,50), (50,100,50), (100,)], "activation": ['tanh', 'relu','logistic'], "solver": ['sgd', 'adam'], "alpha": [0.0001, 0.05], "max_iter": [100], "early_stopping": [True], "learning_rate": ['constant','adaptive']	96	10	10	100	{'mlpclassifier__solver': 'adam', 'mlpclassifier__max_iter': 100, 'mlpclassifier__learning_rate': 'adaptive', 'mlpclassifier__hidden_layer_sizes': (50, 50, 50), 'mlpclassifier__early_stopping': True, 'mlpclassifier__alpha': 0.0001, 'mlpclassifier__activation': 'relu'}	35 min

Fonte: Elaboração Própria

Na Tabela 3 pode-se observar a grade de hiperparâmetros que foi considerada para as combinações possíveis, o número de iterações para as combinações aleatórias, o número de divisões (k-*folds*) e os melhores estimadores, aqueles que cada classificador apresentou melhor desempenho. Também é apresentado o custo computacional para treinar cada modelo, através da variável de tempo.

Após o processamento, foi aplicada a validação dos melhores estimadores com o resultado sendo observado na Tabela 4, através dos valores de acurácia média e desvio padrão.

Tabela 4 – Validação de melhores estimadores

ALGORITMO CLASSIFICADOR	ACURÁCIA MÉDIA	DESVIO PADRÃO
Árvore de Decisão	99,92%	0,12%
Random Forest	99,25%	0,42%
Naive Bayes	95,27%	0,98%
K-Nearest Neighbors	99,93%	0,14%
Regressão Logística	89,75%	1,40%
Support Vector Machine	95,72%	0,76%
Multilayer Perceptron	98,49%	1,10%

Fonte: Elaboração Própria

A acurácia média e o desvio padrão apresentados demonstram o quanto cada modelo está generalizando, indicando melhor performance quanto maior for a acurácia e menor for o desvio padrão.

Observou-se que de uma maneira geral os classificadores tiveram um bom desempenho com acurácia média alta e desvio padrão baixo. A acurácia média dos modelos ficou acima de 89,5%, ou seja, de cada 100 previsões todos os modelos acertaram mais de 89, destacando-se os modelos baseados em Árvore de Decisão, *Random Forest*, KNN e *Multilayer Perceptron*, com uma taxa de acerto acima de 98 para cada 100. O desvio padrão por sua vez, apresentou uma taxa de dispersão dos dados em torno da média, abaixo de 1,5%, o que significa tendência para os dados estarem próximo do valor esperado, destacando-se os modelos baseados em Árvore de Decisão e KNN, com valores abaixo de 0,15%. Concluiu-se então que os indicadores sinalizaram para o prosseguimento do estudo em busca do melhor classificador dentre os 7 modelos criados.

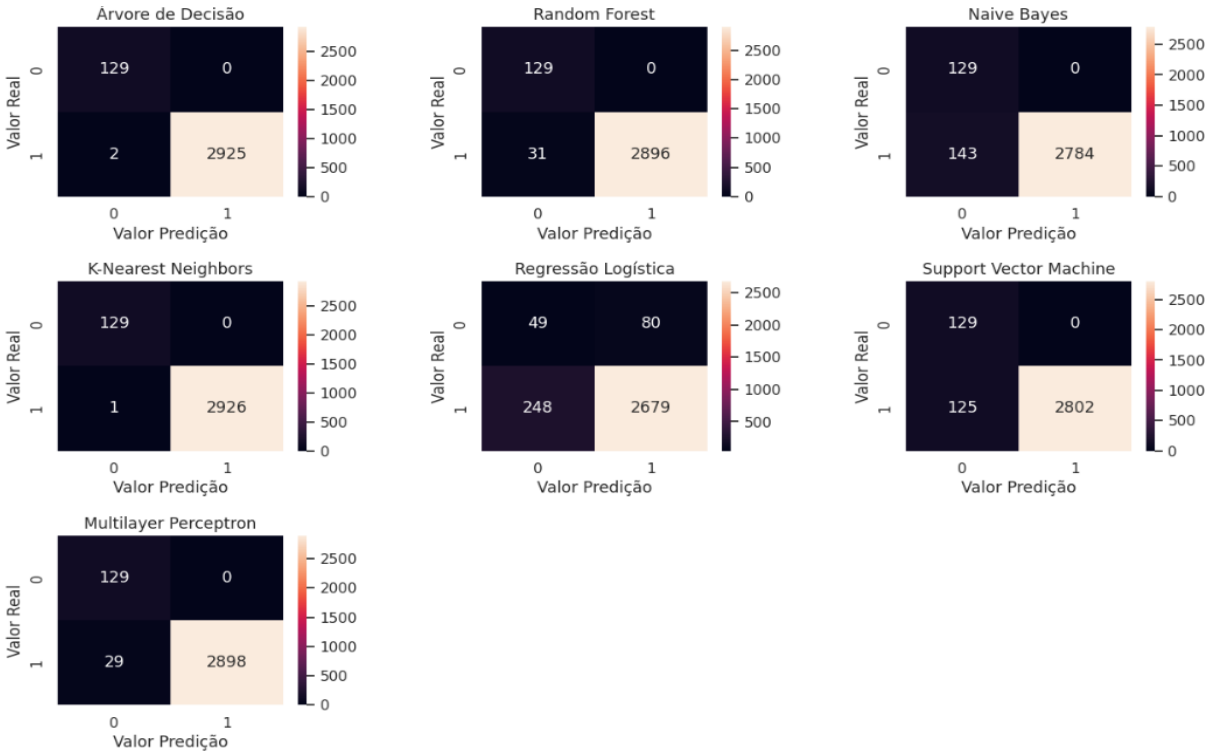
5.5 AVALIAÇÃO DE RESULTADOS

Os modelos de classificação treinados e validados com os melhores estimadores, identificados a partir da otimização dos hiperparâmetros, foram testados mediante a previsão de classes sobre um conjunto de dados desconhecido, simulando situações reais. Estes dados representam 20% dos dados coletados para este estudo, conforme explicado na seção 5.3.

A avaliação de desempenho dos modelos na tarefa de prevê corretamente as classes sobre dados novos, ocorreu a partir da medição dos resultados através de diversas métricas de classificação, conforme será descrito a seguir.

A primeira fonte de informação para medir a qualidade dos modelos de classificação binária treinados foi a matriz de confusão (Figura 15), apresentando a quantidade de classes positivas e negativas reais e previstas. Destacaram-se como modelos mais equilibrados, os classificadores Árvore de Decisão e KNN, praticamente não errando suas previsões, considerando apenas uma falha na classe positiva com um percentual não superior a 0,07%. Em contrapartida, destacaram-se como os modelos que mais erraram, o classificador SVM com uma taxa de 14,32% da classe positiva, e o classificador Regressão Logística com uma taxa de 8,47% da classe positiva e 62,02% da classe negativa.

Figura 15 – Matriz de Confusão



Fonte: Elaboração Própria

Assim como na validação dos modelos durante o treinamento, constatou-se também uma alta taxa de acertos sobre o conjunto de dados de teste, através do cálculo de acurácia nas previsões conforme Tabela 5, o que sinaliza para uma generalização já que os modelos tiveram o mesmo bom desempenho dos treinos nas previsões com dados novos. Com isso, pode-se dizer que o resultado desta métrica não caracterizou sobreajuste (*overfitting*) nos modelos, os validando. Os modelos que mais uma vez se sobressaíram no acerto de previsões foram os classificadores Árvore de Decisão e KNN com uma taxa de 100%, seguidos do Random Forest e MLP com 99%.

Tabela 5 – Métricas de Classificação

ALGORITMO CLASSIFICADOR	ACURÁCIA↑	PRECISÃO↑	RECALL↑	F1-SCORE↑	MCC↑	LOG-LOSS↓	AUPRC↑	AUROC↑
Árvore de Decisão	100%	100%	100%	100%	0,99	0,02	1,0000	0,9997
Random Forest	99%	100%	99%	99%	0,89	0,06	1,0000	0,9996
Naive Bayes	95%	100%	95%	97%	0,67	1,22	0,9996	0,9896
K-Nearest Neighbors	100%	100%	100%	100%	1,00	0,01	1,0000	0,9998
Regressão Logística	89%	97%	92%	94%	0,20	0,68	0,9951	0,8948
Support Vector Machine	96%	100%	96%	98%	0,70	0,11	0,9993	0,9844
Multilayer Perceptron	99%	100%	99%	100%	0,90	0,04	0,9998	0,9948

Fonte: Elaboração Própria

Para o negócio aqui estudado, a classe positiva tem maior importância que a classe negativa devido ao grande impacto da não aplicação do *setup* quando necessário, afetando a qualidade do produto seguinte na linha de produção, o que pode gerar prejuízos com perdas de produção ou reclamação de cliente, indicador monitorado como diferencial de mercado nesta indústria. Por esse fato, optou-se pela escolha das métricas de classificação precisão, revocação (*recall*) e f1-*score*.

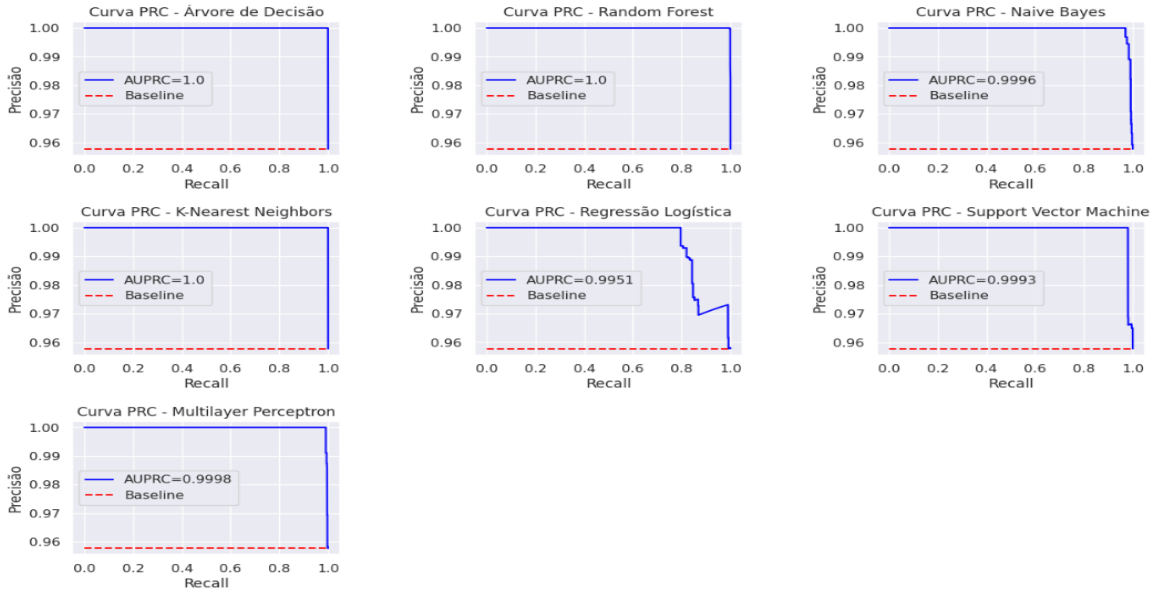
Verificou-se que os modelos tiveram bons resultados tanto na precisão quanto na revocação apenas apurando os altos valores apresentados na métrica f1-*score*. Isso porque essa métrica calcula a média harmônica entre precisão e revocação, o que significa ter bons resultados apenas quando os valores forem altos para as 2 métricas, pois a média harmônica sempre resulta em valor próximo aos valores mais baixos envolvidos. Os melhores resultados de f1-*score* (Tabela 5) ficaram com os modelos de classificadores baseados em Árvore de Decisão, KNN e MLP obtendo

100% de desempenho, seguidos de *Random Forest* com 99%, SVM com 98%, *Naive Bayes* com 97% e Regressão Logística com 94%.

A apuração da métrica revocação de forma isolada se torna interessante por porque nela os falsos negativos (“não passar *setup* quando é necessário”) são considerados mais prejudiciais que os falsos positivos (“passar *setup* quando não é necessário”), o que corrobora com a expectativa do negócio conforme já explicado. Esta métrica confirmou o quanto de valores positivos foram detectados no conjunto de teste por cada modelo (Tabela 5), com destaque para os classificadores baseados em Árvore de Decisão e KNN com uma taxa de 100%, seguidos pelo *Random Forest* e MLP com 99%, SVM com 96%, *Naive Bayes* com 95% e Regressão Logística com 92%.

Ainda através dessas métricas, outra forma utilizada para avaliar o desempenho dos modelos foi observando a área abaixo da curva precisão-revocação (*area under the precision recall curve*), chamada de métrica AUPRC. A grande vantagem da curva como métrica é que demonstra o desempenho do modelo em vários ou todos os limiares possíveis de classificação, e calculando sua área obtêm-se o resultado desta curva em um único número, considerado melhor quanto mais próximo de 1 tiver. Os resultados obtidos com a métrica AUPRC (Figura 16) foram de 1 para os classificadores baseados em Árvore de Decisão, Random Forest e KNN, de 0,9998 para MLP, 0,9996 para Naive Bayes, 0,9993 para SVM e 0,9951 para Regressão Logística.

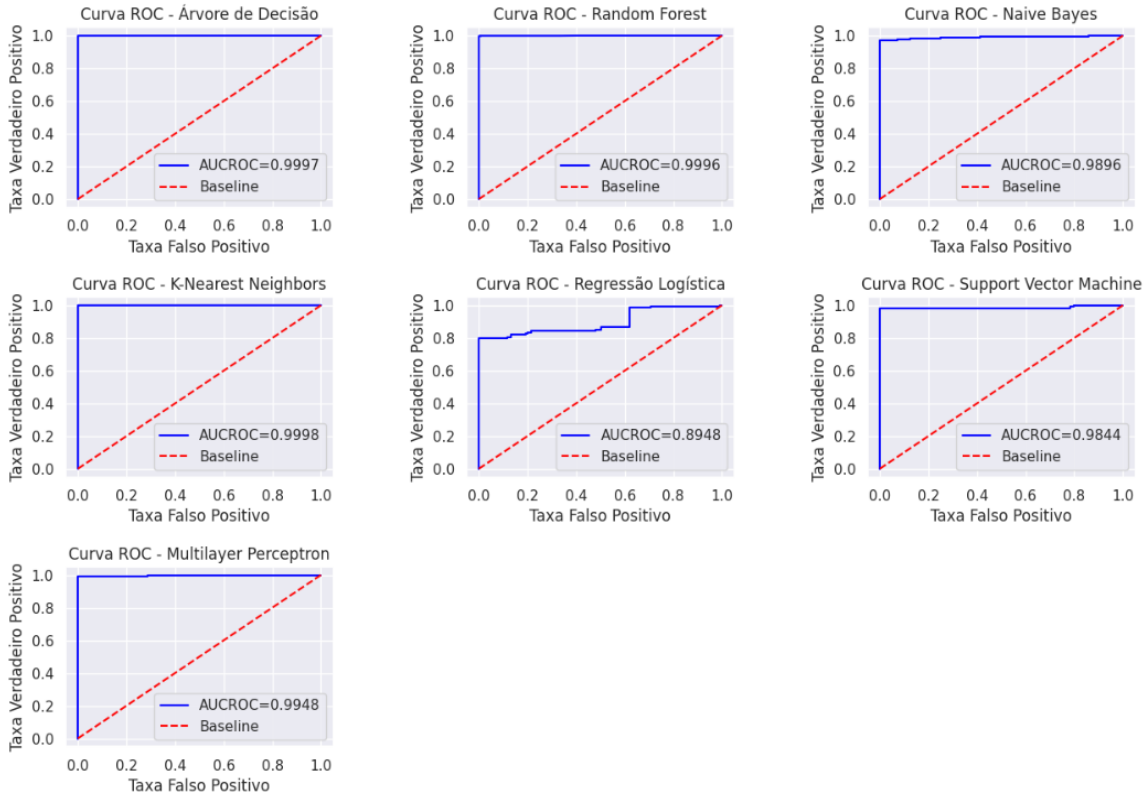
Figura 16 – Curva PRC, Métrica AUPRC



Fonte: Elaboração Própria

Uma outra métrica baseada em curva é a AUC ou AUROC, área sob a curva ROC (*Receiver Operating Characteristic*), onde é traçado um *trade-off* entre Taxa de Verdadeiro Positivo e Taxa de Falso Positivo com valores entre 0 e 1, onde quanto maior o valor, melhor é a capacidade do modelo em identificar se uma classe deve ser positiva (“aplicar *setup*”) ou negativa (“não aplicar *setup*”). Nesta métrica observou-se (Figura 17) melhor desempenho no classificador baseado em KNN com AUC de 0,9998, seguido do classificador baseado em Árvore de Decisão com AUC de 0,9997, *Random Forest* com 0,9996, MLP com 0,9948, *Naive Bayes* com 0,9896, SVM com 0,9844 e Regressão Logística com 0,8948.

Figura 17 – Curva ROC, Métrica AUC



Fonte: Elaboração Própria

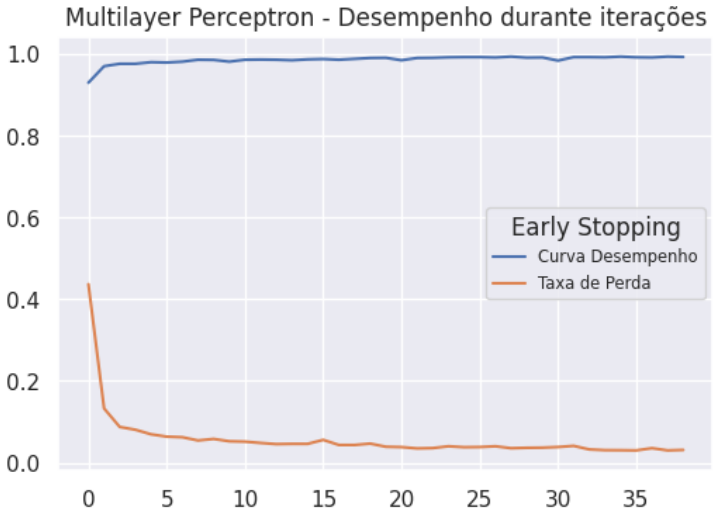
Foi utilizado neste estudo, uma métrica que resume todas as possibilidades de uma classificação binária, o MCC, que retorna um valor no intervalo entre -1 e 1, onde 1 é considerado uma “classificação perfeita”, 0 uma “classificação aleatória”, um “chute”, e -1 uma “classificação errada”. Observou-se o modelo com melhor resultado nesta métrica (Tabela 5), o classificador baseado em KNN com valor 1, seguido do classificador baseado em Árvore de Decisão com 0,99, MLP com 0,90,

Random Forest com 0,89, SVM com 0,70, *Naive Bayes* com 0,67 e Regressão Logística com apenas 0,2.

Para verificar o quanto as previsões estavam próximas dos valores reais, avaliou-se a métrica *log loss*, uma função de custo que retorna um valor entre 0 e infinito conforme a performance probabilística do modelo ao errar previsões. Quanto maior for a probabilidade indicada para uma previsão ser certa quando não era, maior o valor do *log loss,* penalizando o modelo pelo grau da falha. Um modelo confiável terá um *log loss* próximo a 0, e neste estudo os modelos que se mostraram mais confiáveis (Tabela 5) foram os classificadores baseados em KNN com 0,01, em Árvore de Decisão com 0,02 e em MLP com 0,04. Os classificadores baseados em *Naive Bayes* e Regressão Logística se destacaram como os menos confiáveis, com *log loss* de 1,22 e 0,68, respectivamente.

Complementarmente as métricas de classificação apuradas, foi processado uma avaliação de confiabilidade do classificador baseado em rede neural MLP conforme gráfico da Figura 18. Observou-se a identificação da melhor curva de desempenho e taxa de perda (*loss*) ao longo do treinamento, conforme atuação do hiperparâmetro *early stopping que* interrompe o processo de construção do modelo quando este deixa de melhorar para evitar sobreajuste (*overfitting*).

Figura 18 – Curva Desempenho MLP



Fonte: Elaboração Própria

6. CONSIDERAÇÕES FINAIS

Este artigo demonstrou como a aprendizagem de máquina pode ser útil na construção de modelos que apoiem decisões importantes para o processo produtivo de uma indústria, indicando neste caso, quando a produção de uma indústria têxtil deve ser interrompida para ajustes de parâmetros de especificação da máquina que processa o tratamento químico dado aos tecidos produzidos, lonas para reforço de pneus.

Ficou claro a importância da preparação dos dados com aplicação de todas as técnicas necessárias para atender aos requisitos que possibilitam tirar o melhor de cada algoritmo, já que cada classificador se comporta de maneira mais ou menos adequada conforme o contexto que lhe é apresentado.

Outro aspecto abordado visto como fundamental para tirar proveito do melhor de cada algoritmo foi a utilização de métodos para validar o modelo ainda na fase de treinamento com dados desconhecidos, simulando uma série de combinações de parâmetros até chegar naquela considerada mais otimizada para obter um melhor desempenho para cada modelo.

Observou-se nos classificadores baseados em SVM e redes neurais MLP uma demanda por mais recursos computacionais para o treinamento e validação de seus modelos, o que influenciou a decisão de ter menos iterações. Assim, sugere-se então um aumento destes recursos computacionais em trabalhos futuros com a finalidade de obter maior ganho de performance.

Os resultados finais obtidos apontaram para os modelos de classificação baseados em KNN, Árvore de Decisão, redes neurais MLP e *Random Forest* pelos excelentes desempenhos observados, considerando que em todas as métricas apuradas obtiveram altas taxas de acertos gerais, e altas taxas de acertos e detecção de classes positivas, mais relevante neste estudo, inclusive na média ponderada dos limiares possíveis. Como também foi constatado baixas taxas de erros em suas previsões. A exceção no desempenho destes 4 classificadores ficou por conta da métrica MCC, onde KNN e Árvore de Decisão tiveram um destaque muito maior que os modelos MLP e *Random Forest*.

Concluiu-se então que a escolha do melhor modelo ficaria entre os classificadores KNN e Árvore de Decisão, e a decisão final passaria pela avaliação das vantagens e desvantagens dos mesmos conforme descrito na seção 3. Confrontando-os

pelas vantagens, destacou-se o fato do algoritmo KNN ser muito rápido pela característica de fazer predição baseando-se em memorização dos dados de treino, a chamada aprendizagem por analogia, e pelas desvantagens, destacou-se o fato do algoritmo Árvore de Decisão ser instável a modificações, podendo alterar suas previsões a partir de uma leve mudança nos dados, sendo assim, este estudo indicou como vencedor' o modelo ML baseado no classificador KNN.

A partir dos resultados satisfatórios deste trabalho, será desenvolvido uma aplicação para colocar o modelo ML vencedor em produção, como uma ferramenta de apoio a tomada de decisão na área produtiva da indústria têxtil selecionada para este estudo, sendo vislumbrado a expansão para outras unidades da corporação a qual pertence.

7. REFERÊNCIAS

BENTO, Fábio; VASSALO, Raquel; SAMATELO, Jorge. **Pipeline de Dados para Detecção de Anomalias em Vídeos de Cena Única**. Sociedade Brasileira de Automática – SBA, 2021. Disponível em < https://www.sba.org.br/open_journal_systems/index.php/sbai/article/view/2687/2227> . Acesso em: 18 jun. 2023.

BERTAN, Erica. O que é Precisão e Revocação. **Medium**, 2020. Disponível em: < https://medium.com/computando-arte/o-que-%C3%A9-precis%C3%A3o-e-revoca%C3%A7%C3%A3o-b0b991b67cde>. Acesso em: 18 jun. 2023.

CARDOSO, Sergio. **Estudo das Propriedades Mecânicas e dos Mecanismos de Fratura de Fibras Sintéticas do Tipo Náilon e Poliéster em Tecidos de Engenharia**. 2009, 151 f. Tese (Doutorado em Tecnologia Nuclear – Materiais) – Universidade de São Paulo, São Paulo, 2009.

CARDOSO, Isaac. **Técnicas de Otimização e Métricas de Avaliação Aplicadas a Machine Learning**. 2022, 52 f. Monografia (Bacharel em Ciência da Computação) – Instituto Federal de Educação, Ciência e Tecnologia Goiano, Rio Verde, 2022.

Como fazer um projeto de Machine Learning bem-sucedido. **ILUMEO**, 2022. Disponível em: <https://ilumeo.com.br/todos-posts/2022/08/15/como-fazer-um-projeto-de-machine-learning-bem-sucedido>. Acesso em: 08 abr. 2023.

FILHO, Mario. Guia Completo da Log Loss (Perda Logarítmica) em Machine Learning. **Mario Filho | Machine Learning**, 2023. Disponível em: <https://mariofilho.com/guia-completo-da-log-loss-perda-logaritmica-em-machine-learning/>. Acesso em: 08 abr. 2023.

GODOY, Daniel. Uma explicação visual para uma função de custo “binary cross-entropy” ou “log loss”. **Medium**, 2018. Disponível em: < https://medium.com/ensina-ai/uma-explica%C3%A7%C3%A3o-visual-para-fun%C3%A7%C3%A3o-de-custo-binary-cross-entropy-ou-log-loss-eaee662c396c>. Acesso em: 19 jun. 2023.

GUALBERTO, Arnaldo. O que não te contam sobre métricas de classificação binária. **Medium**, 2020. Disponível em: https://medium.com/@arnaldog12/o-que-n%C3%A3o-te-contam-sobre-m%C3%A9tricas-de-classifica%C3%A7%C3%A3o-bin%C3%A1ria-d1834e385402>. Acesso em: 18 jun. 2023.

MARTIN, Damien. How to do cross-validation when upsampling data. **Stacked Turtles,** 2019. Disponível em: < https://kiwidamien.github.io/how-to-do-cross-validation-when-upsampling-data.html>; Acesso em: 25 abr. 2023.

IGNACIO, Lucas. **Aprendizado de máquina: da teoria à aplicação**. 2021, 80 f. Monografia (Bacharel em Matemática) – Universidade Federal Fluminense, Volta Redonda, 2021.

INAZAWA, Pedro; *et al.* PROJETO VICTOR, **REVISTA COMPUTAÇÃO BRASIL**, Porto Alegre, 39, 1. ed., p. (19 a 24), 2019.

LEITE, Rodrigo. Introdução a Validação-Cruzada: K-Fold. **Medium**, 2020. Disponível em: <https://drigols.medium.com/introdu%C3%A7%C3%A3o-a-valida%C3%A7%C3%A3o-cruzada-k-fold-2a6bced32a90>. Acesso em: 19 abr. 2023.

MARCONDES, Alexandre. Tradeoff entre precisão e recall: explorando o output de seu modelo de classificação. **Medium**, 2021. Disponível em: < https://medium.com/datarisk-io/tradeoff-entre-precis%C3%A3o-e-recall-explorando-o-output-de-seu-modelo-de-classifica%C3%A7%C3%A3o-bd1694111033#:~:text=O%20tradeoff%20entre%20precis%C3%A3o%20e%20recall,-Observando%20as%20equa%C3%A7%C3%B5es&text=Um%20aumento%20na%20precis%C3%A3o%2C%20em,pode%20acabar%20apontando%20poucos%20positivos.>. Acesso em: 18 jun. 2023.

MENDES, Darcy. Fatores Humanos: Gerenciando Falhas Humanas. **SEGURANÇA DO TRABAHO – TEM SEGURANÇA**. Sorocaba, 27 abr. 2014. Disponível em: <https://temseguranca.com/fatores-humanos-gerenciando-falhas-humanas/>. Acesso em: 06 jun. 2023.

MOREIRA, Sandro. Redes Neurais Perceptron Multicamadas. **Medium**, 2018. Disponível em: < https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9>. Acesso em: 26 jun. 2023.

NETO, Alfredo; BONINI, Carolina. Redes Neurais Artificiais: Apresentação e Utilização do Algoritmo Perceptron em Biossistemas. **Revista Brasileira de Engenharia de Biossistemas**, Tupã, v.4, n.2, p. 87-95, Mai/Ago. 2010. Disponível em: <http://seer.tupa.unesp.br/index.php/BIOENG/article/view/95/95>. Acesso em: 31 mar. 2023.

O que difere Inteligência Artificial, Machine Learning e Ciência de Dados. **ILUMEO**, 2020. Disponível em: <https://ilumeo.com.br/todos-posts/2020/10/05/o-que-difere-inteligencia-artificial-machine-learning-e-ciencia-de-dados>. Acesso em: 23 mar. 2023.

PACHECO, André. Medida de desempenho de classificadores - Parte 1. **Computação Inteligente**, 2018. Disponível em: <http://computacaointeligente.com.br/conceitos/medidas-classificadores-1/>. Acesso em: 17 jun. 2023.

PACHECO, André. Medida de desempenho de classificadores - Parte 2. **Computação Inteligente**, 2018. Disponível em: <http://computacaointeligente.com.br/conceitos/medidas-classificadores-2/>. Acesso em: 17 jun. 2023.

PAIXÃO, Gabriela; *et al*. **Machine Learning na Medicina: Revisão e Aplicabilidade**. Sociedade Brasileira de Cardiologia – SBC, 2022. Disponível em <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8959062/pdf/0066-782X-abc-118-01-0095.pdf>. Acesso em: 21 mar. 2023.

QUINTAS, José. **Análise através da curva ROC: que ferramentas utilizar?**. 2020, 96 f. Dissertação (Mestrado em Bioinformática) – Universidade do Minho, Braga, 2020.

SILVA, Michel. **O Pré-Processamento em Mineração de Dados como método de suporte à modelagem algorítmica**. 2014, 84 f. Dissertação (Mestrado em Modelagem Computacional de Sistemas) – Universidade Federal do Tocatins, Palmas, 2014.

SOUZA, Áurea. Coeficiente de Correlação de Pearson e Coeficiente de correlação de Spearman. O que medem e em que situações devem ser utilizados?, **CORREIO DOS AÇORES**, Ilha de São Miguel, 1, p. (1 a 1), 2019.

TITO, Nathália. Como escolher o número de neurônios na camada oculta de uma Rede Neural?. **Medium**, 2020. Disponível em: https://nathaliatito.medium.com/como-escolher-o-n%C3%BAmero-de-neur%C3%B4nios-na-camada-oculta-da-sua-rede-neural-93d591cb838d>. Acesso em: 19 jun. 2023.

Sistema FIEB | SENAI CIMATEC
PELO FUTURO DA INOVAÇÃO

CENTRO UNIVERSITÁRIO SENAI CIMATEC

CURSO DE ESPECIALIZAÇÃO LATO SENSU EM DATA SCIENCE E ANALYTICS

ATA DE APRESENTAÇÃO DE PROJETO FINAL DE CURSO

Ata de apresentação do Projeto Final de Curso **"APRENDIZAGEM DE MÁQUINA APLICADA NA PREVISÃO DE DEMANDA POR SETUP DO PROCESSO DE TRATAMENTO QUÍMICO DE TECIDO NA INDÚSTRIA TÊXTIL"**, submetido pelo aluno **Paulo Sergio Nascimento de Jesus,** como parte dos requisitos para obtenção do Certificado de Especialista em *Data Science e Analytics* pelo Centro Universitário SENAI CIMATEC, às 18:00h do dia 31 de agosto de 2023. Reuniu-se no CIMATEC, a Banca Examinadora designada pela Coordenação de curso, constituída por Prof. Dr. Oberdan Rocha Pinheiro e Dr. Taniel Silva Franklin. A coordenadora do curso Patricia Freitas Tourinho deu início aos trabalhos e a exposição foi realizada pelo estudante dentro do prazo de tempo estabelecido. Ao final da apresentação a banca reuniu-se atribuindo a seguinte nota: 8,5 **(oito e cinco).**

A banca de avaliadores decidiu pela:

(x) Aprovação do trabalho
Caberá ao aluno apresentar em no máximo em 30 (trinta) dias a contar da data de assinatura desta Ata, uma cópia do trabalho em PDF com restrição de edição. A Ata de Apresentação do Projeto Final de Curso deve ser digitalizada e inserida na terceira página do PFC.

( ) Reprovação do trabalho
O aluno terá que se matricular novamente no TCC – Trabalho de Conclusão de Curso e ser submetido a uma banca avaliadora no semestre seguinte**.**

As ações consequentes ao status de Aprovação deverão obedecer ao prazo proposto acima sob pena do parecer final ser modificado para o status de Reprovado automaticamente e sem possibilidade de recurso.

Sistema FIEB | SENAI CIMATEC
PELO FUTURO DA INOVAÇÃO

Para constar, lavrou-se a presente ata que vai assinada por todos os membros da Banca. Por estarem cientes de suas obrigações estão de acordo com os termos desse documento:

Salvador, 31 de agosto de 2023

Assinado eletronicamente por:
Patricia Freitas Tourinho
CPF: ***.733.265-**
Data: 01/09/2023 10:55:07 -03:00

Profª. Esp. Patricia Freitas Tourinho
Coordenadora do Curso

Assinado eletronicamente por:
Oberdan Rocha Pinheiro
CPF: ***.073.705-**
Data: 01/09/2023 08:51:15 -03:00

Prof. Dr. Oberdan Rocha Pinheiro Professor Orientador

Electronically signed by:
Taniel Silva Franklin
CPF: ***.769.875-**
Date: 9/1/2023 9:01:10 AM -03:00

Prof. Dr. Taniel Silva Franklin – Professor convidado

MANIFESTO DE ASSINATURAS

Código de validação: QYHH8-E2GBC-P4YGH-KUKCE

Esse documento foi assinado pelos seguintes signatários nas datas indicadas (Fuso horário de Brasília):

✔ Oberdan Rocha Pinheiro (CPF ***.073.705-**) em 01/09/2023 08:51 - Assinado eletronicamente

Endereço IP 179.105.130.71	Geolocalização Lat: -12,941857 Long: -38,339485 Precisão: 15 (metros)
Autenticação Login	oberdan.pinheiro@fieb.org.br (Verificado)

Tee3gRhsX7kRJLCM2MXwCdLWEGJ6rZNjR9QLbZwlzKI= SHA-256

✔ Taniel Silva Franklin (CPF ***.769.875-**) em 01/09/2023 09:01 - Assinado eletronicamente

Endereço IP 189.106.41.59	Geolocalização Lat: -12,971569 Long: -38,411351 Precisão: 1761 (metros)
Autenticação Email verificado	taniel.franklin@fieb.org.br

ZSJ2VoSfQZR4LkCiX2JPkmfLZFjTrTRD97MGE9K5lkQ= SHA-256

✔ Patricia Freitas Tourinho (CPF ***.733.265-**) em 01/09/2023 10:55 - Assinado eletronicamente

Endereço IP	Geolocalização
179.105.140.6	Lat: -13,008565 Long: -38,484320 Precisão: 21 (metros)

Autenticação	patricia.tourinho@fieb.org.br
Email verificado	

Drxkp76MQ5qsvPZlWE3TZltAfTGcU0LQZmk5+kuwfAU= SHA-256

Para verificar as assinaturas, acesse o link direto de validação deste documento:

https://assinatura.senaibahia.com.br/validate/QYHH8-E2GBC-P4YGH-KUKCE

Ou acesse a consulta de documentos assinados disponível no link abaixo e informe o código de validação:

https://assinatura.senaibahia.com.br/validate