Machine Learning Algorithms Applied on Classification of Processes for Conciliation on Brazilian Labour Judiciary*

Filipe M. C. Barros[1], Cleison D. Silva[1], Igor R. M. Silva[2], Victor S. Martins[1], Antonio J. S. Araújo[1]

[1] Universidade Federal do Pará (UFPA)
Programa de Pós-graduação em Computação Aplicada (PPCA)
Campus Universitário Tucuruí - Tucuruí/PA - Brazil
{filipe.barros,antonio.araujo}@tucurui.ufpa.br
{cleison,victorsm}@ufpa.br

[2] Universidade Federal do Rio Grande do Norte (UFRN)
Campus Universitário Lagoa Nova - Natal/RN - Brazil
igor.silva@ufrn.br

Abstract. The Labour Judiciary ensures protection and justice in labour relations, resolving conflicts such as unfair dismissals and wage delays. Artificial intelligence emerges to expedite legal activities, assisting in dealing with the increasing case load in the Judiciary over the past years. In labor dispute resolution, conciliation is a recommended solution, offering speed and cost reduction. In this sense, this study proposes to evaluate models to predict the success of labor cases being resolved through conciliation. The dataset used to generate the models considered in this study consists of initial petitions from cases extracted from the *Processo Judiciário Eletrônico* (PJe) maintained by the *Tribunal Regional do Trabalho da 8ª Região*. Pre-processing steps were performed on these documents, including the removal of accents, special characters, numerals, punctuation, stopwords, conversion of text to lowercase, stemming, and tokenization. The next step was text vectorization using the Term Frequency-Inverse Document Frequency (TF-IDF) for model generation. For our analysis, three machine learning algorithms were taken into account: Support Vector Machines (SVM), logistic regression, and decision trees. Additionally, a boosted tree model (XGBoost) was also generated. Based on the analysis conducted, the SVM with RBF kernel yielded better results, achieving an accuracy of 83% and an F1-Score of 82%, with a Matthews Correlation Coefficient (MCC) of 0.66 and an Area Under the ROC Curve (AUC) of 0.83.

Keywords: Labour Justice · Conciliation · Term Frequency-Inverse Document Frequency · Support Vector Machines · Logistic Regression · Decision Tree

* 20th Encontro Nacional de Inteligência Artificial e Computacional, ENIAC 2023.

1 Introdução

A Justiça do Trabalho é responsável por julgar as questões relacionadas aos direitos trabalhistas, atuando como uma forma de garantir a proteção dos trabalhadores e a justiça nas relações trabalhistas. No Brasil, esta ramificação do Poder Judiciário tem como objetivo principal solucionar conflitos que envolvem as relações de trabalho, tais como demissões sem justa causa, atraso de salários, horas extras não pagas, acidentes de trabalho, entre outros. Além disso, essa especialidade da justiça também tem a função de fiscalizar e garantir o cumprimento das normas trabalhistas, aplicando multas e penalidades em casos de descumprimento da legislação.

A carga de processos nos tribunais vem aumentando cada vez mais, interferindo diretamente no desempenho e na execução das atividades nas cortes. Segundo o Tribunal Superior do Trabalho, o tempo médio de tramitação processual nas Varas do Trabalho tem duração de oito meses até que seja proferida a sentença [15], o que revela um verdadeiro gargalo no fluxo executado pela justiça do trabalho. Tendo em vista a necessidade de soluções para esta situação, passou-se a buscar auxílio na inteligência artificial para tratativa documental, através do uso de ferramentas e técnicas de processamento de linguagem natural (PLN), promovendo uma mudança significativa no modo como são realizadas as atividades no jurídico.

Uma das conclusões possíveis para um processo trabalhista é a conciliação, que consiste em um meio alternativo de resolução de conflitos, em que as partes envolvidas (o empregado e o empregador) se reúnem com um conciliador designado pelo tribunal para tentar chegar a um acordo amigável e evitar a continuação do processo judicial.

Dessa forma, a resolução de um processo por meio da conciliação entre as partes é considerado um dos caminhos mais prudentes e adequados para se resolver uma causa trabalhista. Além de oferecer celeridade na resolução do problema, ajuda a diminuir custos relacionados ao processo, tanto para as partes envolvidas quanto para o serviço público, mostrando se ainda como um meio pacífico de solução do litígio.

Para se alcançar estes resultados, o Poder Judiciário do Trabalho conta com os Centros Judiciários de Métodos Consensuais de Solução de Disputas (CEJUSCs), os quais são responsáveis pela realização das sessões e audiências de conciliação e mediação de processos em qualquer fase ou instância, inclusive naqueles pendentes de julgamento perante o Tribunal Superior do Trabalho. Por intermédio das ações conciliadoras promovidas pelos CEJUSCs, as partes envolvidas em um processo trabalhista têm a oportunidade de resolver seus conflitos em um ambiente neutro de modo a preservar um relacionamento anterior entre elas [3].

Atualmente os CEJUSCs funcionam de modo incipiente em termos de emprego de soluções tecnológicas para o apoio na execução de suas atividades, contando apenas com o trabalho de pessoas que agem como agentes mediadores no processo de conciliação. Tomando como base o valor da conciliação no fluxo de um processo trabalhista, percebe se a importância da utilização de ferramen-

tas da tecnologia da informação para apoiar a tomada de decisão no contexto de resolução de conflitos na justiça do trabalho.

Este trabalho, portanto, orientar-se-á no sentido de realizar um estudo avaliativo de modelos que sejam capazes de prever a chance de sucesso de um processo trabalhista ser resolvido em audiência de conciliação. Para tanto, buscou-se analisar e avaliar as características textuais de petições iniciais presentes em processos trabalhistas, tomando como base de dados um *dataset* formado por petições iniciais de processos mantidos pelo Tribunal Regional do Trabalho da 8ª Região - Pará/Amapá (TRT 8).

O restante deste trabalho está organizado da seguinte maneira: na Seção 2 são apresentados os trabalhos relacionados que apoiaram as análises realizadas neste estudo. A seção 3 apresenta as etapas envolvidas na metodologia de trabalho aplicada, detalhando o método proposto. A seção 4 descreve as métricas selecionadas para o estudo comparativo dos modelos gerados. Na seção 5 é discutido como as configurações aplicadas em cada modelo refletiu nos dados obtidos, consolidando os resultados. Finalmente, a última seção conclui qual modelo obteve o melhor desempenho em relação as métricas consideradas, traçando diretrizes para a evolução do estudo em trabalhos futuros.

2 Trabalhos Relacionados

O processamento de linguagem natural (PLN) tem se tornado cada vez mais popular na comunidade jurídica, especialmente no setor público, devido ao potencial de utilizar dados não estruturados presentes em documentos e registros. Os avanços das pesquisas nessa área impulsionaram a aplicação de técnicas de aprendizado de máquina e outras tecnologias no campo jurídico, revelando a importância desses estudos para criar ferramentas que venham auxiliar as atividades realizadas no setor público.

No que diz respeito às pesquisas desenvolvidas nesta área, é relevante mencionar o estudo de [10], onde foi produzido um conjunto de dados composto por petições do sistema processual do Ministério Público do Estado do Paraná (PROMP), contendo documentos compreendidos pelo período de 2016 a 2019. Por meio de abordagens mais simples, como o uso do *Term Frequency–Inverse Document Frequency* (TF-IDF), ficou evidente a necessidade de um pré-processamento mais extenso para alcançar bons resultados, enquanto que abordagens orientadas à semântica, como a utilização de *word embeddings*, necessitam apenas de um pré-processamento mínimo para serem aplicadas de maneira eficiente.

O TF-IDF é uma técnica estatística que avalia a relevância de um termo em um documento dentro de um conjunto de documentos. É composto por duas medidas principais: a frequência do termo (TF), que conta quantas vezes o termo aparece no documento, e a frequência inversa do documento (IDF), que mede a raridade do termo no *corpus*. *Corpus* pode ser definido como um conjunto de textos ou dados linguísticos cuidadosamente coletados e preparados para treinar e avaliar modelos de processamento de linguagem natural [9].

O TF é calculado dividindo o número de vezes que o termo ocorre no documento pelo número total de palavras no documento. O IDF é calculado como o logaritmo da razão entre o número total de documentos no *corpus* e o número de documentos que contêm o termo. O TF-IDF é obtido multiplicando o TF pelo IDF para cada palavra no documento, gerando um valor que indica a importância da palavra tanto no documento quanto no conjunto de documentos em geral [9].

Segundo [10], o modelo que apresenta melhores resultados na classificação de textos curtos da área jurídica foi criado por meio de uma combinação de Word2Vec, treinado com um *corpus* de domínio específico, aplicado em uma arquitetura *Recurrent Neural Network* (RNN), mais especificamente *Long Short-Term Memory* (LSTM).

Em [10], os autores concluíram que modelos construídos utilizando *embeddings* específicos do domínio abordado, neste caso através de documentação jurídica no idioma português do Brasil, são superiores à modelos treinados com *embeddings* baseados em documentos genéricos.

Para os casos que envolvem conciliação processual no setor da Justiça do Trabalho, um modelo LSTM é treinado por [7], o qual é denominado LST-Mensembler, aplicando-o em processos judiciais com o objetivo de prever se uma mediação terá sucesso ou não. O dataset utilizado possui 5.776 casos do comitê de mediação de Tainan, Taiwan, compreendidos em um período de março 2009 a janeiro de 2017. O modelo combinou as vantagens de diferentes classificadores, além de considerar dependências do processo com casos anteriores, aumentando o desempenho da previsão da mediação.

Para realizar a tarefa de previsão das mediações,os autores de [7] utilizam os classificadores XGBoost e LightGBM para tratamento dos metadados relativos aos processos (sub-categoria do caso, número de participantes, mediador envolvido, dentre outros), devido à natureza numérica e categórica de seus atributos. Enquanto para a parte de descrição textual dos casos, é utilizada a ferramenta de extração de conteúdo para gerar vetores de *embedding*: BERT [7].

O XGBoost é um algoritmo, baseado em árvore de decisão, que utiliza uma estrutura de Gradient Boosting. Já o LightGBM, algoritmo de árvore de decisão desenvolvido pela Microsoft em 2017, quando comparado ao XGBoost, revela-se capaz de processar dados em maior velocidade, utilizando uma menor quantidade de memória [7].

Na etapa de mineração textual, além do BERT, também foi empregada uma ferramenta de PLN: TextCNN [7]. TextCNN é um algoritmo de deep learning adequado para executar tarefas de classificação de frases, valendo-se de uma arquitetura CNN focada para processamento de texto.

Em [5], os autores abordam que a geração de word embeddings, uma forma de representação de termos através de um vetor que leva em consideração o contexto, vem exercendo papel fundamental para se realizar análises em conjuntos de dados não estruturados, dados estes presentes em grande escala nos documentos das cortes judiciais.

No trabalho de [5] foi apresentado que a técnica RoBERTa pt-BR atinge os melhores resultados ao executar a tarefa de agrupamento de documentos judiciais do tipo Recurso Ordinário, superando o BERT e o GPT-2. Em termos de poder de processamento a técnica RoBERTa pt-BR se destaca, mostrando-se bem eficiente quando aplicada a modelos treinados por períodos mais longos e submetidos a maiores cargas de dados.

3 Método Proposto

Nesta seção são detalhadas as etapas e os métodos que são utilizados para o desenvolvimento dos modelos propostos para apoiar a tomada de decisão judicial, envolvendo classificação processual, com o objetivo de verificar a probabilidade de sucesso de um processo trabalhista ser resolvido em audiência de conciliação.

Na construção da base de dados, adotamos critérios específicos visando à representatividade e abrangência necessárias. Para selecionar os processos, empregamos critérios de data e localidade. No que diz respeito à data, foram considerados os anos de 2020, 2021 e 2022, a fim de abarcar evoluções de decisões jurídicas ao longo desse período de três anos. Quanto à localidade, focamos nas Varas do Trabalho tanto da capital como do interior, garantindo assim uma diversidade geográfica que reflete as nuances do sistema jurídico em diferentes regiões. Essa abordagem estratégica na escolha dos processos permitiu a criação de uma base de dados robusta e representativa para as análises e avaliações subsequentes dos modelos gerados.

A obtenção dos dados se deu por meio de alguns sistemas e ferramentas mantidos pelo TRT 8. O sistema Hórus [6] é um sistema de uso interno utilizado para levantamento de estatísticas e dados dos processos trabalhistas do tribunal em questão. A seleção realizada no Hórus contou com o auxílio de membros do TRT 8 para geração das planilhas contendo as referências dos processos que irão compor o conjunto de dados de interesse.

A referência a estes dados é repassada a um extrator responsável por obter diretamente os processos do sistema Processo Judiciário Eletrônico (PJe) do TRT 8 [12]. O PJe é um sistema de domínio público que tem por objetivo informatizar e agilizar os procedimentos jurídicos, proporcionando a tramitação digital de processos judiciais, eliminando ou reduzindo a necessidade de documentos físicos e tornando todo o processo mais eficiente e acessível.

O *dataset* utilizado para geração dos modelos considerados neste estudo é composto por processos extraídos do PJe mantido pelo TRT 8 [12], os quais contém documentos com extensões em PDF e HTML. Para o tratamento dos dados em formato PDF é utilizada a biblioteca Apache Tika versão 2.6.0 [1] e para tratamento dos arquivos em formato HTML é empregada a biblioteca html2text versão 2020.1.16 [8]. Após a etapa de pré-processamento, a qual é detalhada na Seção 3.1, estes dados são tabelados em um arquivo CSV, formando o *corpus* de treinamento.

Para a geração dos modelos neste estudo é empregada a linguagem Python em ambiente Google Colab. Neste processo o método de aprendizado supervi-

sionado é utilizado, onde o treinamento inicial se baseou em um conjunto de dados rotulados composto por petições iniciais dos processos e a decisão destes (conciliado ou não conciliado).

Este estudo busca preservar a confidencialidade dos dados envolvidos na montagem dos modelos, garantindo que informações pessoais ou sensíveis não sejam expostas, minimizando riscos potenciais de abuso ou violações de privacidade, o que demonstra conformidade com a Lei Geral de Proteção de Dados (LGPD). Atendendo ao objetivo final deste estudo, os dados presentes no *dataset* foram utilizados apenas com a finalidade de treinamento e criação dos modelos, permitindo a avaliação destes em relação às métricas definidas, sem a exposição dos dados envolvidos.

A Figura 1 esquematiza as etapas envolvidas na montagem do *dataset*.

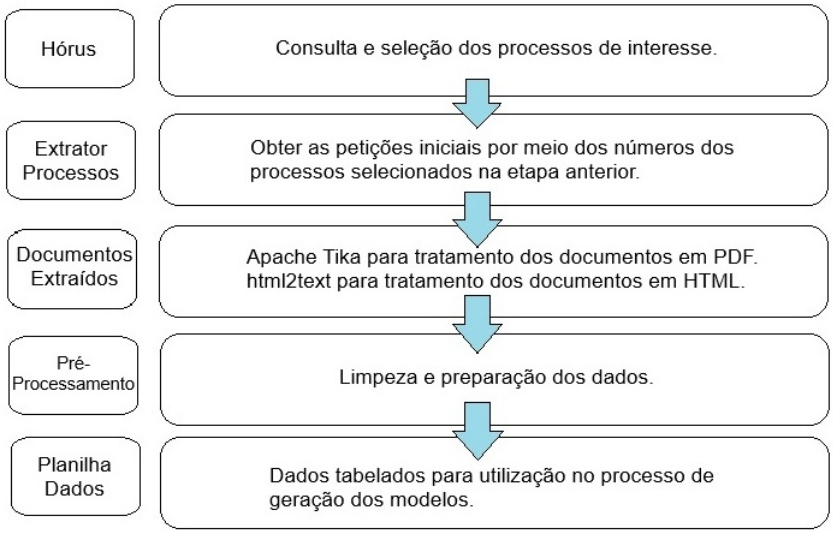

Fig. 1. Fases para montagem do *dataset*.

3.1 Pre-processamento

As etapas envolvidas no pré-processamento consistem em:

1. Remoção de acentuações, caracteres especiais (caracteres não-ASCII), numerais e pontuações;
2. Conversão do texto para caixa baixa;
3. Remoção de *stopwords* (termos comuns da língua que não possuem grande significado semântico para a interpretação de documentos, ocorrendo com frequência ao longo do texto);

4. *Stemming* para decompor os termos em radicais e afixos, a fim de proporcionar mais generalidade aos termos considerados [11].
5. Tokenização (quebrar as sentenças em palavras).

Nesta etapa de pré-processamento a biblioteca Natural Language Toolkit (NLTK) [2] é utilizada para aplicar as funções de limpeza nos textos, uma vez que esta trabalha com PLN em diversos idiomas, incluindo o português do Brasil. A Figura 2 ilustra, exemplificando, as fases envolvidas no pré-processamento.

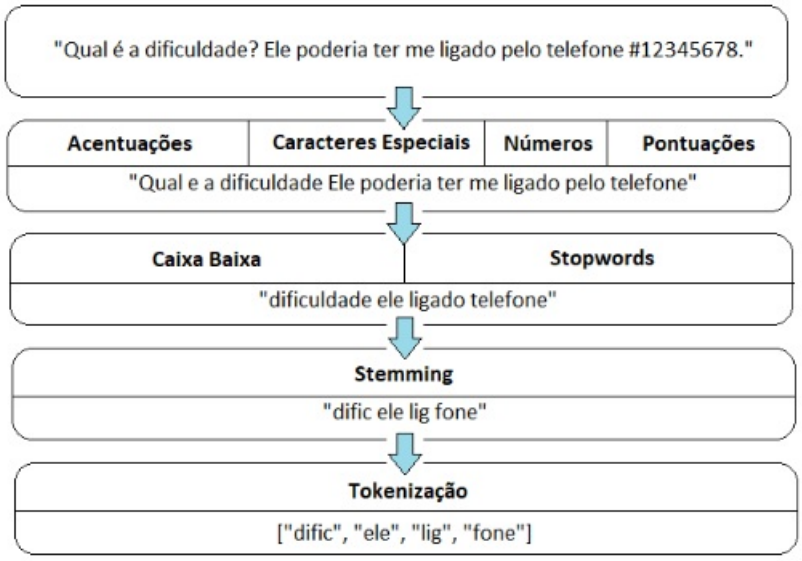

Fig. 2. Etapas de pré-processamento.

3.2 Representação do Texto

Após a etapa de pré-processamento, o próximo passo é a conversão do texto para um formato representativo apropriado que possa ser utilizado na geração dos modelos. Para a abordagem escolhida neste trabalho os vetores de *tokens* são processados por um algoritmo que atribui a medida estatística TF-IDF [9], a partir da contagem de ocorrências de um determinado termo dentro de um texto. Dessa forma, é possível determinar a quantidade de ocorrências de um determinado termo em todo o *corpus*, gerando, assim, um fator matemático de relevância para cada *token* contido no texto. A biblioteca empregada para converter os textos na representação matricial TF-IDF, a qual é usada no treinamento e geração dos modelos, é a *sklearn.feature_ extraction.text.TfidfVectorizer* [14]

3.3 Classificadores e modelos

A utilização de algoritmos clássicos de aprendizado de máquina como uma abordagem inicial em uma análise oferece uma base sólida para entender os princípios do aprendizado de máquina, estabelecer um ponto de partida para o desempenho do modelo e obter *insights* valiosos dos dados.

Embora os avanços recentes em *deep learning* tenham expandido o cenário de aprendizado de máquina, os algoritmos clássicos ainda desempenham um papel fundamental em muitas aplicações devido à sua eficácia comprovada, interpretabilidade, além da simpliscidade e compreensão que eles apresentam.

Por este trabalho ter escopo pautado em uma análise preliminar do problema abordado, considerou-se três algoritmos clássicos de aprendizado de máquina para geração dos modelos: *Support Vector Machines* (SVM), regressão logística e árvores de decisão. Além desses três, também é gerado um quarto modelo por meio do algoritmo *boosted tree* (XGBoost) que combina várias árvores de decisão simples para criar um modelo de maior poder preditivo.

A escolha do *kernel* é essencial para a eficácia dos modelos gerados por algoritmos SVM, pois diferentes tipos de dados e problemas requerem abordagens de mapeamento distintas. A seleção adequada do *kernel* envolve a compreensão da estrutura dos dados e a natureza do problema em questão, onde se destacam como os mais comumente utilizados o linear, o polinomial e o *radial basis function* (RBF). Para o escopo deste trabalho são gerados dois modelos, no caso dos algoritmos SVM, um para o *kernel* linear e outro para o RBF.

O *Linear Kernel* é adequado para separar dados linearmente, o que é útil quando os dados têm uma fronteira de decisão simples. Já o RBF *kernel* é versátil e eficaz na captura de fronteiras de decisão complexas em espaços de alta dimensão, no entanto, a escolha inadequada da largura do RBF pode levar a *overfitting*.

Para regressão logística também são gerados dois modelos, um utilizando o parâmetro *solver lbfgs* e o outro com valor *liblinear*. Para algoritmos de regressão logística, os parâmetros desempenham um papel crucial na modelagem da relação entre as características dos dados e a probabilidade de pertencer a uma determinada classe. A importância desses parâmetros reside na capacidade de ajustar o modelo para se ajustar aos dados da melhor maneira possível. A seleção inadequada desses parâmetros pode levar a um ajuste excessivo (*overfitting*) ou a um ajuste insuficiente (*underfitting*) do modelo aos dados de treinamento, afetando negativamente sua capacidade de generalização para novos dados.

Já para o modelo de árvore de decisão avaliado neste trabalho são consideradas as configurações com valores padrão do algoritmo.

Por fim, o modelo gerado pelo algoritmo XGBoost trabalha voltado para problemas de regressão. Assim, é preciso converter o problema de regressão em um problema de classificação binária, definindo um limite de decisão. É selecionado um valor de corte que separe as classes positiva e negativa, onde neste caso, selecionamos o valor igual a $0,5$. Após isso, as previsões e os rótulos são convertidos em classes binárias, permitindo a posterior avaliação em padrões comparáveis com os outros modelos.

3.4 Conjunto de dados e configuração experimental

O *dataset* considerado neste trabalho contém 30.398 documentos de petições iniciais provenientes de processos dos anos de 2020, 2021 e 2022, os quais foram extraídos da base do PJe do TRT da 8ª Região. Todas as amostras estão devidamente rotuladas identificando se os processos foram concluídos por meio de conciliação ou não.

As amostras foram organizadas de forma a construir um *dataset* equilibrado, sendo metade da classe dos conciliados e a outra metade da classe oposta.

A realização de um levantamento estatístico do número de palavras contidas nos documento do *dataset* oferece *insights* valiosos sobre a natureza textual dos documentos. A análise incluindo quartis, média e desvio padrão fornece uma compreensão abrangente da distribuição dos tamanhos dos documentos.

Os quartis permitem identificar faixas de valores que dividem os documentos em grupos (25%, 50% e 75%), enquanto a média oferece uma representação central da quantidade média de palavras por documento. O desvio padrão, por sua vez, indica a dispersão dos dados ao redor da média, indicando a variabilidade nos comprimentos dos documentos. A Tabela 1 consolida os valores para os quartis, média e desvio padrão referentes ao número de palavras dos documentos.

Table 1. Medidas estatísticas para o número de palavras dos documentos.

Medida	**Valor**
Primeiro Quartil (Q1)	1098
Segundo Quartil (Mediana)	1784
Terceiro Quartil (Q3)	2839
Média	2235,81
Desvio Padrão	1832,67

Essas estatísticas combinadas revelam tendências, outliers e a variabilidade dos tamanhos dos documentos, o que pode ser crucial para modelagem, análise de texto e compreensão da complexidade do conteúdo nos dados. A Figura 3 apresenta a distribuição da quantidade de palavras nos textos dos documentos do *dataset*.

Na construção dos conjuntos utilizados para geração dos modelos o dataset é dividido em 70% e 30% para treino e teste, respectivamente. Em seguida, aplica-se a função de vetorização TF-IDF para representação matricial dos textos, gerando-se dois conjuntos: *tfidf_ train_ vectors*, a partir dos dados de treino, e *tfidf_ test_ vectors*, a partir dos dados de teste. Estes são os conjuntos utilizados para construção dos modelos.

Para a representação matricial TF-IDF é definido que o valor do parâmetro *maxfeatures* seja igual a 10.000, considerado o levantamento estatístico da quantidade de palavras por documento, fazendo com que o vocabulário do *corpus* considerado esteja limitado a este valor. Dessa forma, elimina-se possíveis dimensões indesejadas que sejam formadas por palavras excepcionalmente raras.

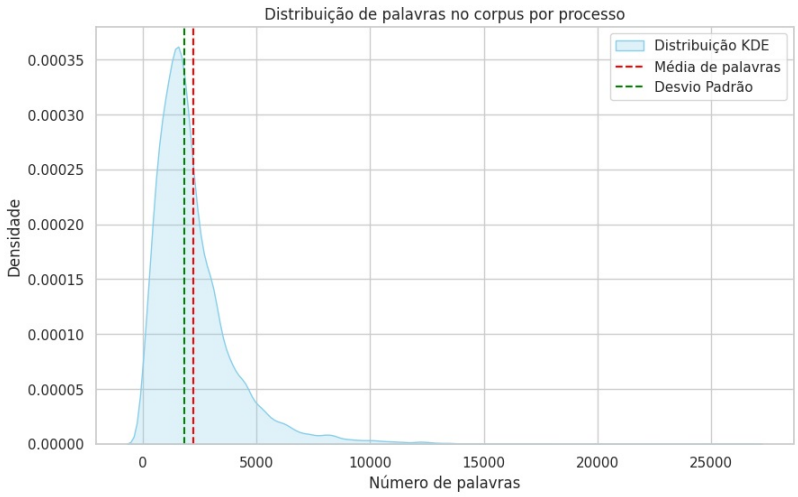

Fig. 3. Distribuição do número de palavras por documentos.

4 Métricas para avaliação dos modelos

Ao avaliar um modelo, é importante considerar o contexto da tarefa de aprendizado de máquina e o domínio em questão para selecionar métricas que possam fornecer medidas adequadas para possibilitar a realização de uma análise de qualidade.

Dessa forma, as métricas de avaliação de modelos são uma ferramenta fundamental para avaliar e comparar modelos de aprendizado de máquina. Por isso, ao escolher e interpretar essas métricas, é importante considerar o contexto da tarefa e usar outras técnicas de avaliação a fim de se obter uma avaliação mais completa do modelo.

Para este trabalho são consideradas as métricas de: Acurácia, F1-Score, *Matthews Correlation Coefficient* (MCC) e *Area Under the ROC Curve* (AUC).

4.1 Acurácia

A acurácia é uma métrica simples e direta que mede a proporção de previsões corretas em relação ao total de previsões feitas pelo modelo. Matematicamente, é calculada pela divisão do número de previsões corretas sobre o total de previsões.

A acurácia é especialmente útil quando as classes têm um tamanho aproximadamente igual e quando os erros em ambas as direções (falsos positivos e falsos negativos) são igualmente importantes. No entanto, ela pode ser enganosa em situações em que há desequilíbrio de classes, ou seja, quando uma classe é significativamente mais frequente do que a outra. Nesses casos, o modelo pode atingir uma alta acurácia simplesmente prevendo a classe majoritária em quase todos os casos, sem realmente capturar a complexidade da classificação.

4.2 F1-Score

O F1-Score é uma métrica que combina as informações de precisão e recall para fornecer uma visão mais abrangente do desempenho do modelo. Precisão e recall são calculados da seguinte forma:

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}}$$

$$\text{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}}$$

O F1-Score é a média harmônica entre precisão e recall, sendo seu cálculo apresentado da seguinte maneira:

$$F1\text{-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

4.3 Matthews Correlation Coefficient (MCC)

É uma medida de avaliação de desempenho de classificadores binários que leva em conta tanto verdadeiros positivos quanto verdadeiros negativos, bem como falsos positivos e falsos negativos [13].

O MCC varia entre -1 e 1, sendo que 1 indica uma classificação perfeita, 0 indica uma classificação aleatória e -1 indica uma classificação totalmente oposta à verdadeira [13].

4.4 Area Under the ROC Curve (AUC)

Comumente usada para avaliar a qualidade de um modelo de classificação binária. A curva ROC (Receiver Operating Characteristic) é uma representação gráfica do desempenho do modelo em diferentes limiares de probabilidade de classificação. A AUC é calculada como a área sob a curva ROC e varia entre 0 e 1, onde 1 indica um modelo perfeito e $0,5$ indica um modelo que não tem poder de discriminação entre as classes [4].

A AUC fornece uma medida agregada do desempenho do modelo para todos os possíveis limiares de classificação. Uma AUC alta indica que o modelo tem uma boa capacidade de distinguir entre as classes, enquanto uma AUC baixa indica que o modelo tem dificuldade em distinguir entre as classes.

5 Resultados e discussão

O presente trabalho apresenta modelos gerados por classificadores considerados clássicos da literatura, como os lineares, no caso do SVM e regressão logística, árvores de decisão, além de considerar o algoritmo de combinação de árvores de decisão, o XGBoost. Por meio das métricas definidas na Seção 4 é possível

estabelecer uma análise comparativa dos resultados obtidos em cada modelo considerado.

O modelo gerado pelo SVC (*Support Vector Classifier*) com um *kernel* linear obteve uma acurácia de $81,88\%$, além de apresentar desempenho alto nas outras métricas, indicando um bom desempenho geral. Enquanto o modelo SVC com um *kernel* Gaussiano (RBF) apresenta acurácia de $83,12\%$, superior em comparação ao SVC linear, com resultados superiores em todas as métricas utilizadas. Isso sugere que a aplicação do *kernel* Gaussiano pode ter melhorado a capacidade de separação das classes conciliado e não-conciliado.

O modelo de Regressão Logística, com o algoritmo *lbfgs*, obteve resultados de acurácia $81,41\%$, semelhantes ao SVC linear. As métricas de desempenho estão todas próximas, como pode ser visto na Tabela 2, sugerindo um desempenho comparável para o conjunto de dados. A Regressão Logística com o algoritmo *liblinear* também apresentou resultados semelhantes aos modelos analisados. Novamente, as métricas de desempenho são próximas, indicando um desempenho comparável entre os modelos.

O modelo gerado de Árvore de Decisão obteve uma acurácia de $77,83\%$, menor em comparação aos modelos SVC linear, SVC Gaussian, Regressão Logística. As métricas MCC e o F1-Score, $0,5566$ e $0,7825$, respectivemante, também estão inferiores, sugerindo um desempenho inferior na capacidadde de separar as classes em estudo.

O modelo Boosted Tree, utilizando a biblioteca XGBoost, obteve uma acurácia e AUC relativamente mais baixos em comparação com os outros modelos. O MCC e o F1-Score também são inferiores, indicando que este modelo pode não ser tão eficaz na classificação desses dados específicos.

Table 2. Modelos e métricas com resultados.

Modelos	**Acurácia**	**AUC**	**MCC**	**F1-Score**
SVC linear	0,8188	0,8185	0,6404	**0,8284**
SVC Gaussian *kernel* RBF	**0,8312**	**0,8307**	**0,6658**	0,8223
Regressão Logística lbfgs	0,8141	0,8138	0,6298	0,8222
Regressão Logística liblinear	0,8142	0,8139	0,6300	0,8223
Árvore de Decisão	0,7783	0,7782	0,5566	0,7825
Boosted Tree (XGBoost)	0,7654	0,7647	0,5400	0,7873

Embora o XGBoost seja amplamente reconhecido como um algoritmo de aprendizado de máquina do estado da arte para tarefas de classificação, é importante considerar que diversos fatores podem resultar em um desempenho insatisfatório ao empregá-lo.

Um aspecto crucial é a configuração dos hiperparâmetros, uma vez que o XGBoost oferece uma gama considerável deles, exigindo ajustes adequados para cada conjunto de dados. Uma escolha inadequada dos hiperparâmetros pode conduzir ao overfitting, onde o modelo se ajusta excessivamente aos dados de treinamento, mas não consegue generalizar eficazmente para novos dados. Além

disso, a qualidade das características utilizadas é igualmente relevante. Características irrelevantes ou redundantes, desprovidas de correlação com a variável alvo, assim como aquelas excessivamente correlacionadas entre si, têm o potencial de impactar negativamente o desempenho do XGBoost, diminuindo sua capacidade de aprender padrões discriminativos e relevantes para a tarefa de classificação.

Portanto, atentar para a escolha dos hiperparâmetros e para a seleção de características é essencial para extrair todo o potencial do XGBoost e obter resultados de alta qualidade em tarefas de classificação.

6 Conclusões e trabalhos futuros

Diante da análise realizada, temos que o SVM com *kernel* RBF apresentou melhores resultados por meio das métricas consideras na avaliação dos modelos. Este modelo alcançou uma acurácia de 83% e F1-Score de 82%, apresentando um MCC de 0.66 e uma AUC de $0,83$.

Para evolução da análise e avaliação considerada no escopo deste trabalho, pretende-se aplicar em estudos futuros técnicas que permitam ajustes mais finos nos modelos gerados. Desse modo, percebe-se que a combinação da validação cruzada e da hiperparametrização desempenha um papel fundamental na otimização do desempenho dos algoritmos de aprendizado de máquina empregados para criar modelos mais robustos.

A validação cruzada permite avaliar a capacidade de generalização de um modelo, dividindo os dados em conjuntos de treinamento e teste múltiplas vezes. Isso ajuda a evitar avaliações enviesadas e fornece uma visão mais robusta sobre a capacidade do modelo de se ajustar a diferentes conjuntos de dados.

A hiperparametrização, por outro lado, envolve a busca pelos melhores valores de hiperparâmetros que controlam o comportamento do algoritmo. Isso inclui parâmetros que influenciam a complexidade do modelo, a regularização e outros aspectos cruciais.

Assim, através da combinação da validação cruzada para avaliação do desempenho em diferentes configurações de hiperparâmetros, é possível identificar combinações que resultam em modelos mais generalizáveis e eficazes.

Os trabalhos futuros destinam-se a aprofundar a pesquisa, buscando a incorporação de modelos de linguagem em português para geração de novos modelos de classificação. Uma outra abordagem que também pode ser aplicada nos textos considerados em estudo é a utilização da técnica de PLN Word2Vec, permitindo a vetorização e transformação dos dados como feito pelo TF-IDF usado no presente trabalho.

Por fim, destacamos o importante papel da justiça do trabalho para a sociedade brasileira, onde a otimização do processo trabalhista é essencial para a proteção dos direitos do trabalho, a promoção da justiça social e o fortalecimento das relações de trabalho no país. Assim, ao incorporar modelos de aprendizado de máquina e integrá-los aos seus sistemas eletrônicos e ferramentas, a justiça

trabalhista conquista benefícios que geram ganhos de qualidade e eficiência aos seus processos, além de melhorar diretamente o atendimento à sociedade.

References

1. Apache Tika, biblioteca de análise de texto Apache Tika. Disponível em: https://tika.apache.org/. Acesso em 23 de mar. de 2023.
2. Bird, S.; Loper, E.; Klein, E. Natural Language Processing with Python. [S.l.]: O'Reilly Media Inc, 2009. ISBN 0596516495.
3. CEJUSC. Centro Judiciário de Solução de Conflitos e Cidadania. CEJUSC, 2022. Disponível em https://www.trt8.jus.br/cejusc. Acesso em 21 de mar. de 2022.
4. Davis, J., Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine learning, 233-240.
5. De Oliveira, Raphael Souza, and Nascimento, Erick Giovani Sperandio. "Brazilian Court Documents Clustered by Similarity Together Using Natural Language Processing Approaches with Transformers." arXiv preprint arXiv:2204.07182 (2022)
6. Horus, Hórus (Inteligência do negócio), Sistemas do Tribunal Regional do Trabalho 8ª Região, 2022. Disponível em: <https://www.trt8.jus.br/servicos> <https://horus.trt8.jus.br/index.htm>
7. Hsieh, Hsun-Ping, et al. "Predicting the Success of Mediation Requests Using Case Properties and Textual Information for Reducing the Burden on the Court." Digital Government: Research and Practice 2.4 (2022): 1-18
8. Html2text, ferramenta para converter um documento HTML em texto. Disponível em: https://github.com/grobian/html2text. Acesso em 23 de mar. de 2023.
9. Jurafsky, D., Martin, J. H. (2020). Speech and language processing. An introduction to natural language processing, computational linguistics, and speech recognition. Pearson.
10. Noguti, Mariana Y., Eduardo Vellasques, and Luiz S. Oliveira. "Legal document classification: An application to law area prediction of petitions to public prosecution service." 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.
11. Orengo, V.; Huyck, C. A stemming algorithmm for the portuguese language. IEEE (em inglês): 186–193. Novembro de 2001. doi:10.1109/SPIRE.2001.989755
12. PJE, Processo Judicial Eletrônico, Sistemas do Tribunal Regional do Trabalho 8ª Região, 2023. Disponível em: https://www.trt8.jus.br/servicos https://www.trt8.jus.br/pje
13. Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
14. Sklearn TfidfVectorizer, biblioteca python para converter textos em matriz TF-IDF. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Acesso em 13 de ago. de 2023.
15. TST. Tribunal Superior do Trabalho. Matérias Temáticas Conciliação. TST, 2021. Disponível em http://www.tst.jus.br/web/guest/conciliacao. Acesso em 23 de nov. de 2021.