Processamento da Linguagem Natural para análise de documentos jurídicos

Gabriel Souza Silva, Leandro Ribeiro, Henrique Dezani (orientador)

e-mail: gabrielsrad08@gmail.com; leandro.ribeiro.fatec@gmail.com; henrique.dezani@fatec.sp.gov.br

Faculdade de Tecnologia de São José do Rio Preto

Resumo: Diante do alto número de processos que tramitam na justiça, o Poder Judiciário busca alternativas para garantir maior celeridade e eficiência na prestação jurisdicional e uma delas é a utilização da tecnologia. Esse estudo tem por objetivo analisar a aplicação do Processamento da Linguagem Natural, um ramo da inteligência artificial que transforma a linguagem humana para compreensão da linguagem computacional e sua aplicação com técnicas de *machine learning* para classificação automática dos documentos em áreas cível ou criminal. Para isso, uma base de dados foi elaborada mediante sistemas de busca jurídicos criando um acervo de ementas processuais dessas duas áreas do direito. Por intermédio da linguagem de programação Python, os dados textuais são processados e tratados para uma classificação eficaz. Os resultados demonstram que a inteligência artificial foi capaz de reconhecer e classificar esses documentos com relevante acurácia, de tal modo que há possiblidade de ampliar o trabalho para classificação nas demais áreas do Direito bem como tornar a base de dados mais robusta para englobar uma maior parte do vocabulário jurídico.

Palavras-chave: Processamento da linguagem natural; inteligência artificial; classificação jurídica.

Abstract: *Faced with the high number of cases that are already being processed in court, the Judiciary is looking for alternatives to ensure greater speed and efficiency in the judicial provision, and one of them is the use of technology. This study aims to analyze the application of Natural Language Processing, a branch of artificial intelligence that transforms human language to understand computational language and its application with machine learning techniques for automatic classification of documents in civil or criminal areas. For this, a database was created using legal search systems, creating a collection of procedural summaries of these two areas of law. Through the Python programming language, textual data are processed and treated for effective classification. The results demonstrate that the artificial intelligence was able to recognize and classify these documents with relevant accuracy, in such a way that there is the possibility of expanding the work for classification in other areas of Law, as well as making the database more robust to encompass a greater part of of the legal vocabulary.*

Keywords: *Natural language processing; artificial intelligence; legal classification.*

1 INTRODUÇÃO

A Lei n. 11.419, de 19 de dezembro de 2006 foi o primeiro diploma legislativo que permitiu o uso de meio eletrônico na tramitação de processos judiciais, comunicação de atos e transmissão de peças processuais. Nos anos seguintes, os Tribunais não mediram esforços para gradativamente digitalizar seus acervos processuais.

Com as restrições sanitárias decorrentes da pandemia provocada pela COVID19, as formas usuais pelas quais partes e advogados utilizavam os serviços do Poder Judiciário foram impactadas. Em contrapartida, algumas soluções digitais foram consolidadas nesse período para garantir a continuidade e eficiência da prestação jurisdicional.

Nesse contexto, o Conselho Nacional de Justiça implementou o Juízo 100% Digital e o Balcão Virtual, como forma de planejar e estruturar prospectivamente por meio de uma atuação estratégica de iniciativas digitais encadeadas no Programa Justiça 4.0.

Segundo o relatório Justiça em números 2022 (2022) do Conselho Nacional de Justiça, o Programa Justiça 4.0 tem por finalidade promover o acesso à Justiça, através de ações e projetos desenvolvidos para o uso colaborativo de produtos que empregam novas tecnologias e inteligência artificial. As inovações tecnológicas revelam a possibilidade de dar celeridade à prestação jurisdicional e reduzir despesas orçamentárias decorrentes desse importante serviço público.

Nos processos judiciais, a principal forma de comunicação humana é realizada pela escrita. Como ramo da inteligência artificial, o Processamento da Linguagem Natural pode ser entendido como o processamento automático da linguagem humana (SINGH, 2021). Com ele, há maneiras pelas quais os sistemas de computador analisam e interpretam textos. Em outras palavras, Processamento de linguagem natural é a capacidade das tecnologias de processar a linguagem natural humana.

Nesse estudo, as técnicas de Processamento da Linguagem Natural serão aplicadas na área jurídica para classificação de documentos. Para isso, é necessário a elaboração de um Corpus. Ele fornece fonte de dados para as abordagens de aprendizado de máquina. Nesse estudo, o Corpus é do tipo categorizado e será organizado em duas categorias, cível e criminal de modo supervisionado para que novos documentos sejam automaticamente classificados dentro dessas duas grandes áreas do Direito.

1.1 Justificativa

A análise dos números de processos pendentes no Brasil revela os esforços necessários para a utilização da tecnologia como forma de buscar maior celeridade e eficiência na prestação jurisdicional. De acordo com os dados do CNJ (Conselho Nacional de Justiça), o Poder Judiciário finalizou o ano de 2021 com 77,3 milhões de processos em tramitação.

É evidente que o Poder Judiciário busca modernização e o apoio da tecnologia para facilitar os serviços prestados bem como buscar alternativas para que a prestação jurisdicional ocorra de maneira mais célere. Isso fica claro ao analisarmos o percentual de processos eletrônicos no País. No ano de 2009 apenas 11% eram digitais ao passo que no ano de 2022 o índice saltou para 98,90% (CREPALDI, 2022).

Em que pese os esforços da implementação da Justiça 4.0, com o Balcão Virtual e Juízo 100%, é necessário acompanhamento das ferramentas tecnológicas, observar a sua aplicação e a manutenção em termos de atualidade para o contínuo trabalho de assegurar uma atividade jurisdicional célere e eficiente.

1.2 Objetivos

O presente estudo tem por objetivo geral contribuir para os estudos de aplicação de tecnologias para aperfeiçoamento das áreas jurídicas. A classificação automática de documentos, através das técnicas de Processamento da Linguagem Natural, pode eliminar etapas realizadas lentamente pela ação humana, substituindo-as pela atividade computacional. Especificamente, o objetivo é aplicar o Processamento da Linguagem Natural utilizando-se da linguagem de programação Python para classificar ementas de decisões de órgãos colegiados para indicar de modo autônomo se eles são da área cível ou criminal.

2 REVISÃO DA LITERATURA

O Processamento da Linguagem Natural é o processamento automático da linguagem humana. Há maneiras pelas quais os sistemas de computador analisam e interpretam textos. Nesse sentido, Processamento de linguagem natural é a capacidade das tecnologias de processar a linguagem natural humana, seja ela escrita ou falada (THANAKI, 2017, p.09).

Segundo Rezende (2003, p. 339), a abordagem dos dados obtidos pelo processamento de textos pode ser do tipo semântica ou estatística. Na análise semântica haverá emprego de técnicas para avaliar a sequência de termos no contexto da oração ou período para que cada termo tenha sua função identificada. Por outro lado, a análise estatística, que será aplicada nesse estudo, considera a importância dos termos pela frequência de vezes em que aparecem nos textos.

Para o processamento desse estudo, que será debruçado sobre a linguagem escrita, uma base de dados contendo as informações escritas utilizadas para comparação de textos deve ser utilizada. Essa base recebe o nome de Corpus. É uma coleção de material de linguagem natural, armazenado e usado para descobrir como a linguagem é usada. Em outras palavras, o corpus consiste em coleção computadorizada sistemática de linguagem autêntica que é usada para análise linguística (THANAKI, 2017, p.20).

O corpus foi composto por ementas de jurisprudência. As ementas resumem o conteúdo de decisões judiciais, sintetizando as razões jurídicas e as consequências de fato atinentes ao caso em julgamento (BRASIL, 2021). Trata-se do principal canal de divulgação da jurisprudência ao público, composto por pequenas frases que contextualizam o julgado.

O trabalho utiliza da linguagem de programação Python. Essa linguagem possui uma biblioteca denominada NLTK (Natural Language Toolkit). Ela é uma coleção de bibliotecas Python projetadas especialmente para identificar e marcas partes dos textos. Também utilizamos a biblioteca Pandas, que fornece ferramentas úteis para análise de dados e visualização de dataframes.

Para tratar os textos pode ser necessária a remoção das denominadas *Stop Words*. São palavras comuns que precisam ser removidas do texto durante o pré-processamento de dados em NLP (Processamento de Linguagem Natural), como artigos, preposições, conjunções e outras palavras muito comuns e que não indicam um significado muito útil para análise do texto.

A *tokenização* quebra em unidades menores determinado texto e também pode ser utilizado para segmentação de palavras. Também há o tratamento no sentido de colocar o texto em letras minúsculas (*lowercase*) e remoção de acentos por meio de Unidecode, biblioteca que converte caracteres para seus ASCII aproximados.

Os resultados são analisados utilizando pela acurácia. A acurácia é uma métrica que considera a proporção de exemplos classificados corretamente em relação ao total de exemplos, dividindo-se o corpus em classe de teste e classe de treino. Assim sendo, o resultado é obtido pelo cálculo que divide o número de exemplos classificados corretamente pelo número total de exemplos.

Alguns trabalhos similares foram encontrados e serviram de apoio nesse estudo. Foi possível notar que algumas técnicas de Processamento da Linguagem Natural para análise e classificação de comentários de filmes (data set IMBD) para indicar se são positivos ou negativos podem ser aproveitadas para a classificação jurídica proposta. É o caso do estudo apresentado no curso disponível no portal de cursos Alura do Professor Thiago Santos bem como o estudo denominado Introdução ao Processamento de Linguagem Natural com Python do Professor orientador Prof. Dr. Henrique Dezani. Por fim, há um trabalho semelhante encontrado de autoria do arquiteto de software Juliano Pacheco denominado ‘Aplicações de Machine Learning na área Jurídica, um exemplo com classificação de textos’.

3 METODOLOGIA

O desenvolvimento do estudo do Processamento da Linguagem Natural para análise e classificação de documentos jurídicos utilizando a linguagem de programação Python depende de alguns conjuntos de processos para sua implementação para apresentação de um estudo descritivo.

Há pesquisa bibliográfica, no intuito de levantar os dados e teses publicados sobre o tema, como artigos e livros que abordam o tema Processamento da Linguagem Natural. Além disso, tem-se uma pesquisa documental, para levantamento de ementas de jurisprudência que embasam o Corpus. Com isso, o levantamento realizado na modalidade de pesquisa quantitativa que visa coletar dados sobre uma amostra representativa de ementas cíveis e criminais, sendo inicialmente levantado de maneira arbitrária 114 ementas cíveis e mesma quantidade de ementas criminais.

Os dados das ementas são coletados em pesquisas no site de busca jurídica http://www.jusbrasil.com . São filtradas jurisprudências provenientes do STJ (Superior Tribunal de Justiça) e Tribunais de Justiça Estaduais. A filtragem ocorre de modo automático, colocando por filtro "cível" e "criminal". Importante destacar que os dados informados no site JusBrasil são de processos públicos, de acesso a qualquer pessoa. Além disso, as ementas não trazem dados pessoais de partes envolvidas no processo, limitando-se a resumir a matéria de direito discutida nos autos.

Através de técnicas de Processamento da Linguagem natural são analisadas a acurácia para classificação de documentos ou teses jurídicas. Necessário computador ou notebook para construção dos códigos necessários para processamento da linguagem, tratamento de textos e aplicação de *machine learning*, todos realizados dentro do ambiente do Google Colab.

Um cronograma orienta o estudo. A primeira atividade desenvolvida foi o levantamento de publicações e documentos úteis relacionados ao tema do Processamento da Linguagem Natural. A atividade 2 consistiu no levantamento dos dados que compõe o Corpus, através dos levantamentos das jurisprudências.

Na etapa seguinte, atividade 3, construídos os códigos baseados em Python para buscar o tratamento dos textos, análises e classificações. A atividade 4 consiste na análise do desenvolvimento e aperfeiçoamento para melhores resultados.

Por fim, a atividade 5 apresenta os resultados e conclusões do estudo realizado acerca do Processamento da Linguagem Natural aplicado a documentos jurídicos.

4 DESENVOLVIMENTO

4.1 Formação da base de dados e primeiras análises

A base de dados utilizada no estudo foi elaborada por uma pesquisa realizada no site de busca jurídica denominado JusBrasil. Como dito alhures, as ementas resumem o conteúdo de decisões judiciais por intermédio de expressões ou palavras-chave. Foram coletadas 114 ementas da área cível e 114 ementas da área criminal. Em seguida, os dados coletados foram estruturados em uma planilha Excel composta por três colunas. A primeira apenas com um número de identificação da ementa em ordem crescente. A segunda, denominada "Ementa" com todo o texto das ementas extraídas. E a terceira denominada "Tipo" em que os dados foram rotulados com 1 para cível e 0 para criminal.

Uma vez criado o Corpus, para apreciar os dados realizamos a primeira implementação de aprendizado de máquina, antes de qualquer tratamento, separando parte dos dados em classe de treino e outra de teste. Para isso, o estudo utilizou a Scikit-learn , uma biblioteca gratuita em Python que oferece uma variedade de recursos para analisar e minerar dados bem como suporte ao aprendizado de máquina supervisionado e não supervisionado (AWARI, 2022). Dessa biblioteca, aplicamos a função train_test_split, para retornar uma lista com as ementas segregadas de acordo com o tipo 0 e 1.

O primeiro tratamento nos dados consiste na utilização do Pandas replace para que fique claro quais são cíveis e quais são criminais. Para isso, atribuiremos uma nova coluna com essas separações. Para 1 a nova coluna apresentou "Cível" e para 0 apresentou "Criminal".

4.2 Criação do Vocabulário de Palavras (Bag of Words)

O estudo criou um vetor para atribuir valor a cada uma das palavras, é a forma de traduzir a linguagem natural para a linguagem de máquina. Para isso, as palavras são colocadas no vocabulário conforme surgem como novidade. Em Python, a função CountVectorizer da *sklearn* é capaz de criar as matrizes com as palavras passadas nas frases escolhidas.

Vetorizamos duas frases, uma com sentido cível ("Ação de indenização por danos morais") e outra criminal ("O réu foi condenado") para demonstrar como é estruturado o vocabulário de palavras. Ele agrupa todas as palavras das frases e indica conforme surgem como novidade. No exemplo, apresentamos em uma matriz esparsa a forma pela qual as palavras surgem já categorizadas, conforme figura a seguir.

	ação	condenado	danos	de	foi	indenização	morais	por	réu
0	1	0	1	1	0	1	1	1	0
1	0	1	0	0	1	0	0	0	1

Figura 1 Matriz esparsa para demonstrar a *Bag of words*

Observado em fase de teste com as frases acima descritas, o passo seguinte foi aplicar a vetorização em toda a coluna "Ementa" da base de dados utilizada no estudo. Isso permite observar o acervo de palavras e como podemos tratar o texto para melhor acurácia da classificação realizadas nas etapas seguintes.

4.2 Visualização dos Dados com WordCloud

O objetivo desta etapa é demonstrar visualmente dados em por uma imagem que contém várias palavras e o tamanho de cada palavra mostrada relaciona-se proporcionalmente com a frequência que aparece na base analisada (UEZONO, 2020). Para isso, inicialmente foi instalada a biblioteca denominada justamente *wordcloud* e importação do pacote *matplotblib* cuja função é transformar os dados em imagem.

A primeira visualização em imagem trouxe um quadro contendo diversas palavras das ementas levantadas no estudo, conforme figura a seguir:

Figura 2 Nuvem de palavras de toda coluna ementa

Para aprimorar a visualização e também auxiliar no tratamento do texto, foram criadas duas funções , uma para área cível e outra criminal, com intuito de apresentar quadros contendo as palavras para cada uma das pareas de classificação.

Figura 3 Nuvem de palavras ‘cível’

Figura 4 Nuvem de palavras ‘criminal’

4.3 Tokenização e visualização com Gráfico de Pareto

Após definir o quadro das palavras separados por cível e criminal, tratar esse vocabulário para melhorar e apresentar as palavras mais relevantes. A função freqDist da biblioteca NLTK retorna a frequência dos parâmetros passados. Ao passar uma frase, retornará a frequencia de toda a frase, não das palavras de maneira individual. Para isso, utiliza-se a tokenização, importando o pacote *tokenize.*

Após realizar a separação de toda a coluna ementa por espaços, temos a possiblidade de analisar a frequência de cada uma das palavras contidas. Para visualizar, foi aplicado um Gráfico de Pareto com a biblioteca Python denominada Seaborn. Notamos que as palavras que aparecem com mais frequência tem pouco valor semântico para classificação, de tal modo de que preposições e conjunções pouco contribuem para revelar se um determinado texto pertence a categoria cível ou criminal.

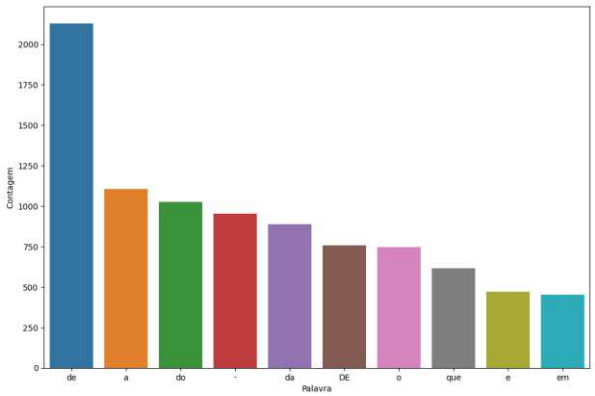

Figura 5 Gráfico de Pareto antes de tratamento do texto

Desse modo, o passo seguinte foi tratar a base de dados para remoção dessas palavras irrelevantes. Para isso, utilizamos um conjunto denominado *stopwords* que a própria biblioteca NLTK Python oferece com compatibilidade para língua portuguesa.

Após a remoção das *stop words* , observamos no gráfico de pareto que elas desapareceram, pois como dito são irrelevantes. Ocorre que contuniavam aparecendo vocábulos irrelevantes, sem teor semântico para classificação dos textos em cível ou criminal. Por isso, adicionamos palavras específicas que não poderiam aparecer: "julgamento", "data", "relator", "publicação" e "art" pois estão presentes no rodapé de quase todas as ementas.

Também retiramos as pontuações importando a função *punctuation* do pacote *string.* Continuando o tratamento da base de dados, retiramos a acentuação das palavras, por intermédio do *unidecode.* Após esses tratamentos, visualizamos o gráfico de todas as ementas, conforme figura abaixo.

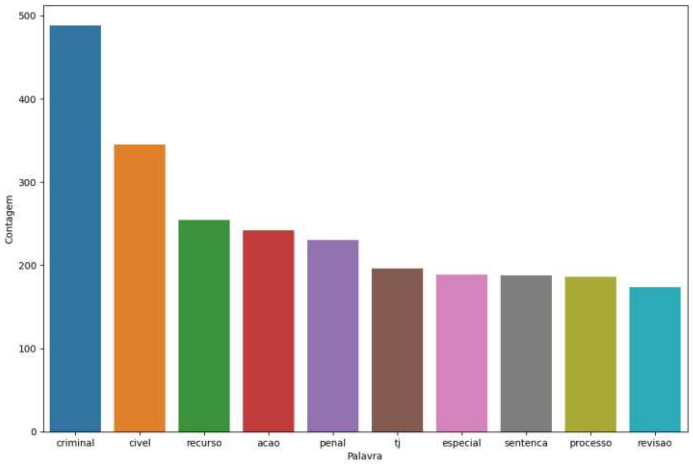

Figura 6 Gráfico de Pareto após remoção de stopwords e acentos

Após todos os tratamentos realizados na base de dados, ao gerar novamente a WordCloud para cada uma das áreas de classificação notamos uma nuvem menos poluída, contendo palavras relevantes e que realmente fazem sentido no propósito de separação das ementas. A Wordclould da esquerda apresenta as palavras relevantes para uma classificação cível e a da direita as importantes para área criminal.

Figura 7 Wordclouds após tratamentos nos textos da base de dados

4.4 Vetorização por peso *TF IDF*

Uma forma de analisar e contar as palavras do *bag of words* criado é através da técnica TF IDF, que consiste em analisar a frequência dos termos e o inverso da frequência do documento. Ela permite que termos comuns sejam considerados irrelevantes para classificação na medida em que surgem com frequência nas duas áreas analisadas e com isso perdem poder de diferenciação. Em Python, importamos o *TfidfVectorizer* do sklearn.feature_extraction.text.

Para exemplificar, utilizamos duas frases que possuem termos em comum mas pertencem a diferentes áreas. Imaginemos "Processado em ação por danos morais" para área cível e "Processado por tráfico de drogas" para área criminal. Existem termos em comum em ambas, de modo que não tem peso para diferenciação. Por outro lado, alguns termos são bem característicos de cada uma das áreas, de tal modo que terão peso maior. É exatamente esse o resultado, conforme o dataframe gerado.

```
from sklearn.feature_extraction.text import TfidfVectorizer

frases = ["Processado em ação por danos morais", "Processado em ação por tráfico de drogas"]

tfidf = TfidfVectorizer(lowercase = False, max_features=50)

caracteristicas = tfidf.fit_transform(frases)
pd.DataFrame(
    caracteristicas.todense(),
    columns = tfidf.get_feature_names_out()
)
```

	Processado	ação	danos	de	drogas	em	morais	por	tráfico
0	0.354649	0.354649	0.498446	0.000000	0.000000	0.354649	0.498446	0.354649	0.000000
1	0.317404	0.317404	0.000000	0.446101	0.446101	0.317404	0.000000	0.317404	0.446101

Figura 8 Vetorização por peso para diferenciação

Quando aplicada em toda a base de dados tratada, a acurácia foi aumentada consideravelmente. Por fim, apresentamos um dataframe que demonstra os pesos dos principais termos que o modelo aprendeu. São as palavras que realmente diferenciam e realizam a classificação. Temos na figura abaixo, a esquerda estão os principais termos e respectivo peso para área criminal e aqueles que diferenciam negativamente, ou seja, seriam da área cível.

	0
crim	2.927225
revisa	1.152913
penal	1.060713
justificaca	0.644527
pen	0.606934
reabilitaca	0.459691
traf	0.447960
apr	0.437905
condenaca	0.436037
absolvica	0.426343

	0
civel	-2.504790
juiz	-1.233819
espec	-1.206719
dan	-0.883099
aca	-0.742902
competenc	-0.667255
civil	-0.638879
moral	-0.524184
turm	-0.522215
aliment	-0.495504

Figura 9 Principais termos de diferenciação das áreas e seus pesos

5 RESULTADOS E DISCUSSÕES

Dividindo a base de dados em uma classe de teste e outra de treino, logo no início do desenvolvimento foi implementada técnica de *machine learning* para medir acurácia, ou seja, o nível de precisão que a atividade computacional poderia corretamente classificar os documentos. Os resultados foram analisados pela acurácia, métrica que considera a proporção de exemplos classificados corretamente em relação ao total de exemplos. Assim sendo, o resultado é obtido pelo cálculo que divide o número de exemplos classificados corretamente pelo número total de exemplos.

Antes mesmo do tratamento, essa acurácia foi expressiva alcançando 0,94 em uma escala de 0 a 1. Isso revelou que os termos jurídicos possuem grande grau de diferenciação entre si, atribuindo bastante sentido e possibilidade de classificação em cada vocábulo empregado.

É necessário registrar que a base de dados utilizada foi simbólica, utilizada para testes e aplicações do processamento da linguagem natural. Desse modo, é evidente que a ampliação dessa base de dados, inserindo muitas ementas, enriqueceria o vocabulário e tornaria a classificação mais ampla e possivelmente com redução da acurácia inicial. Entretanto, notamos que a medida que os textos foram tratados houve incremento da acurácia, o que revela que as técnicas aplicadas poderiam atender satisfatoriamente a ampliação sobredita pois no estudo após a implementação das técnicas atingiu-se uma acurácia de 1.

6 CONCLUSÕES

O estudo demonstrou relevância da aplicação do Processamento da Linguagem Natural com a linguagem de programação Python na área jurídica e sua possibilidade de contribuição para automatização de processos, apresentando bons resultados. Sua aplicação pode ser uma das alternativas para acelerar etapas realizadas exclusivamente pela ação humana. Evidentemente que são estudos iniciais, que podem ser aperfeiçoados em trabalhos futuros, especialmente ampliados para que sejam aplicados nas demais áreas do Direito. Ainda, novos estudos poderiam comparar a técnica utilizadas com outras técnicas e algoritmos.

REFERÊNCIAS

AWARI (Brasil) (ed.). **Entenda o que é Scikit Learn e aprenda como usar essa biblioteca. 2022**. Disponível em: https://awari.com.br/scikit-learn/?utm_source=blog. Acesso em: 25 abr. 2023.

BRASIL. CONSELHO NACIONAL DE JUSTIÇA. **Diretrizes para a elaboração de ementas**. 2021. Disponível em: https://www.cnj.jus.br/wp-content/uploads/2021/09/diretrizes-elaboracao-ementas-uerj-reg-cnj-v15122021.pdf. Acesso em: 19 mar. 2023.

BRASIL. CONSELHO NACIONAL DE JUSTIÇA. **Justiça em números 2022**. Brasília: Cnj, 2022. 331 p. Disponível em: https://www.cnj.jus.br/wp-content/uploads/2022/09/justica-em-numeros-2022-1.pdf. Acesso em: 18 mar. 2023.

CREPALDI, Thiago; GOES, Severino. **Justiça brasileira alcança marca de 80 milhões de processos em tramitação**. 2022. Disponível em: https://www.conjur.com.br/2022-jun-30/poder-decide-faz. Acesso em: 13 maio 2023.

JUSBRASIL (Brasil). **Jusbrasil**. 2023. Disponível em: http://www.jusbrasil.com.br/. Acesso em: 19 mar. 2023.

PACHECO, Juliano. **ML - Aplicações de Machine Learning na área Jurídica, um exemplo com classificação de textos**. Porto Alegre: The Developer'S Conference, [2020]. 47 slides, color. Disponível em: https://s3-sa-east-1.amazonaws.com/thedevconf/presentations/TDC2019POA/machine/ZXJ-8873_2019-12-09T105647_20191128%20-%20TDC%20-%20Aplicacoes%20de%20Machine%20Learning%20na%20area%20Juridica,%20um%20exemplo%20com%20classificacao%20de%20textos.pdf. Acesso em: 08 dez. 2022.

REZENDE, Solange Oliveira. **Sistemas inteligentes**: fundamentos e aplicações. Barueri: Manole, 2003. 525 p.

SANTOS, Thiago. **Linguagem Natural: NLP com análise de sentimento**. 2023. Disponível em: https://www.alura.com.br/curso-online-introducao-a-nlp-com-analise-de-sentimento. Acesso em: 25 mar. 2023.

SÃO PAULO. TRIBUNAL DE JUSTIÇA DO ESTADO DE SÃO PAULO. . **Justiça paulista supera marca de 5 milhões de processos julgados em 2022**. 2023. Disponível em: https://www.tjsp.jus.br/Noticias/Noticia?codigoNoticia=88726. Acesso em: 18 mar. 2023.

SINGH, Ajit. **Processamento de linguagem natural com Python**. Madrid: Babelcube, 2021. 102 p.

THANAKI, Jalaj. **Python Natural Language Processing**. Birmingham: Packt, 2017. 456 p.

UEZONO, Aline Yumi. **Wordcloud com Python**. 2020. Disponível em: https://lineyumi.medium.com/wordcloud-com-python-2cd99e832e8e. Acesso em: 26 abr. 2023.