UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

FACULDADE DE CIÊNCIAS - FC
Campus de Bauru - SP

Bacharelado em Sistemas de Informação

MATHEUS HENRIQUE LOPES

Orientador: Prof. Clayton Reginaldo Pereira

CLASSIFICADORES TEXTUAIS DE DADOS JURÍDICOS

Bauru - SP
2019

UNIVERSIDADE ESTADUAL PAULISTA JÚLIO DE MESQUITA FILHO
FACULDADE DE CIÊNCIAS

Trabalho de Conclusão de Curso

Matheus Henrique Lopes

Classificadores Textuais de Dados Jurídicos

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Sistemas de Informação da Universidade Estadual Paulista "Júlio de Mesquita Filho" como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Clayton Reginaldo Pereira

Bauru - SP
2019

RESUMO

A partir da implementação de algoritmos de *Machine Learning* para classificação multi-classes são avaliadas duas possibilidades de soluções, sendo a primeira para classificar documentos jurídicos chamados de Publicações Jurídicas, dentre 5 categorias judiciais pré-determinadas, e a segunda de predizer o status atual de processos judiciais através da análise dos andamentos processuais mais recentes. Ao alcançar alta porcentagem de assertividade das classificações autônomas, é possível repassar a responsabilidade desta análise de texto dos advogados à Inteligência Artificial, mitigando falhas de interpretação e aumentando a velocidade dos procedimentos em torno das publicações e dos processos jurídicos.

Palavras-chaves: classificador multi-classes, machine learning, classificador textual, classificador jurídico, publicações jurídicas, status de processos

ABSTRACT

From Machine Learning's algorithms implementation to multi-classes' classify will be possible to evaluate two possibilities for AI solutions, the first for classify law documents named as Legal Publications in 5 legal categories and the second one for predict the current status of juridical processes by analyzing the latest legal proceedings. Achieving high percentage assertiveness of autonomous ratings its possible to pass the responsibility of this analysis from lawyers to Artificial Intelligence, mitigating interpretation errors and increasing the speed of procedures around publications.

Keywords: multiclass classification, machine learning, textual classifier, legal classifier, legal publications, status of legal proceedings

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

A advocacia, ou seja, os exercícios executados pelos advogados, é repleta de ações repetitivas e morosas além da sua principal atividade que é: defender o Direito. Estas atividades recorrentes, entretanto, têm como objetivo subsidiar o acompanhamento e evolução jurídica dos processos judiciais e auxiliar na tomada de decisão dos profissionais da área.

Para exemplificar, uma das atividades mais executadas atualmente em escritórios de advocacia é conhecida como “Consulta processual”: trata-se do acompanhamento minucioso da evolução jurídica de todos os processos interessantes à determinada empresa, onde para tal acompanhamento, é necessário realizar consultas públicas em diversos sites do Governo Brasileiro, informando o número de processo por processo para obter os andamentos processuais mais atualizados – como as decisões finais dos juízes responsáveis ou a data e hora da próxima audiência no fórum, por exemplo.

Esse grupo de atividades repetitivas, “braçais” e massivas atrasam o profissional jurídico do compromisso de realmente advogar em torno dos casos de seus clientes – como defender e justificar causas, proteger e acusar partes, pleitear e sustentar ações judiciais e outras.

Devido a isso a tecnologia adentrou no mercado jurídico a fim de criar soluções facilitadoras ao dia a dia dos advogados.

Em relação ao exemplo supracitado, apresenta-se os famosos *crawlers*: robôs que simulam as ações humanas em *websites* para informar e extrair um grande volume de informações presentes nos portais que outrora era obtido através de repetidas consultas manuais por humanos.

O intuito deste trabalho é apresentar duas soluções, com técnicas de *Machine Learning*, com valor e ganho final similar à exemplificada acima. O principal objetivo é predizer e classificar publicações e processos jurídicos, delegando à Inteligência Artificial a usual interpretação de texto realizada até então por advogados e similares.

2. SOLUÇÕES PROPOSTAS

Atualmente, empresas de advocacia ou setores jurídicos de grandes e diversificadas corporações possuem um alto volume diário de documentos públicos a serem analisados e interpretados. Após a análise destes documentos, é realizado o encaminhamento jurídico, interno ou externo à corporação, de acordo com a categoria identificada para cada arquivo. A primeira solução proposta neste trabalho é o classificador textual de publicações jurídicas.

O mesmo esforço existe para identificação do status do processo jurídico, com base nos andamentos processuais mais recentes. As ações dos advogados responsáveis devem ser coerentes e ágeis dependendo de qual etapa o processo encontra-se. O classificador textual de status do processo é a segunda solução construída.

2.1. Classificador textual de Publicações Jurídicas

O foco deste trabalho está nas publicações jurídicas da área de Direito Empresarial, antigo Direito Comercial, responsável pelo estudo das relações privatistas que envolvem a empresa e o empresário. Nessas relações estão o estudo da empresa, o direito societário, as relações de título de crédito, as relações de direito concorrencial, as relações de direito intelectual e industrial e os contratos mercantis.

Publicação jurídica é o ato que traz notoriedade à lei ou garante conhecimento público de um fato. Dentro das etapas de um processo legislativo, a publicação é quando acontece a publicidade da decisão ou de alguma novidade das partes e órgãos envolvidos na ação.

Portanto, existem publicações jurídicas para expor diversos assuntos, cada um com significados judiciais diferentes, sendo alguns exemplos: processo concluso, processo distribuído, carga advogado do autor, vista ao autor e etc.

O algoritmo de Classificação Textual deste trabalho deve classificar autonomamente publicações jurídicas em 6 categorias distintas, sendo estas:

1. Publicação de Insolvência Civil
2. Publicação de Decreto de Falência
3. Publicação de Convolação da Recuperação Judicial em Falência
4. Publicação de Extensão dos efeitos de Falência
5. Publicação de Recuperação Judicial Deferida

6. Publicação de Recuperação Extrajudicial Deferida

Este algoritmo fora estruturado com foco na necessidade da área jurídica de uma instituição financeira brasileira, onde cerca de 20 profissionais da área da Advocacia atuam diariamente lendo aproximadamente 3.000 publicações e classificando-as dentre as 6 categorias listadas anteriormente.

Entretanto, a solução é flexível e adaptável a quaisquer outras categorias interessantes ao mercado e profissionais.

O desenvolvimento do algoritmo utiliza mais de uma técnica de implementação.

No Tópico 5.1 estão descritos os processos e arquiteturas técnicas necessárias para construção da solução.

2.2. Classificador textual de Status de Processos Jurídicos

Em um atendimento e acompanhamento jurídico eficaz é necessário - para processos de todas as áreas do Direito - conhecer e identificar corretamente qual o status atual dos processos.

O status do processo designa em qual etapa da resolução jurídica o processo judicial encontra-se ou qual foi o desfecho, muitas vezes determinado por juízes, do processo.

Nas Figuras 1 e 2 é possível observar os status de dois processos públicos, disponibilizados no portal do Tribunal de Justiça do Estado de São Paulo.

Figura 1 - Exemplo de processo com status “Extinto”

Fonte: Portal eSaj TJSP

Figura 2 - Exemplo de processo com status “Suspenso”

Fonte: Portal eSaj TJSP

A partir da identificação do status as atividades jurídicas devem ser executadas de forma ágil, como por exemplo a confirmação de participação de audiências em fóruns, elaboração de petições, postagem de documentos em cartórios e etc.

Mesmo com dada importância o status nem sempre está indicado nos portais dos Tribunais online de Justiça Brasileira, como evidenciado nas Figuras 1 e 2, demandando aos advogados uma análise e leitura criteriosa dos andamentos processuais, que indicam o histórico e momento atual do processo.

Essa atividade é custosa, principalmente quando deve ser executada em massa, para um grande número de processos jurídicos. Escritórios de pequeno porte costumam atender de 3.000 a 20.000 processos mensalmente, enquanto que escritórios maiores, com 400 colaboradores, trabalham com aproximadamente 550.000 processos ao mês.

Na Tabela 1 é possível observar a proporção entre quantidade de advogados e volume de processos ativos a serem monitorados.

Tabela 1 – Proporção de quantidade de advogados e volume de processos ativos

Quantidade de advogados	**Quantidade de processos analisados mensalmente**
De 6 a 10	3.000
De 25 a 50	20.000
De 100 a 150	160.000
De 400 a 500	550.000

Fonte: Próprio autor

O volume de texto e informações a serem analisadas é altíssimo. Consequentemente, a probabilidade de falhas aumenta.

Portanto, o classificador aqui proposto tem como objetivo predizer e indicar qual a fase do processo, de forma autônoma, com base na análise dos 20 andamentos processuais mais recentes, libertando os advogados e assistentes para atividades que exigem mais raciocínio e intelecto.

Os status elencados para o treinamento do modelo do classificador são:

1. Aguarda partes
2. Arquivado
3. Ativo
4. Baixado
5. Cartório
6. Concluso
7. Extinto
8. Grau recurso
9. Instância superior
10. Julgado
11. Suspenso

Este modelo foi construído com base em um treinamento supervisionado, através de 2 algoritmos estatísticos, conforme exposto tecnicamente no Tópico 5.2.

3. SOLUÇÕES EXISTENTES

Na pesquisa intitulada “Categorização automática de documentos jurídicos utilizando o classificador *Naive-Bayes*” encontra-se a implementação de um algoritmo, também de *Machine Learning*, para uma classificação jurídica mais ampla, categorizando petições em áreas de atuação do Direito: Penal, Trabalhista, Civil, ação de despejo e ação de cobrança.

Nesta pesquisa implementou-se 2 etapas adicionais ao algoritmo de classificação: leitura de documentos via OCR e técnica de *bag-of-words*. Sendo todas as implementações na linguagem Python.

A alta porcentagem de assertividade (próximo aos 80%) indicou a eficiência do uso desse tipo de solução na área jurídica.

Já o trabalho apresentado pelo Engenheiro de Software Everton Gago no evento QCon, São Paulo, em 2016, trata-se da implementação de *Machine Learning* na linguagem Java, com o framework Apache Mahout para classificar notícias em 4 categorias distintas: Política, Religião, Saúde e Esporte. Utilizou-se uma base de treino rotulada e atingiu-se o nível de acurácia de 99,21%, também utilizando o método probabilístico de *Machine Learning Naive-Bayes*.

4. TECNOLOGIAS PESQUISADAS E ADOTADAS

Para o desenvolvimento da solução utilizou-se a linguagem de programação *Python*, em sua versão 3.7. A escolha deve-se a popularização da linguagem para implementações de *Machine Learning*, a baixa curva de adaptação para uso da linguagem e a liberdade existente para definição do paradigma de desenvolvimento. Python é uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. (Python Software Foundation, 2008)

Os códigos foram criados e testados na aplicação web *Jupyter Notebook*, instalado e disponibilizado no *Windows 10*. A utilização do *Jupyter* facilita o gerenciamento de bibliotecas, o compartilhamento de códigos e a visualização de resultados gráficos.

A principal biblioteca do Python utilizada é a *Scikit-learn* (Scikit Learn, 2019) para uso da implementação pré-existente dos algoritmos de aprendizagem de máquina supervisionada e aprendizagem não-supervisionada para classificação multi-classes. A escolha deve-se a abrangente documentação existente, ao uso não licenciado da ferramenta e histórico de utilização pelo orientador.

Inicialmente cogitou-se o uso da Linguagem Java com o framework Apache Mahout porém este foi substituído por Python pela facilidade em manipulação de dados e maior quantidade de opções de bibliotecas de Inteligência Artificial.

5. ESPECIFICAÇÃO TÉCNICA DAS SOLUÇÕES

O processo de construção e desenvolvimento dos algoritmos é dividido em 2 fases, sendo a primeira a etapa de obtenção e tratamento dos dados e a segunda a de treino do modelo, teste de classificação e validação de resultados.

Ambas as soluções são embasadas em classificadores textuais de *Machine Learning*, porém com métodos de treinamento distintos devido às características dos *datasets*.

Uma opção de construção de um classificador é utilizar a abordagem de aprendizagem, ou também chamado de treinamento, supervisionado.

A aprendizagem supervisionada, no qual dado um conjunto de observações ou exemplos rotulados, isto é, conjunto de observações em que a classe, denominada também atributo meta, de cada exemplo é conhecida, o objetivo é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes. (Russell & Norvig, 2003)

Outro processo sugerido pela literatura de Aprendizado de Máquina é o treinamento não-supervisionado, no qual dado um conjunto de observações ou exemplos não rotulados, o objetivo é tentar estabelecer a existência de grupos ou similaridades nesses exemplos. (Russell & Norvig, 2003)

Nos Sub-tópicos 5.1 e 5.2 estão os cenários de implementação e os diagramas de fases de cada classificador proposto neste trabalho, abordando os detalhes de forma minuciosa.

Os resultados obtidos por cada algoritmo proposto estarão evidenciados no Tópico 6.

5.1. Classificador textual de Publicações Jurídicas

Como mencionado no Tópico 2.1, foram implementadas duas formas de obter um modelo de classificação de publicações jurídicas. A primeira sendo através do treinamento supervisionado e a segunda através do treinamento não-supervisionado, mais especificamente, através da técnica de clusterização.

A justificativa para existir a segunda implementação utilizando o treinamento não-supervisionado deve-se ao resultado negativo do classificador com treinamento supervisionado, o qual será melhor evidenciado nos Tópicos 5.1.2 e 6.1.1.

Para ambas as abordagens, existiu a etapa de obtenção dos dados, ou *dataset*, descrito no Tópico 5.1.1 abaixo.

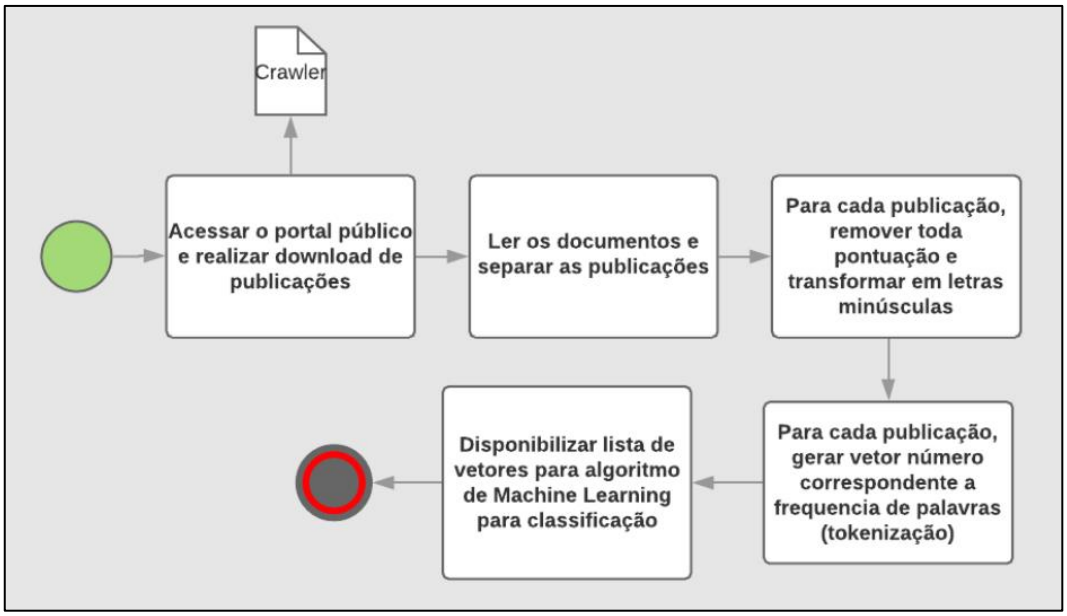
5.1.1. Primeira etapa: obtenção dos dados

Esta etapa contempla o processo de obter uma massa de dados (rotulada ou não), realizar o tratamento de limpeza e padronização textual e geração dos vetores numéricos. Os vetores são as representações numéricas dos textos que servem de *input* para o início do algoritmo de treinamento. A conversão deve-se ao fato da máquina, e consequentemente do algoritmo, ser capaz de entender apenas números.

Para o classificador de publicações jurídicas acessamos um repositório público e online de publicações diversas e realizamos o *download* dos arquivos para leitura e as demais operações de texto, como a formatação para letras minúsculas, remoção de pontuação e etc.

É possível observar os passos executados para obtenção do *dataset* no diagrama da Figura 3.

Figura 3 - Passos para obtenção e tratamento do *dataset*



Fonte: Próprio autor

Ao final do processo, obteve-se 2.470 documentos e 299.148 publicações jurídicas distintas, uma vez que cada documento possui um número de publicações variado.

Todas as publicações são não-rotuladas, ou seja, as categorias jurídicas as quais elas pertencem são desconhecidas.

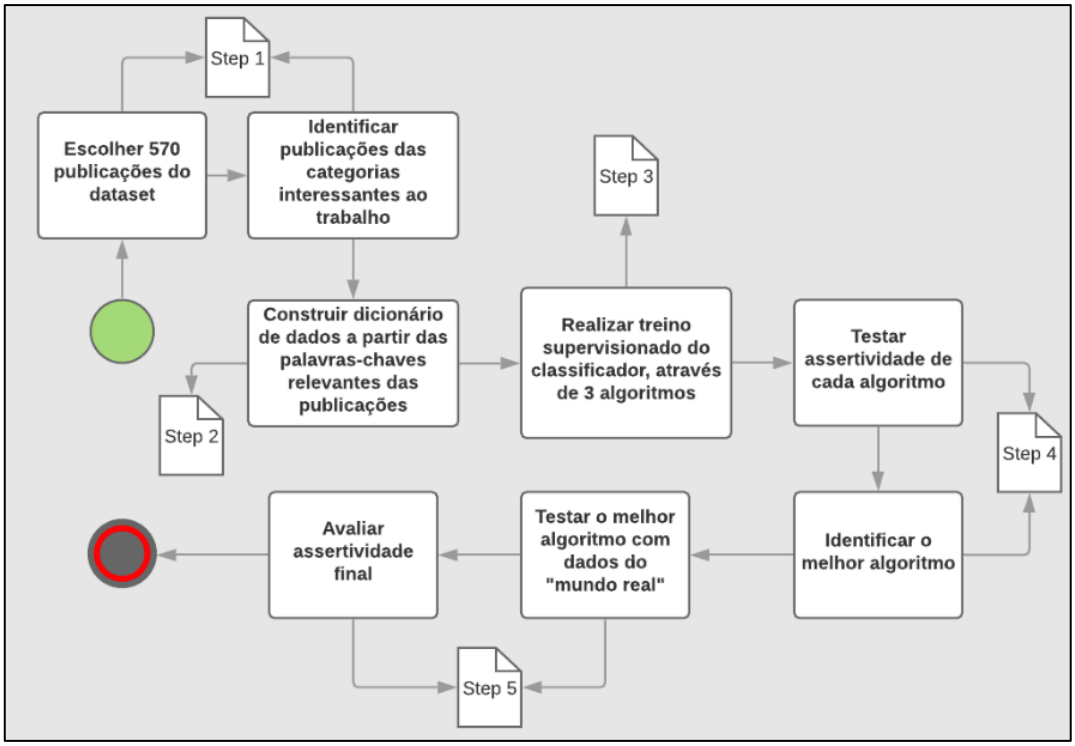
Com o *dataset* disponível os próximos passos são os treinamentos, descritos nos Tópicos 5.1.2 e 5.1.3.

5.1.2. Segunda etapa: Cenário a - treinamento supervisionado

Para melhor referenciação no momento de expor os resultados obtidos deste trabalho, vamos nomear este tópico como “Publicações - Supervisionado”. Neste tópico, estamos tratando o desafio de classificar publicações jurídicas através de treinamento supervisionado.

Na Figura 4 está o passo-a-passo para obtenção do modelo de classificação.

Figura 4 - Passos para classificação textual a partir de treinamento supervisionado



Fonte: Próprio autor

O processo do treinamento supervisionado, exposto na Figura 4, pode ser descrito em 5 *steps*, sendo eles:

1. Obtenção de base de dados rotulada
2. Criação de dicionário de palavras
3. Treino do modelo de classificação supervisionada
4. Testes do modelo treinado

5. Validação do melhor modelo de classificação obtido

No *step* 1, para obtenção da base de dados rotulada, foram analisadas cerca de 570 publicações jurídicas, advindas aleatoriamente do *dataset* obtido, de forma tradicional (leitura humana) para identificação de exemplos de publicações das categorias jurídicas apresentadas no Tópico 2.1.

A interpretação das publicações fora validada por um advogado com conhecimento técnico da área jurídica conhecida por Direito Empresarial.

Após a análise humana, apenas 18 publicações foram classificadas dentre as 5 categorias jurídicas interessantes.

Portanto, o *dataset* rotulado para o treinamento supervisionado possui apenas 18 elementos.

A construção de um dicionário de palavras, realizada no *step* 2, é etapa necessária do processo de uma classificação textual supervisionada.

Este dicionário é composto pelas principais palavras-chaves que estarão contidas nos documentos - neste caso nas publicações jurídicas - a serem classificados.

No desenvolvimento da solução deste trabalho, o dicionário fora construído com base em uma lista de palavras-chaves pré-definida por profissionais responsáveis por classificar publicações jurídicas em uma empresa conhecida pelo autor deste trabalho, dispensando o uso da técnica *bag-of-words*.

É possível visualizar o dicionário, em sua versão final após tratamento das palavras, na Figura 5.

Figura 5 - Dicionário de palavras utilizado para classificação textual

```
set(['declarar', 'convolacao', 'deferido', 'acolho', 'efeitos', 'extens
ao', 'pedido', 'tribuna', 'administrador', 'declaro', 'decretacao', 'ex
trajudicial', 'convoco', 'habilitacoes', 'concedo', 'recuperacao', 'nom
eio', 'revogo', 'justica', 'habilitacao', 'acatado', 'deferida', 'decla
rada', 'massa', 'convocacao', 'homologo', 'decreto', 'extinto', 'proces
samento', 'estendo', 'multipla', 'edital', 'expeca-se', 'concedido', 'f
alencia', 'regional', 'efeito', 'defiro', 'falida', 'declaracao', 'conc
edida', 'converto', 'habilito', 'liquidacao', 'judicial', 'convolo', 'a
cato', 'posto', 'procedente', 'decretada', 'encerramento', 'falencias',
 'trabalho', 'decretado', 'julgo', 'insolvencia'])
```

Fonte: Próprio autor

Para a etapa de treino (*step* 3) utilizou-se 80% dos dados rotulados (13 publicações) e o restante dos dados foram divididos entre o teste (3 publicações) e a validação (2 publicações).

Os algoritmos de classificação utilizados foram:

1. *OneVsRest* (com o modelo LinearSVC)
2. *OneVsOne* (com o modelo LinearSVC)
3. *Naive Bayes* MultinomialNB
4. *AdaBoost Classifier*

O primeiro algoritmo, *OneVsRest*, considera uma das classes como “positiva”, o restante das classes como “negativas” e realiza o treinamento do classificador. Portanto, para um conjunto de dados que possuem n classes, o algoritmo gera n classificadores treinados. Essa estratégia exige que os classificadores de base produzam uma pontuação de confiança com valor real para sua decisão, ao invés de retornar apenas rótulos de classes; quando há somente rótulos de classe discretos pode-se gerar ambiguidades, onde são previstas várias classes para uma única amostra a ser classificada. (Bishop, 2006)

O segundo algoritmo, *OneVsOne*, gera K (K − 1) / 2 classificadores binários para um contexto de K classes; cada classificador recebe exemplos de elementos de 2 classes do conjunto de K classes e aprende a distinguir essas 2 classes especificamente. No momento dos testes e previsões, é aplicada uma lógica de votação: todos os K (K − 1) / 2 classificadores são aplicados a um elemento desconhecido e a classe que obter o maior número de previsões “+1” é a classe indicada pelo classificador. (Bishop, 2006)

Já o terceiro algoritmo, *Naive Bayes*, é um classificador com base no princípio da máxima a posteriori (PAM). Trata-se de um algoritmo simples, que utiliza dados históricos para prever a classificação de um novo dado. Este algoritmo calcula a probabilidade de um evento ocorrer dado que outro evento já ocorreu. Ele é denominado ingênuo (*naive*) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa sobre nenhum outro. (Amaral, 2006)

Por último utilizou-se o *AdaBoost*, inventado por Yoav Freund e Robert Schapire, que combina de forma ponderada vários classificadores fracos com taxa de acerto acima de 50% para obter um classificador forte. É um algoritmo meta-heurístico, e pode ser utilizado para aumentar a performance de outros algoritmos de aprendizagem. O nome "*AdaBoost*" deriva de *Adaptive Boosting* (em português, impulso ou estímulo adaptativo). (Freund e Schapire, 1997)

Todos os algoritmos foram implementados através da biblioteca *Sckit Learn*, em Python.

Os resultados do treino dos modelos, obtidos no *step* 4, estão na Tabela 2.

Tabela 2 - Taxa de acerto dos algoritmos de classificação

ALGORITMO DE CLASSIFICAÇÃO	TAXA DE ACERTO (%)
OneVsRest (modelo LinearSVC)	38,75
OneVsOne (modelo LinearSVC)	32,5
Naive Bayes (MultinomialNB)	28,75
AdaBoost	21,25

Fonte: Próprio autor

Para a etapa de validação (*step* 5) foi escolhido apenas o algoritmo com maior taxa de acerto, sendo o *OneVsRest*.

A etapa de validação retorna a porcentagem de assertividade do algoritmo baseada em novos dados que não serão testados em outros algoritmos considerados anteriormente. Esses dados são chamados de “dados do mundo real”. A taxa de acerto na validação do algoritmo *OneVsRest* é 40%.

Esta é a porcentagem de assertividade esperada do algoritmo *OneVsRest* para a classificação de publicações “do mundo real”.

A análise e hipótese do motivo da assertividade deste modelo ser baixa está presente na seção de resultados, especificamente no Tópico 6.1.1.

5.1.3. Segunda etapa: Cenário b - treinamento não-supervisionado

Este cenário, nomeado por “Publicações - Não-supervisionado”, existe devido a baixa assertividade do modelo construído através de treinamento supervisionado.

Para o treinamento não-supervisionado foram filtradas 7.288 (2,44%) publicações do *dataset* de 299.148 elementos, que possuem as sentenças-chaves identificadas nos exemplos das seis categorias de publicações, obtidos da empresa que realiza a classificação manual do material. Esse filtro ocorreu através da execução de um script em Python nomeado por “pré-classificador”.

As sentenças-chaves utilizadas pelo “pré-classificador” estão na Tabela 3.

Tabela 3 - Sentenças-chaves identificadoras das categorias das publicações

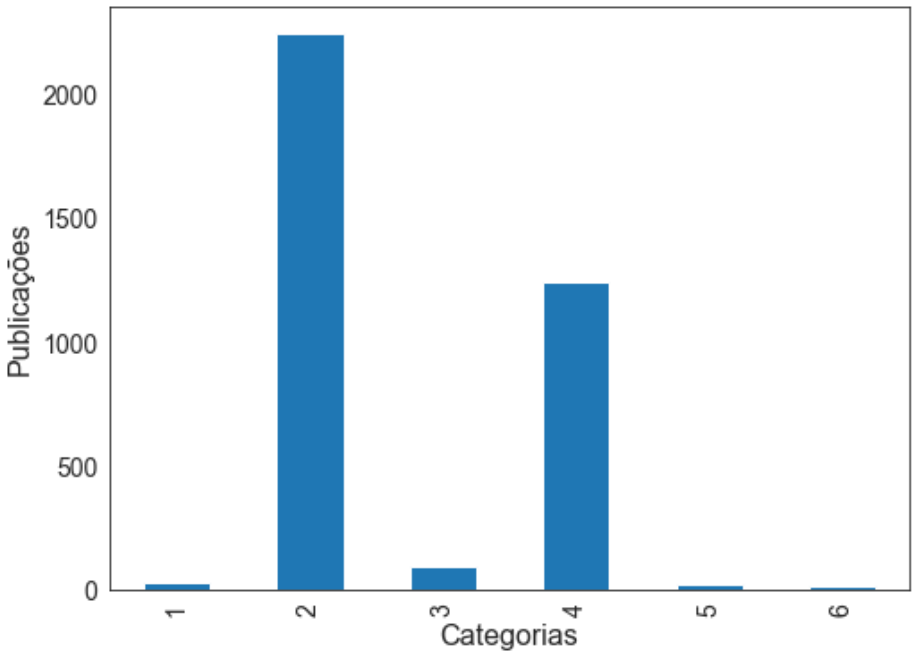
CATEGORIA	SENTENÇAS-CHAVES
1-Insolvência Civil	“declaro a insolvencia” ou “declaracao de insolvencia”
2-Decreto de Falência	“decretacao de falencia” ou “decretada a falencia”
3-Convolação da Recuperação Judicial em Falência	“convolacao da recuperacao judicial em falencia” ou (“convolacao” e “recuperacao judicial” e “falencia”)
4-Extensão dos efeitos de Falência	“extensao dos efeitos da falencia” ou (“extensao dos efeitos” e “falencia” e “recuperacao judicial”)
5-Recuperação Judicial Deferida	“processamento da recuperacao judicial”
6-Recuperação Extrajudicial Deferida	“decretada a liquidacao extrajudicial” ou (“recuperacao extrajudicial” e “decretad” e “falencia”)

Fonte: Próprio autor

É importante ressaltar que, mesmo encontrando as sentenças-chaves nas publicações não é garantia que a publicação pertence àquela respectiva categoria, uma vez que não houve análise técnica jurídica e os termos do dialeto jurídico são recorrentemente utilizados em mais de um contexto.

A Figura 6 demonstra o gráfico de barras com a distribuição do número de publicações para cada uma das seis categorias, definidas pelas sentenças da Tabela 2 e encontradas pelo script “pré-classificador”. É possível observar um grande número de publicações com sentenças-chaves extraídas das categorias 2 e 4.

Figura 6 – Gráfico de barras com a quantidade de publicações por categoria (definidas pelo pré-classificador)



Fonte: Próprio autor

Com o *dataset* construído, aplicou-se o processo de clusterização.

Técnicas de clusterização inserem observações em grupos (clusters), de forma que as similaridades sejam grandes entre observações pertencentes a um mesmo cluster e diferentes das inseridas em outro (Hair et al., 1995; Agard & Penz, 2009). Existem dois conjuntos de técnicas usualmente utilizados para clusterização de observações: hierárquicos e não hierárquicos (Hair et al., 1995). Dentre os não hierárquicos, destaca-se o *k-means*, utilizado neste trabalho, o qual agrupa as observações em k clusters, sendo k um valor previamente conhecido para o algoritmo (Kaufman & Rousseeuw, 2005).

O número escolhido para o parâmetro k é 6, devido a quantidade de categorias jurídicas relevantes para a pesquisa.

Importante mencionar que, para visualização dos resultados após a aplicação da clusterização, são necessárias técnicas de redimensionamento dos dados, uma vez que o output do algoritmo inicialmente é multidimensional e para exibi-los nos gráficos é necessário torná-lo bidimensional. As técnicas utilizadas neste trabalho são Análise de Componentes Principais (ACP ou PCA, do inglês, Principal Component Analysis) e t-SNE (do inglês, t-Distributed Stochastic Neighbor Embedding).

A análise de componentes principais (PCA) é uma técnica multivariada de modelagem da estrutura de covariância. A técnica foi inicialmente descrita por Pearson (1901) e uma descrição de métodos computacionais práticos veio muito mais tarde com Hotelling (1933, 1936) que usou com o propósito determinado de analisar as estruturas de correlação. A PCA é uma técnica estatística de análise multivariada que transforma linearmente um conjunto original de variáveis, inicialmente correlacionadas entre si, num conjunto substancialmente menor de variáveis não correlacionadas que contém a maior parte da informação do conjunto original.

A ACP é a técnica mais conhecida e está associada à ideia de redução de massa de dados, com menor perda possível da informação, contudo é importante ter uma visão conjunta de todas ou quase todas as técnicas da estatística multivariada para resolver a maioria dos problemas práticos, também é associada à ideia de redução de massa de dados, com menor perda possível da informação. Procura-se redistribuir a variação observada nos eixos originais de forma a se obter um conjunto de eixos ortogonais não correlacionados (Manly, 1986; Hongyu, 2015).

Já a técnica t-SNE, apresentada por Maaten e Hinton em 2008, trata-se de um método que utiliza modelos probabilísticos para encontrar o mapeamento das instâncias do espaço multidimensional em um espaço de baixa dimensionalidade. Esta técnica é capaz de capturar estruturas locais presentes nos dados e também revelar estruturas globais, tais como a presença de grupos de instâncias similares.

O processo de clusterização fora aplicado duas vezes e os resultados serão apresentados no Tópico 6.1.2.

5.2. Classificador textual de Status de Processos Jurídicos

Para o desenvolvimento do modelo do classificador de forma supervisionada, primeiramente ocorreu a obtenção e tratamento do *dataset*, para então realizar a aplicação de algoritmos probabilísticos e construção do modelo.

Nos Tópicos 5.2.1 e 5.2.2 estão descritos os processos de pré-processamento e de processamento do algoritmo de *Machine Learning*, respectivamente.

5.2.1. Primeira etapa: obtenção dos dados

O *dataset* deste classificador fora construído através de um *crawler*

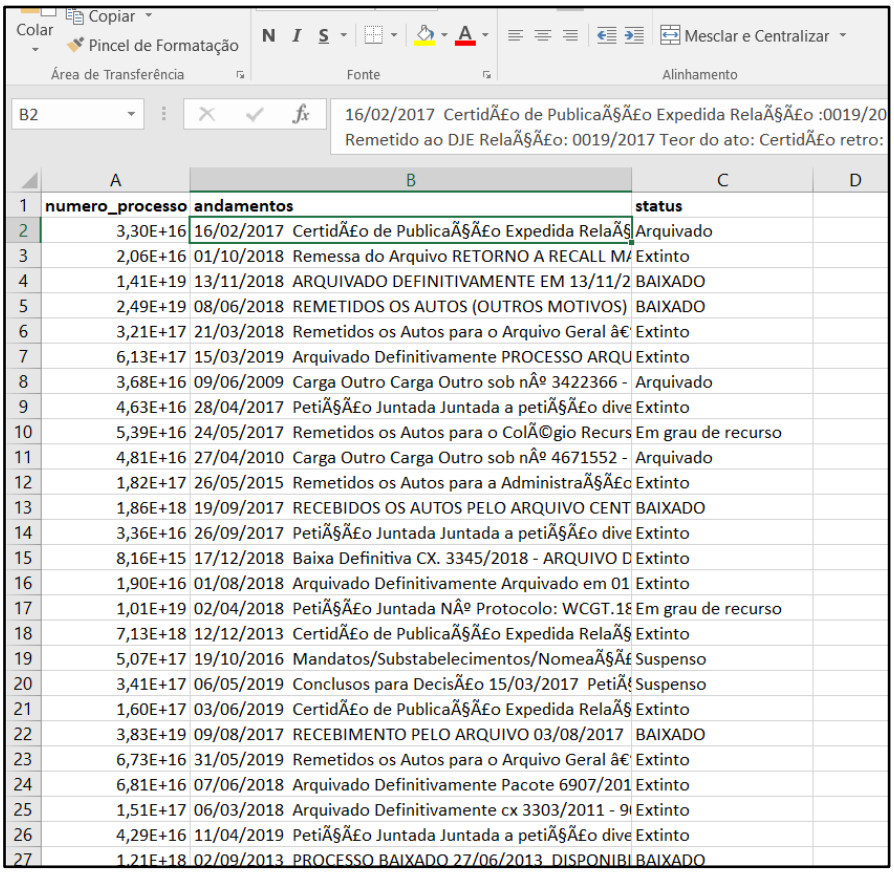
integrado a diversos portais jurídicos do Brasil, onde, para cada portal, fora pesquisado processos diferentes que já possuíam seus status definidos e atualizados, informados pelos respectivos juízes responsáveis.

Importante mencionar que nem todos os portais jurídicos apresentam o status do processo e muitos apresentam de forma muito tardia. É neste cenário que o classificador proposto agrega o valor aos advogados.

As informações colhidas pelo *crawler* são: número do processo; os 20 últimos andamentos processuais disponíveis para leitura e o status do processo.

Todos os 20 andamentos processuais foram armazenados na mesma coluna, concatenados utilizando espaço entre eles. As colunas do *dataset* estão visíveis na Figura 7. A formatação de acentuação fora aplicada na leitura do arquivo.

Figura 7 - *Dataset* rotulado de processos e status

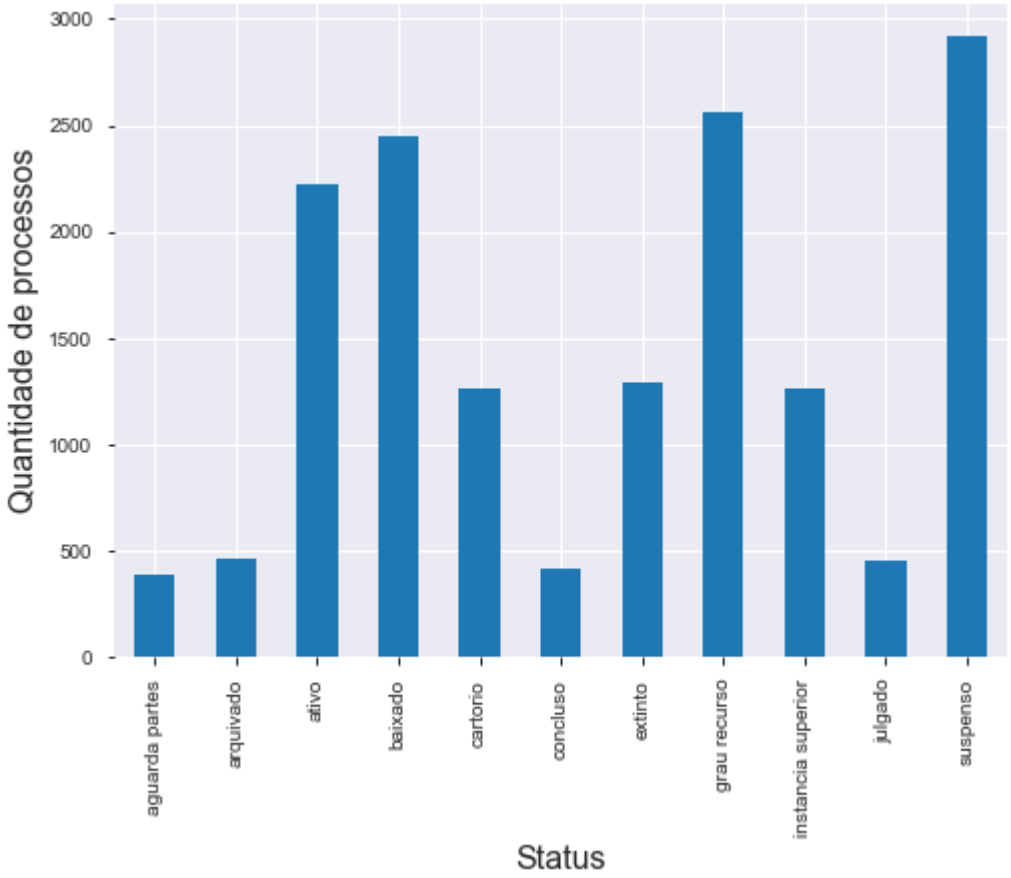


	A	B	C	D
1	numero_processo	andamentos	status	
2	3,30E+16	16/02/2017 CertidÃ£o de PublicaÃ§Ã£o Expedida RelaÃ§	Arquivado	
3	2,06E+16	01/10/2018 Remessa do Arquivo RETORNO A RECALL MA	Extinto	
4	1,41E+19	13/11/2018 ARQUIVADO DEFINITIVAMENTE EM 13/11/2	BAIXADO	
5	2,49E+19	08/06/2018 REMETIDOS OS AUTOS (OUTROS MOTIVOS)	BAIXADO	
6	3,21E+17	21/03/2018 Remetidos os Autos para o Arquivo Geral â€	Extinto	
7	6,13E+17	15/03/2019 Arquivado Definitivamente PROCESSO ARQU	Extinto	
8	3,68E+16	09/06/2009 Carga Outro Carga Outro sob nÂº 3422366 -	Arquivado	
9	4,63E+16	28/04/2017 PetiÃ§Ã£o Juntada Juntada a petiÃ§Ã£o dive	Extinto	
10	5,39E+16	24/05/2017 Remetidos os Autos para o ColÃ©gio Recurs	Em grau de recurso	
11	4,81E+16	27/04/2010 Carga Outro Carga Outro sob nÂº 4671552 -	Arquivado	
12	1,82E+17	26/05/2015 Remetidos os Autos para a AdministraÃ§Ã£o	Extinto	
13	1,86E+18	19/09/2017 RECEBIDOS OS AUTOS PELO ARQUIVO CENT	BAIXADO	
14	3,36E+16	26/09/2017 PetiÃ§Ã£o Juntada Juntada a petiÃ§Ã£o dive	Extinto	
15	8,16E+15	17/12/2018 Baixa Definitiva CX. 3345/2018 - ARQUIVO D	Extinto	
16	1,90E+16	01/08/2018 Arquivado Definitivamente Arquivado em 01	Extinto	
17	1,01E+19	02/04/2018 PetiÃ§Ã£o Juntada NÂº Protocolo: WCGT.18	Em grau de recurso	
18	7,13E+18	12/12/2013 CertidÃ£o de PublicaÃ§Ã£o Expedida RelaÃ§	Extinto	
19	5,07E+17	19/10/2016 Mandatos/Substabelecimentos/NomeaÃ§Ã£	Suspenso	
20	3,41E+17	06/05/2019 Conclusos para DecisÃ£o 15/03/2017 PetiÃ§	Suspenso	
21	1,60E+17	03/06/2019 CertidÃ£o de PublicaÃ§Ã£o Expedida RelaÃ§	Extinto	
22	3,83E+19	09/08/2017 RECEBIMENTO PELO ARQUIVO 03/08/2017	BAIXADO	
23	6,73E+16	31/05/2019 Remetidos os Autos para o Arquivo Geral â€	Extinto	
24	6,81E+16	07/06/2018 Arquivado Definitivamente Pacote 6907/201	Extinto	
25	1,51E+17	06/03/2018 Arquivado Definitivamente cx 3303/2011 - 9	Extinto	
26	4,29E+16	11/04/2019 PetiÃ§Ã£o Juntada Juntada a petiÃ§Ã£o dive	Extinto	
27	1,21E+18	02/09/2013 PROCESSO BAIXADO 27/06/2013 DISPONIBI	BAIXADO	

Fonte: Próprio autor

Ao final da execução do *crawler* e captura das informações, obteve-se 15.698 processos rotulados, distribuídos entre os 11 diferentes status (listados no Tópico 2.2), como está no gráfico da Figura 8.

Figura 8 – Gráfico de barras com a quantidade de processos para cada status considerado



Fonte: Próprio autor

Dessa forma, o *dataset* disponível para classificação dos status dos processos é rotulado, permitindo o treinamento supervisionado do modelo, explicitado no Tópico 5.2.2.

5.2.2. Segunda etapa: treinamento do modelo

Devido a existência de um *dataset* rotulado, a técnica de treinamento supervisionado é a mais indicada para atingir a melhor assertividade possível. Não é necessário aplicar o treinamento não-supervisionado.

São separados 20% do *dataset* para servir como processos de teste dos modelos gerados. Em números absolutos são 3.140 processos com seus status.

Os andamentos processuais foram transformados de texto para uma matriz de *tokens* através do módulo *CountVectorizer* da biblioteca SkLearn. Com a matriz de *tokens*, obteve-se a matriz de frequência de termos utilizando o módulo *TfidfTransformer*, também da biblioteca SkLearn.

Os dois algoritmos utilizados para obtenção dos modelos de classificação foram *Support Vector Machine* (SVM) e *Naive-Bayes* (NB). As porcentagens de assertividade dos modelos obtidos foram diferentes e estão expostas no Tópico 6.2.

6. RESULTADOS DAS SOLUÇÕES

Como especificado no Tópico 5, foram realizadas 3 implementações diferentes, sendo duas delas para o classificador de publicações e uma para o classificador de status de processos.

Nesta seção estão os resultados dos testes dos modelos obtidos, a discussão técnica de cada resultado e algumas hipóteses dos motivos dos níveis de assertividades encontrados.

6.1. Classificador de Publicações Jurídicas

Os resultados dos classificadores de publicações estão separados para os dois cenários propostos e implementados. Primeiramente é apresentado o resultado do modelo construído de forma supervisionada e em seguida, o modelo obtido através de clusterização.

6.1.1. Cenário “Publicações - supervisionado”

Como antecipado no Tópico 5.1.2, a maior assertividade obtida através do treinamento supervisionado de publicações foi de 40%. Trata-se de uma assertividade baixa, uma vez que de acordo com a literatura, bons resultados de algoritmos de Inteligência Artificial estão acima de 80%, sempre variando de acordo com o problema abordado.

Essa porcentagem insatisfatória está atribuída ao tamanho do *dataset* utilizado. Como a única abordagem proposta para a rotulagem dos dados foi a análise humana das publicações, não foi possível atingir um número de amostras rotuladas suficiente para treinamento de um modelo.

Portanto, é possível concluir que, para o estado atual do *dataset*, o treinamento supervisionado não é uma boa opção para construção de um classificador textual de publicações eficaz.

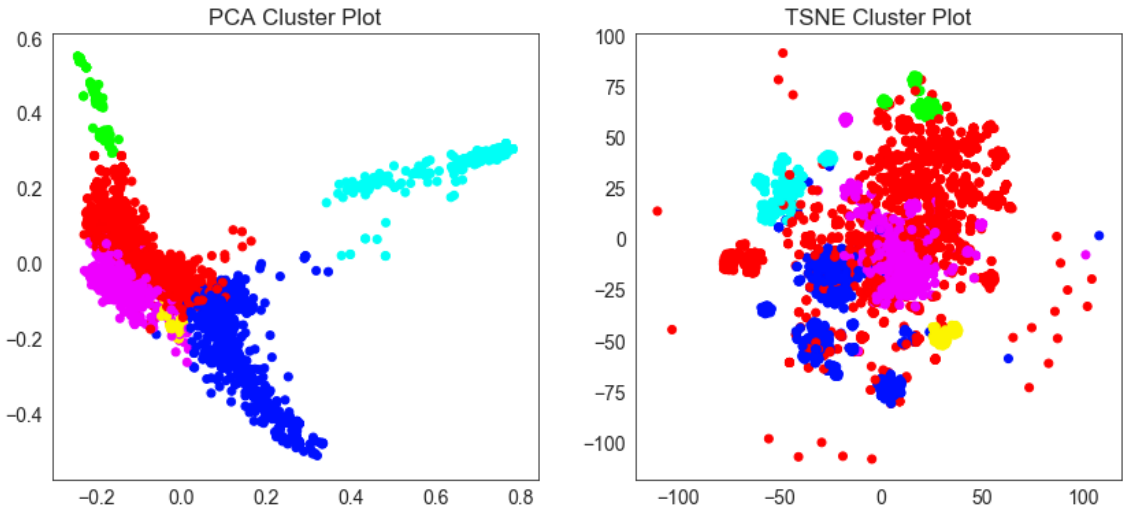
6.1.2. Cenário “Publicações - não-supervisionado”

De acordo com a Especificação Técnica do Tópico 5.1.3, as 7.288 publicações foram importadas para o algoritmo de treinamento não-supervisionado.

O tempo para a execução completa do algoritmo e renderização dos gráficos, após aplicação das técnicas de PCA e t-SNE, é de aproximadamente 5 minutos.

Na Figura 9 estão os dois gráficos, do tipo *Scatter*, exibindo a distribuição dos elementos nos 6 clusters definidos pela máquina.

Figura 9 - Resultado da clusterização das 7.288 publicações, após aplicação de PCA e t-SNE



Fonte: Próprio autor

A proximidade dos pontos nos gráficos demonstra a dificuldade de obter-se *features* exclusivas de determinados clusters. Estima-se que essa situação se deve às repetições de alguns termos jurídicos para diferentes tipos de publicações, em muitos momentos devido a necessidade dos advogados retomarem o histórico do processo ao produzirem novas publicações com novidades ou desfechos judiciais.

Para medir o nível de assertividade da clusterização, é necessária a análise jurídica de amostras de publicações de cada cluster definido pela máquina. Esta tal análise demanda a participação de um profissional (advogados) com conhecimento técnico do Direito Empresarial.

Não fora possível localizar um patrocinador em tempo de desenvolvimento deste projeto para avaliar a assertividade deste algoritmo, impossibilitando a confirmação se cada grupo definido possui apenas publicações da mesma categoria,

ou se, pelo menos, possui algum significado jurídico relevante para os respectivos agrupamentos.

Portanto, neste estágio da solução ainda não é possível implementá-la para agilizar o trabalho dos advogados de forma autônoma. Caso a assertividade seja avaliada e confirmada, o algoritmo poderá atuar no mercado realizando a primeira triagem de uma massa de publicações não-rotuladas, diminuindo o esforço investido pelos profissionais para análise e interpretação dos textos.

6.2. Classificador de Status de Processos

Foram obtidos dois modelos treinados, com base nos algoritmos *Naive Bayes* e *Support Vector Machine*.

6.2.1. Algoritmo *Naive Bayes*

O primeiro modelo obtido apresentou uma assertividade de 66,11% para predição do status de um processo após a análise dos andamentos processuais mais recentes.

O tempo de execução deste algoritmo é de 2,3 segundos.

A Tabela 4 demonstra os valores das métricas de avaliação do classificador.

Tabela 4 - Métricas do modelo *Naive Bayes*

STATUS	PRECISÃO	RECALL
aguarda partes	0	0
arquivado	0,98	0,41
ativo	0,86	0,91
baixado	0,53	0,76
cartorio	0	0
concluso	0	0
extinto	0,98	0,21
grau recurso	0,56	1
instancia superior	0,95	0,79
julgado	0,87	0,56
suspenso	0,65	0,77

Fonte: Próprio autor

A fim de buscar por melhores resultados, é proposto, para os próximos passos deste trabalho, realizar o balanceamento dos dados, gerando um novo *dataset* onde o número de exemplos de processos para cada status é de 390 elementos. Importante salientar que a amostra de dados para teste também deve estar balanceada.

Assim, estima-se mitigar a possibilidade de construção de um modelo tendencioso à algum status, devido a maior volumetria de exemplos do mesmo.

6.2.2. Algoritmo *Support Vector Machine*

Para o algoritmo SVM (sigla para *Support Vector Machine*) obteve-se um modelo de classificação com assertividade maior se comparado ao modelo de *Naive Bayes*, atingindo até 83,38% de acurácia na identificação dos status dos processos.

O tempo de execução deste algoritmo é de 2,88 segundos.

Como é possível observar na Tabela 5, os tópicos com maior precisão são “aguarda partes”, “arquivado”, “grau recurso” e “instancia superior”, respectivamente.

Já os com maior recall são “grau recurso”, “extinto”, “ativo” e “instancia superior”.

Tabela 5 - Métricas do modelo *Support Vector Machine*

STATUS	PRECISÃO	RECALL
aguarda partes	1	0,01
arquivado	0,92	0,74
ativo	0,89	0,95
baixado	0,89	0,88
cartorio	0,56	0,56
concluso	0,67	0,02
extinto	0,86	0,95
grau recurso	0,9	0,99
instancia superior	0,9	0,93
julgado	0,84	0,81
suspenso	0,75	0,86

Fonte: Próprio autor

O fato de existirem classes em que o recall é baixo (como “aguarda partes” e “concluso”) indicam que o modelo não é assertivo para todas as classes. É possível observar que quanto menor o tamanho da amostra da classe, menor é o recall atribuído a ela.

Para aumento da assertividade deste modelo, são sugeridas duas estratégias:

- Remoção das classes com quantidade de exemplos inferior a 1.000 elementos
- Balanceamento do *dataset* de treino e de teste para no máximo 390 elementos por classe

Porém, no estágio atual do modelo, é possível disponibilizar a solução para um grupo de advogados usarem em ambiente real de trabalho, testando a solução com outros exemplos de processos, para avaliar a assertividade final do algoritmo.

7. CONCLUSÃO

Neste trabalho foi proposto a criação de dois classificadores textuais para apoiar a análise de dados do contexto jurídico brasileiro. O primeiro para realizar a identificação da categoria de publicações do âmbito do Direito Comercial e o segundo para predizer o status do processo através da análise dos andamentos processuais mais recentes.

Assim como evidenciado em outros trabalhos e pesquisas, os dados jurídicos permitem a aplicação de técnicas de Inteligência Artificial e a obtenção de bons resultados.

Neste trabalho em questão, fora possível observar que, para publicações jurídicas, o pré-processamento e a existência de um *dataset* rotulado é fundamental para construção de um bom classificador.

A clusterização, técnica de treinamento não supervisionado aplicada, facilita a triagem das publicações, ao agrupá-las por suas semelhanças, mas ainda é necessário a intervenção humana para identificação dos grupos de documentos criados.

Uma possível melhoria neste desenvolvimento seria utilizar uma plataforma colaborativa para a construção de uma base de dados rotulada, onde vários modelos de publicações seriam disponibilizados a advogados da área para classificarem uma massa de dados e então aplicarmos o treinamento supervisionado.

Em relação a predição do status dos processos o resultado foi satisfatório, uma vez que com o algoritmo SVM atingiu-se mais de 80% de acurácia, permitindo que os advogados utilizem dessa solução para identificarem o status de processos sem a necessidade de lerem todos os andamentos processuais.

Os próximos passos para este classificador é a inclusão de novos status jurídicos, a serem obtidos com profissionais da área, habituados a acompanharem o andamento jurídico das ações judiciais no Brasil.

Portanto, é possível concluir que a adoção de técnicas de Inteligência Artificial ao âmbito do Direito Brasileiro, trazem resultados promissores, principalmente na era da Digitalização, onde todos os dados e as informações estão migrando para o ambiente virtual.

8. REFERÊNCIAS BIBLIOGRÁFICAS

INFOQ, 2016, São Paulo. **Classificação de documentos baseada em Inteligência Artificial (IA) [...]. [*S. l.*: *s. n.*],** 2016. Disponível em: https://www.infoq.com/br/presentations/classificacao-de-documentos-baseada-em-inteligencia-artificial. Acesso em: 31 mar. 2019.

OLIVEIRA, CRISTIANO CESAR. Categorização Automática De Documentos Jurídicos Utilizando o Classificador Naive-Bayes. 2015. **Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO, CAMPOS DO JORDÃO**, 2015. Disponível em: cristianocesar.com.br/Base_controller/download/1. Acesso em: 26 mar. 2019.

AMARAL, F., 2016. **Aprenda Mineração de Dados: Teoria e prática**. São Paulo: Alta Books, 2016. Disponível em: http://www.altabooks.com.br/index.php?dispatch=attachments.getfile&attachment_id=2636. Acesso em: 05 dez. 2019.

PYTHON Software Fundation. [*S. l.*], 2008. Disponível em: https://www.python.org/psf/. Acesso em: 20 jun. 2019.

PYCHARM JetBrains. [*S. l.*], 2019. Disponível em: https://www.jetbrains.com/pycharm/. Acesso em: 20 jun. 2019.

SCIKIT LEARN, **Machine Learning in Python**. [S. l.], 2019. Disponível em: https://scikit-learn.org/stable/index.html. Acesso em: 23 jun. 2019.

RUSSEL, S. & NORVIG, P., 2003. **Artificial Intelligence – A Modern Approach.Prentice Hall, 2 edition**. Disponível em: https://bit.ly/2KFbOFX. Acesso em: 27 jul. 2019.

BISHOP, CHRISTOPHER M., 2006. **Pattern Recognition and Machine Learning. Springer**. Disponível em: https://bit.ly/2Riow1t. Acesso em: 03 dez. 2019.

KAUFMAN, L., & ROUSSEEUW, P., 2005. **Finding Groups in Data: an Introduction to Cluster Analysis**. New Jersey: Wiley Interscience. Disponível em: http://eprints.bournemouth.ac.uk/20910/1/Boryczka2009.pdf. Acesso em: 02 ago. 2019.

HAIR, J., ANDERSON, R., TATHAM, R. & BLACK, W., 1995. **Multivariate Data Analysis with Readings (4th ed.).** New Jersey: Prentice-Hall Inc. Disponível em: https://is.muni.cz/el/1423/podzim2017/PSY028/um/_Hair_-_Multivariate_data_analysis_7th_revised.pdf. Acesso em: 06 out. 2019.

MANLY, B. F. J., 1986. **Multivariate statistical methods**. New York, Chapman and Hall, 1986.159 p. Disponível em: http://120.107.155.180/download/Statistics/Multivariate%20Statistical%20Methods%20A%20Primer,%20Fourth%20Edition.pdf. Acesso em: 06 out. 2019.

HONGYU, K. **Comparação do GGEbiplot ponderado e AMMI-ponderado com outros modelos de interação genótipo × ambiente**. 2015. 155p. Tese (Doutorado em Estatística e Experimentação Agronômica) - Escola Superior de Agricultura “Luiz de Queiroz”, Universidade de São Paulo, Piracicaba, 2015. Disponível em: https://www.teses.usp.br/teses/disponiveis/11/11134/tde-04052015-172304/publico/Kuang_Hongyu.pdf. Acesso em: 12 nov. 2019.

MAATEN, L. VAN DER; HINTON, G. 2008. **Visualizing data using t-SNE**. Journal of Machine Learning Research, v. 9, p. 2579-2605, 2008. Disponível em: http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf. Acesso em: 12 nov. 2019.

FREUND, YOAV E SCHAPIRE, ROBERT., 1997. **A Decision-Theoretic Generalization of On-Line Learning and na Application to Boosting. Journal of Computer and System Sciences.** Disponível em: http://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf. Acesso em: 05 dez. 2019.