UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO ONLINE
ESPECIALIZAÇÃO EM BIG DATA, DATA SCIENCE E DATA ANALYTICS

Fabiano Alberto Boiani

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE JURISPRUDÊNCIAS

Porto Alegre
2019

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO ONLINE
ESPECIALIZAÇÃO EM BIG DATA, DATA SCIENCE E DATA ANALYTICS

Fabiano Alberto Boiani

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE JURISPRUDÊNCIAS

Artigo apresentado como requisito parcial para obtenção do título de Especialista em *Big Data, Data Science e Data Analytics*, pelo Curso de *Big Data, Data Science e Data Analytics* da Universidade do Vale do Rio dos Sinos – UNISINOS
Orientadora: Profª Drª Patrícia Sorgatto Kuyven

Porto Alegre
2019

Aplicação de Aprendizado de Máquina Para Classificação de Jurisprudências

Fabiano Alberto Boiani

fabianoboiani@gmail.com

Abstract. *Research and implementation of a web service capable of classifying jurisprudential summaries as to its result, provided or unprovided. Using the python programming language, a supervised machine learning model was developed, which was trained through the use of a predefined jurisprudence base with their respective results. For this, some machine learning algorithms were selected in order to define which one has the best performance: Naive Bayes, Random Forest and K-Nearest Neighbors. After an evaluation of the performance of these algorithms, it was chosen the model based on the Random Forest algorithm, because it has a better performance regarding assertiveness.*

Resumo. *Pesquisa e implementação de um serviço web capaz de classificar uma ementa jurisprudencial quanto ao seu resultado, provido ou desprovido. Utilizando a linguagem de programação python, foi desenvolvido um modelo de aprendizado de máquina supervisionado, o qual foi treinado através da utilização de uma base de jurisprudências pré-definida com seus respectivos resultados. Para isso, foram selecionados alguns algoritmos de aprendizado de máquina a fim de definir qual possui um melhor desempenho: Bayes Ingênuo, Floresta Aleatória e K-Vizinhos mais próximos. Após uma avaliação quanto ao desempenho desses algoritmos, foi o escolhido o modelo baseado no algoritmo de Floresta Aleatória, por ter uma melhor performance quanto à assertividade.*

1. Introdução

A evolução mundial no que diz respeito a dados já não é novidade para ninguém, mas nos últimos anos isso se tornou uma tendência para as empresas. Pode-se dizer que o conceito *data-driven* (dirigido por dados) é o que rege hoje o mundo dos negócios. A evolução da quantidade de dados fez com que surgissem tecnologias capazes de fazer o processamento dessa grande massa de dados. Para lidar com essas mudanças surgiram tecnologias as quais podem ser citadas como exemplo o aprendizado de máquina, aprendizado profundo, processamento de linguagem natural, entre outras.

A aplicação dessas tecnologias tende a crescer dentro de empresas de todos os ramos, inclusive da área jurídica. Diante disso, surgem, nesse cenário, empresas de tecnologia chamadas de *Lawtechs*, as quais foram criadas com o intuito de desenvolver ferramentas de apoio à área jurídica (COELHO, 2018). Elas utilizam técnicas que misturam tecnologia com conhecimento jurídico, o que pode ser chamado de Jurimetria. Segundo Menezes e Barros (2017, p. 49) “a jurimetria consiste numa ferramenta ou técnica do conhecimento que alia a metodologia estatística a unidades amostrais, como a litigiosidade supradita, para estudar o funcionamento da ordem jurídica”.

As empresas lawtechs vêm desenvolvendo nos últimos anos variadas formas de utilizar-se do mundo do big data para aplicação na área jurídica. Podem ser citadas como exemplo dessas aplicações ferramentas que otimizam tarefas cotidianas, ferramentas que organizam dados de uma forma amigável para a consultas e análises e ferramentas capazes de fazer uma análise preditiva, mitigando riscos, dentre outras aplicações.

Um exemplo prático da aplicação dessas técnicas pode ser conferido no artigo intitulado "A indexação automática de acórdãos por meio de processamento de linguagem natural" (CAMARA JUNIOR, 2007). O referido artigo teve como objetivo a criação de uma indexação automática para a busca de jurisprudências na base de dados do Tribunal de Justiça do Distrito Federal utilizando técnicas de processamento da linguagem natural. Dessa forma, para atingir o objetivo, o autor realizou uma definição dos melhores termos jurídicos que representassem um melhor resultado nessas buscas.

No universo jurídico uma das funções dos juristas é tentar prever o possível resultado de conflitos legais, geralmente pautando sua estratégia a partir de tal previsão. Treinamento jurídico e experiência são fatores chave para possibilitar essa função. Juristas mais bem treinados e experientes são capazes de prever com mais eficiência o deslinde de casos jurídicos. Esses profissionais, porém, custam caro e possuem limitações, como todo o ser humano (COELHO, 2018). Baseado nisso, tecnologias como as supracitadas poderiam auxiliar profissionais menos experientes, mas também servir como uma segunda opinião aos mais experientes para uma tomada de decisão, evitando assim o desperdício de tempo com atividades operacionais e pesquisa em milhares de documentos para embasamento de sua tese.

A partir do contexto apresentado, e analisando os trabalhos já desenvolvidos na área jurídica utilizando tecnologias de big data, percebe-se que a grande maioria dos trabalhos tem como foco a análise documental, extraindo insights em cima de textos jurídicos. Objetivando utilizar uma abordagem diferente, no presente trabalho será realizada uma análise nos dados do site de jurisprudências do Tribunal de Justiça do Rio Grande do Sul (TJRS) para implementar um modelo de aprendizado de máquina com o intuito de verificar, através de uma classificação de dados, se um recurso, a nível de segundo grau, será provido, desprovido ou terá outro resultado. O resultado obtido através da aplicação desse modelo pode ser utilizado por um jurista a fim de tomar as melhores decisões sobre como deve prosseguir com seu processo.

1.1. Objetivos da Pesquisa

1.1.1. Geral

Implementar um modelo de classificação utilizando aprendizado de máquina que identifique o possível resultado em uma causa jurídica, mais especificamente de um Agravo de Instrumento, auxiliando na tomada de decisão do ator jurídico.

1.1.2. Específicos

- Obter dados do site de consulta de jurisprudência do Tribunal de Justiça do Rio Grande do Sul ou através de um pedido formal ao Tribunal de Justiça do RS, para extração das informações para posterior formação da base de dados que será utilizada no modelo.

- Através de pesquisa bibliográfica, determinar técnicas que permitem efetuar a limpeza e organização dos dados com o objetivo de torná-los estruturados.
- Escolher e implementar um modelo que permita a classificação de resultados de processos jurídicos baseado numa base de dados prévia.

1.2. Definição do problema

Apesar do contexto dinâmico em que vive a sociedade, houve um grande avanço na consolidação do Direito e do Estado democrático, por meio do fortalecimento das instituições. Porém, os desafios atuais são o aumento da eficácia e da produtividade nas organizações jurídicas. "A criação e a utilização de soluções voltadas para decisões mais rápidas, decisões melhores, inovações de serviços e redução de custos administrativos estão, atualmente, no centro das conversas e do planejamento das organizações públicas e privadas de todo o planeta" (COELHO, 2018, p.10).

A busca por jurisprudências pode ser uma tarefa muito difícil e demorada para os advogados, mesmo alguns possuindo uma extensa experiência, diminuindo assim a produtividade na sua função. Em vista disso, o presente trabalho tem como finalidade apresentar uma solução que auxilie os advogados na tomada de decisão, permitindo assim decisões mais rápidas e o aumento da produtividade ao evitar buscas desnecessárias por jurisprudências.

Geralmente advogados mais experientes e que possuem maior bagagem intelectual já sabem a probabilidade de sucesso de um recurso em um determinado cenário jurídico. Porém, um advogado recém-formado, por exemplo, já não possui a experiência necessária para identificar a probabilidade de sucesso de um recurso. Diante disso, o presente modelo de aprendizado de máquina poderá auxiliar esses advogados, menos experientes, na tomada de decisão mostrando o resultado de um possível recurso denominado Agravo de Instrumento. Já o advogado mais experiente poderá utilizar o resultado obtido a partir do modelo para realizar uma comparação entre o resultado do modelo e a sua opinião, a fim de obter maior segurança quanto a melhor decisão a ser tomada.

De acordo com Coelho (2018), a extração de informação e conhecimento através de dados objetiva um grande ganho em determinadas situações, geralmente buscando aumentar a assertividade e a celeridade do processo. Por isso, é necessário utilizar essa metodologia como uma agregação de valor, fazendo ela se integrar à forma como lidamos com determinados problemas, isto é, torná-la algo complementar e não excludente.

1.3. Delimitações do trabalho

Dentro da esfera do Poder Judiciário, mais especificamente no segundo grau, existem inúmeros tipos de recursos processuais que podem ser interpostos contra a sentença proferida por um juiz de primeiro grau, com ou sem resolução de mérito. Dentre esses recursos podem ser citados alguns, para caráter de exemplificação: recursos de apelação, de agravo de instrumento, embargos de declaração, recurso especial, dentre outros.

Esse trabalho limita-se a analisar as jurisprudências de recursos de Agravo de Instrumento do Tribunal de Justiça do Rio Grande do Sul, em processos da área cível. Optou-se por esse tipo de recurso porque o mesmo pode ser aplicado em diversas situações dentro do fluxo processual. Além disso, existe um número considerável desse tipo de recurso na base de jurisprudências.

A informatização do Poder Judiciário já existe há alguns anos, porém, a estruturação dos dados é mais recente. Anteriormente os dados eram digitados manualmente em sua maioria, o que acarretou na existência de milhares de dados com erros de digitação como, por exemplo, nomes de cidades, assuntos dos processos, nome de juízes, dentre outros. Em vista disso, a organização desses dados hoje se tornou muito complexa. Diante dessa complexidade, para a presente pesquisa foram utilizados apenas dados do período entre 2014 a 2019, a fim de proporcionar maior garantia e confiança de que os dados tenham uma melhor estruturação e organização.

1.4. Estrutura do Texto

O presente artigo está dividido em 5 seções. A seção atual objetiva mostrar uma introdução sobre o tema proposto, as delimitações do trabalho, a definição do problema abordado e os objetivos que o projeto busca alcançar.

A segunda seção contém alguns conceitos essenciais para o entendimento do artigo a respeito do Judiciário Estadual, como o conceito sobre o fluxo processual da justiça brasileira, o que podemos entender sobre um recurso do tipo Agravo de instrumento e o significado de jurisprudência. Ainda nessa seção é abordado o conceito sobre aprendizado de máquina e modelos de aprendizado supervisionado como: k-vizinhos mais próximos, árvore de decisão e florestas aleatórias. Posteriormente também são abordadas métricas utilizadas para mensurar a qualidade de um modelo de aprendizado de máquina.

A seção 3 aborda a metodologia utilizada na aplicação do projeto, como: a obtenção dos dados utilizados na pesquisa, manipulações necessárias dos dados para uma melhor estruturação, plataformas computacionais utilizadas para a aplicação prática dos modelos.

A seção 4 exibe os resultados da pesquisa, fazendo um comparativo entre os métodos escolhidos para verificar qual possui um melhor resultado, através da utilização de tabelas as quais informam as métricas de avaliação de cada modelo.

A conclusão se encontra na seção 5, a qual também aborda possíveis aplicações de trabalhos futuros relacionados a esse artigo.

2. Referencial Teórico

2.1. Fluxo Processual

A justiça brasileira é organizada e guiada pelo Código de Processo Civil (CPC) e pelo Código de Processo Penal (CPP). O fluxo de tramitação processual jurídica segue uma linha padrão de funcionamento, independente da competência processual ou classe processual (fiscal, cível, criminal, etc.). Porém, é importante ressaltar que existem movimentações processuais particulares que possuam diferenças intrínsecas em sua

tramitação, mas que de forma geral seguem o padrão definido no CPC (CAMARA JUNIOR, 2007).

No que diz respeito à decisão processual, a justiça brasileira é dividida com relação as suas instâncias ou graus. A primeira instância, a qual é atendida por Juízes de Direito em foros e varas especializadas, é a porta de entrada do Poder Judiciário brasileiro. O resultado do processo é gerado a partir de uma decisão em caráter monocrático, na qual somente um Juiz de direito redigi a chamada sentença do processo. Quando a decisão do juiz não for favorável à pessoa que entrou com a ação, ela poderá entrar com um recurso que será analisado pela segunda instância (CONSELHO NACIONAL DE JUSTIÇA, 2012).

A segunda instância é constituída pelos Tribunais de Justiça presentes em cada Estado da federação, os quais possuem suas turmas ou colegiados de Desembargadores que geralmente julgam recursos advindos do primeiro grau, e em caráter de votação decidem, unanimemente ou não, se aquele recurso deve ser provido ou desprovido. Essa decisão em segunda instância, foco deste trabalho, gera como resultado o chamado acórdão, o qual indica o acordo entre aqueles que chegaram à referida decisão. Porém, em alguns casos, a decisão em segundo grau pode ser monocrática, modificando, com isso, o fluxo normal de um processo nessa instância (CNJ, 2012).

E por último, a terceira instância, que é a mais alta instância do Poder Judiciário brasileiro, e acumula tanto competências típicas de uma suprema corte, ou seja, um tribunal de última instância, como de um tribunal constitucional, que julga questões de constitucionalidade independentemente de litígios. No terceiro grau o mérito de um tema é julgado por um colegiado de Ministros da Justiça, que após tomarem a decisão geram a chamada súmula (CNJ, 2012).

2.2. Agravo de Instrumento

De acordo com o CPC (2015) durante o processo civil são cabíveis os seguintes recursos: apelação, agravo de instrumento, agravo interno, embargos de declaração, recurso ordinário, recurso especial, recurso extraordinário, agravo em recurso especial ou extraordinário e embargos de divergência. Cada um desses recursos possui suas especificidades, sendo cabível em determinado momento do processo. Segundo o art. 996 do CPC “o recurso pode ser interposto pela parte vencida, pelo terceiro prejudicado e pelo Ministério Público, como parte ou como fiscal da ordem jurídica”.

Agravo de Instrumento, segundo o art. 994 do CPC (2015), é um recurso dirigido diretamente ao tribunal competente e cabível contra as decisões interlocutórias que versarem sobre: tutelas provisórias; mérito do processo; rejeição da alegação de convenção de arbitragem; incidente de desconsideração da personalidade jurídica; rejeição do pedido de gratuidade da justiça ou acolhimento do pedido de sua revogação; exibição ou posse de documento ou coisa; exclusão de litisconsorte; rejeição do pedido de limitação do litisconsórcio; admissão ou inadmissão de intervenção de terceiros; concessão, modificação ou revogação do efeito suspensivo aos embargos à execução; redistribuição do ônus da prova nos termos do art. 373,§ 1º do CPC/2015; além de demais casos previstos em lei.

Também caberá agravo de instrumento contra decisões interlocutórias proferidas na fase de liquidação de sentença ou de cumprimento de sentença, no processo de execução e no processo de inventário. O agravo será processado fora dos autos da causa

onde se deu a decisão impugnada, razão pela qual a petição deve ser instruída com todas as peças necessárias ao deslinde da controvérsia, formando razões e contrarrazões dos litigantes para o respectivo julgamento (CPC, 2015).

Resumindo, o agravo de instrumento é um dos variados tipos de recursos que podem ser utilizados para pedir um resultado diferente a uma decisão. Geralmente este tipo de recurso visa pedir a reforma de decisões interlocutórias, as quais são decisões que acontecem ao longo do processo não determinando seu fim, ou seja, decidem sobre fatos pontuais do processo. Segundo Montenegro Filho (2016), o agravo de instrumento visa reformar decisões proferidas por juízes de primeiro grau durante a fase de conhecimento do processo, que de alguma forma causam prejuízo a uma das partes.

Um exemplo onde esse tipo de recurso é muito utilizado é quando uma das partes do processo requer o benefício da gratuidade do judiciário, geralmente por ser de classe mais baixa. Diante de uma negativa do juiz de primeiro grau ao seu pedido é interposto recurso ao Tribunal de justiça pedindo a reforma dessa decisão.

2.3. Jurisprudência

Segundo o Supremo Tribunal Federal (2019), jurisprudência pode ser definida como um "conjunto de decisões reiteradas de juízes e tribunais sobre determinado tema" ou como uma "orientação uniforme dos tribunais na decisão de casos semelhantes". Ou seja, na prática a jurisprudência é utilizada para fortalecer a tese de um ator jurídico sobre determinado tema, fazendo então referência a uma jurisprudência já consolidada.

A jurisprudência também pode ser utilizada por juízes como referência em decisões as quais tomará em um julgamento, mas ela não determina a decisão, não somente baseado nas jurisprudências, mas também nos fatos apresentados no processo (NETTO, 2011). De acordo com o art. 926 do CPC (2015) "os tribunais devem uniformizar sua jurisprudência e mantê-la estável, íntegra e coerente".

Olhando pela ótica do ator jurídico (um advogado), a pesquisa por jurisprudências sobre um tema de seu interesse visa compreender o comportamento das instituições com as quais ele interage, nesse caso, os tribunais. Com isso ele consegue determinar como raciocina esses órgãos, a fim de estruturar melhor seus argumentos em uma causa jurídica.

2.4. Ementa

De acordo com o primeiro parágrafo do artigo 943, do CPC (2015), *"todo acórdão conterá ementa"*. A ementa basicamente consiste em uma breve síntese do conteúdo do acórdão e, por isso, deve ser feita de forma clara e objetiva. Através dela sabe-se de imediato a matéria relacionada na decisão do Tribunal, o assunto do processo, entre outras informações. Objetivamente trata-se do resumo, do sumário do acórdão.

Quando se trata de busca jurisprudencial, a ementa tem como função primordial para o ator jurídico a localização das melhores informações que podem ser úteis a ele. Isso faz com que a escrita da mesma pelos órgãos julgadores tenha que possuir uma estrutura organizada e concisa. Abaixo é apresentado um exemplo da composição de uma ementa:

"AGRAVO DE INSTRUMENTO. DIREITO PRIVADO NÃO ESPECIFICADO. IMPUGNAÇÃO À FASE DE CUMPRIMENTO DE SENTENÇA. SUSPENSÃO DO

PROCESSO. RESP 1.525.174/RS. PRELIMINAR CONTRARRECURSAL DE AUSÊNCIA DE CABIMENTO. O cabimento é um dos requisitos intrínsecos de admissibilidade recursal. Nos termos do parágrafo único do art. 1.015 do CPC/2015, também caberá agravo de instrumento contra decisões interlocutórias proferidas na fase de cumprimento de sentença. Preliminar contrarrecursal rejeitada. Determinação de suspensão do cumprimento da sentença em vista do determinado pelo STJ em relação ao tema nº 954 (RESP Nº 1.525.174-RS). Decisão que só atinge os processos ainda não julgados, não podendo alcançar os processos já decididos, frente à imutabilidade da coisa julgada. Precedentes jurisprudenciais. AGRAVO DE INSTRUMENTO PROVIDO. " [1]

2.5. Mineração de Textos

A mineração de textos, também conhecida como mineração de dados textuais é uma área multidisciplinar presente no contexto de mineração de dados que inclui conhecimentos de áreas como Informática, Estatística, Linguística e Ciência Cognitiva. Mineração de textos consiste em um processo de extrair conhecimento útil de dados a partir de textos em linguagem natural, normalmente, para objetivos específicos tais como análise textual, classificação de textos, extração de conhecimento, agrupamento textual, dentre outras (ARANHA; PASSOS, 2006).

Inspirado pelo *data mining* ou mineração de dados, que procura descobrir padrões úteis em banco de dados estruturados, a mineração de textos pretende extrair conhecimentos úteis de dados não estruturados ou semiestruturados (ARANHA; PASSOS, 2006). Ainda, os autores relatam que a tecnologia de mineração de textos vem das técnicas de recuperação de informações, aprendizado de máquina e da descoberta tradicional de informações estruturadas, através do uso de bancos de dados e de procedimentos estatísticos.

2.6. Estrutura dos dados em Mineração de Textos

Tradicionalmente algoritmos de classificação não podem processar diretamente documentos de textos no seu formato original. Por isso, durante a fase de estruturação e pré-processamento dos dados textuais, eles são organizados em um formato que esses algoritmos serão capazes de utilizá-los. Geralmente esse formato é conhecido como vetores de características (*feature vectors*). Um documento é representado como um vetor no espaço de características, isto é, uma sequência de características e seus pesos (FELDMAN; SANGER, 2006, p. 68).

A abordagem mais comum para isso é a utilização da chamada sacola de palavras (*bag of words*). Basicamente essa técnica consiste em utilizar todas as palavras do documento como suas características (*features*). A dimensionalidade do espaço de características se dará por todas as diferentes palavras presentes em todos os documentos (FELDMAN; SANGER, 2006, p. 68). O espaço de características consiste de uma matriz em um formato documentos versus termos. Abaixo segue uma exemplificação do espaço de características para um melhor entendimento, utilizando um exemplo de análise de filmes:

Documento 1 (Doc1) = "O filme muito ruim, fraco, óbvio do início ao fim"

[1] Ementa processual retirada do site de jurisprudências do Tribunal de Justiça do Rio Grande do Sul

Documento 2 (Doc2) = "Bom filme, talvez o fim do filme poderia ter sido melhor"

Tabela 1 - Matriz do espaço de características

	o	filme	muito	ruim	fraco	óbvio	do	inicio	ao	bom	talvez	poderia	ter	sido	fim	melhor
Doc1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
Doc2	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Fonte: Elaborado pelo autor

Existem três principais métodos para atribuir pesos às palavras. O mais simples é o chamado método binário, que consiste em atribuir 0 (zero) para a palavra caso essa não esteja presente no documento ou 1(um) para caso esteja presente. Esse método já foi exposto no exemplo da Tabela 1.

O segundo método utilizado para atribuir pesos aos termos dos documentos, é conhecido como contador de palavras (*word count*). É muito similar ao método binário, mas ao invés de se atribuir 0 ou 1, atribui-se um valor condicionado a quantas vezes aquela palavra aparece no documento. Utilizando o mesmo exemplo anterior, segue uma tabela que exemplifica isso:

Tabela 2 - Contador de palavras

	o	filme	muito	ruim	fraco	óbvio	do	inicio	ao	bom	talvez	poderia	ter	sido	fim	melhor
Doc1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
Doc2	1	2	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Fonte: elaborado pelo autor

Diferentemente do método binário, a palavra filme para o documento 2 teve um valor maior, pois a mesma aparece duas vezes no documento 2. Lembrando que os dois métodos citados acima utilizam uma abordagem onde o peso da palavra está dentro do contexto daquele documento e não leva em conta o restante dos documentos.

O terceiro método traz uma abordagem um pouco diferente, pois essa utiliza cálculos para penalizar a palavra caso essa apareça em muitos documentos. O método é chamado de TF-IDF, onde TF significa *term-frequency*, frequência do termo e IDF *inverse document frequency*, inverso da frequência nos documentos (FELDMAN; SANGER, 2006, p. 68). O cálculo do peso de cada palavra para esse método segue a seguinte fórmula:

$$tf - idf(w) = tf(w) \times idf(w)$$

Na fórmula anterior, *tf(w)* é a quantidade de ocorrências da palavra em um documento divido pela quantidade de palavras do documento. Já *idf(w)* é o fator que irá definir a importância da palavra com relação a toda a massa de documentos da base de dados:

$$idf(w) = \log\left(\frac{N}{df(w)}\right)$$

Onde *N* é a quantidade de documentos na massa de dados e *df(w)* é a quantidade de documentos que contenham a palavra *w*.

2.7. Aprendizado de máquina

O aprendizado de máquina (ou *machine learning*) é uma área dentro da inteligência artificial que tem como foco principal criar métodos ou técnicas capazes de ensinar computadores a desempenharem determinadas funções, geralmente aprendendo com suas próprias experiências. Isso soa muito próximo ao que hoje os animais, incluindo os humanos, já fazem (FACELLI *et al.*, 2011). Nesse contexto, experiências referem-se a dados já existentes, em que os métodos de aprendizado de máquina os utilizam para extrair informações e posteriormente criar um novo conhecimento a fim de determinar comportamentos quando submetidos a novos dados.

Com o intuito de exemplificar, pode-se dizer que se você ensinar a um algoritmo computacional de aprendizado de máquina que um cachorro pode ser traduzido a um animal que late, tem rabo e quatro patas, o sistema, em um segundo momento, quando questionado o que seria um animal com essas características, seja capaz de responder que isso é um cachorro. Podem ser citados os seguintes exemplos de aplicação de aprendizado de máquina: na área da saúde, a aplicação para determinar se há evidências de câncer ou não em imagens tomográficas; na área financeira, se um cliente é um potencial a cometer fraude; na área jurídica, para determinar se um processo terá ou não ganho de causa; entre outros.

Dentro do aprendizado de máquina existem três abordagens diferentes: aprendizado não supervisionado, que tem como objetivo agrupar os dados baseado em determinadas características; aprendizado por reforço, o qual se utiliza de recompensas a fim de se ajustar dependendo dos resultados; e o aprendizado supervisionado, foco deste trabalho, o qual utiliza dados de entrada e seus resultados para treinar os algoritmos a fim de ter a capacidade de predizer ou classificar um novo resultado quando submetido a novos dados (NORVIG; RUSSELL, 2013).

De uma forma abstrata o aprendizado de máquina se resume a ter um conjunto de exemplos os quais contém as características de cada exemplo (também denominado, dado de entrada ou registro) e sua determinada classe (caso seja um aprendizado supervisionado). A classe, como dito anteriormente, representa a classificação que essas características determinam, e os atributos que são as características presente em cada exemplo do conjunto de exemplos (NORVIG; RUSSELL, 2013). A partir disso é escolhida uma técnica apropriada a sua necessidade e aplicada ao seu conjunto de elementos para um prévio treinamento e posterior aplicação a futuros dados de entrada.

O princípio mais básico que há no aprendizado de máquina quando se trata de classificação é o que diz que dados com características parecidas tendem a ter uma mesma classificação, é assim que os métodos baseados em distância funcionam e um dos mais utilizados é o método k-vizinho mais próximo. Já outros métodos utilizam-se de outras técnicas para realizar essa classificação, como os métodos baseados em procura, que consiste em dividir o problema principal em problemas menores trabalhando de forma recursiva a fim de atingir sua classificação; um exemplo desse tipo de método são as árvores de decisão (FACELLI *et al.*, 2011).

2.8. Modelos de Aprendizado Supervisionado

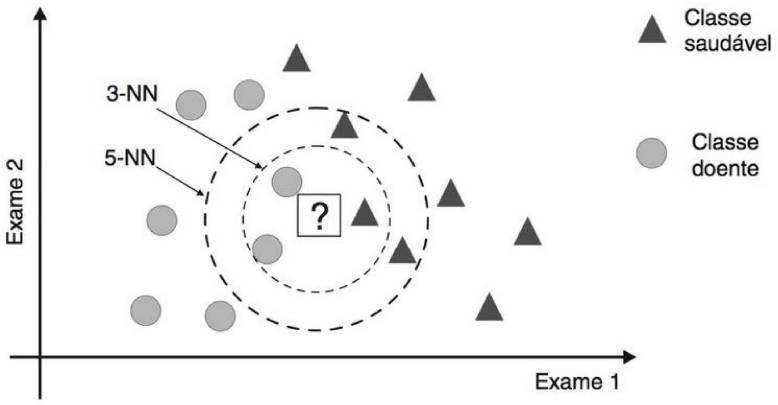
2.8.1. KNN (k-vizinhos mais próximos)

É um dos métodos mais utilizados quando se fala em métodos baseados em distância. Considerado um dos métodos de classificação mais fáceis de entender e o mais simples dentro do aprendizado de máquina, o método basicamente classifica um novo objeto com base no conjunto de exemplos de treinamento que são mais próximos a ele. É considerado um algoritmo preguiçoso, pois ele não aprende um modelo compacto com referência aos dados, mas simplesmente memoriza os dados de treinamento (FACELLI *et al.*, 2011).

De acordo com Mitchel (1997), apesar de ser considerado um algoritmo simples, a sua grande desvantagem é seu alto custo computacional, custo esse que ocorre no momento da classificação de novos exemplos e não no momento de aprendizagem, o que não acontece em outros métodos de aprendizagem.

Como o próprio nome sugere, estamos nos referindo a *k* vizinhos mais próximos, onde *k* é um parâmetro do método. Significa dizer que é escolhido pelo usuário do método quantos vizinhos ele quer considerar para utilizar na classificação. A variável *k* pode ser muito importante, pois o seu valor pode impactar na hora de classificar um novo dado submetido ao modelo. Conforme a Figura 1, a qual exemplifica uma classificação de um paciente em um hospital entre doente e saudável, caso seja selecionado o valor 3 para a variável *k*, o modelo de classificação resultaria a classe doente, mas caso seja selecionado 5 para o valor da variável *k*, a classificação seria saudável. Geralmente utilizam-se valores impares para a variável *k*, quando utilizado sobre problemas de classificação, evitando assim um empate.

Figura 1 – Algoritmo KNN



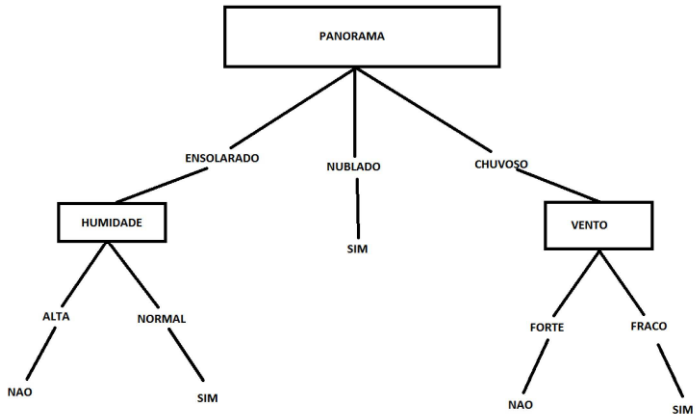
Fonte: Figura retirada de Facelli et al. (2011)

Percebe-se que a escolha desse parâmetro *k* nesse método de classificação não seria um trabalho trivial, pois ele pode determinar diferentes classificações, quando alterado seu valor. Existem técnicas que ajudam a melhorar essa questão, como, por exemplo, estimar *k* utilizando validação cruzada ou determinar pesos para a contribuição de cada vizinho. Porém, como não é foco deste trabalho, o detalhamento de tais técnicas não será explicado.

2.8.2. Árvores de decisão (Decision Tree)

Segundo Facelli *et al*. (2011, p. 83) "uma árvore de decisão usa a estratégia dividir para conquistar para resolver um problema de decisão. Um problema complexo é dividido em problemas mais simples, aos quais recursivamente é aplicada a mesma estratégia". Resume-se dizer que uma árvore de decisão seria um conjunto de nós "se-então/senão" em um formato similar a uma árvore, contemplado por nós de *divisão*, que tem como sucessores dois ou mais nós, ou nós *folhas,* que tem como saída a classificação obtida pela árvore de decisão. Pode-se visualizar melhor na Figura 2 um exemplo de árvore de decisão, na qual são classificadas as condições para jogar tênis.

Figura 2 Exemplo de uma árvore de decisão



Fonte: Figura traduzida pelo autor de Mitchel (1997)

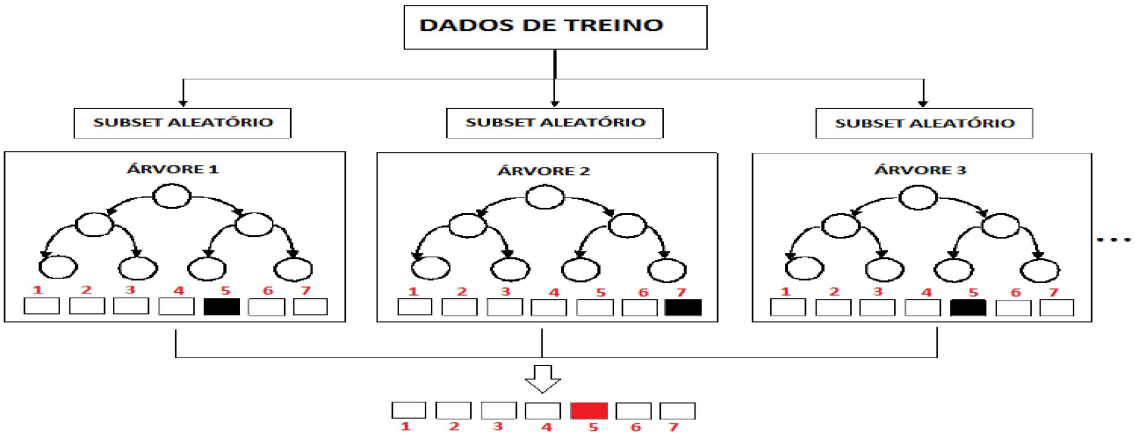
2.8.3. Floresta Aleatória (Random Forest)

Floresta Aleatória, como o nome já sugere, é uma combinação de árvores de decisão as quais são submetidas a diferentes amostragens dos dados de treinamento. As melhores variáveis de entrada para cada árvore utilizar são escolhidas aleatoriamente, com isso, realizando suas previsões independentes. Ao final, o método determina qual foi a classe mais votada entre as árvores da floresta aleatória (BRIEMAN, 2001). Essa abordagem é muito próxima ao que é possível chamar de *bagging*, o qual sugere a agregação de vários classificadores, aumentando assim seu poder de predição e confiança (BRIEMAN, 1996).

Fazendo uma analogia com o mundo real, imagine que um sujeito gostaria de realizar uma viagem, e para isso ele escolhe alguns amigos para questionar possíveis destinos. O primeiro amigo realiza algumas perguntas, como, por exemplo, se ele gosta de mar, se gosta de acampar e etc. Baseado nessas respostas o primeiro amigo então sugere um destino. E essa pessoa então realiza o mesmo procedimento com outros amigos. Passado um número suficiente de amigos, o qual é determinado pelo sujeito, o mesmo contabiliza as opiniões dos amigos e verifica qual o destino mais sugerido, com isso fazendo a sua escolha.

Abaixo, na Figura 3, é possível ver um exemplo de como é formada uma floresta aleatória.

Figura 3 - Floresta Aleatória



Fonte: Figura traduzida e adaptada de Isayed (2015)

2.8.4. Bayes ingênuo (Naive Bayes)

O algoritmo Bayes Ingênuo é considerado um classificador probabilístico e muito conhecido no meio do aprendizado de máquina, ainda mais quando se faz relação com classificação de texto. Ele é baseado no Teorema de Bayes, o qual foi criado pelo matemático inglês Thomas Bayes, com o objetivo de provar a existência de Deus.

Na classificação de textos, esse algoritmo utiliza como recurso para classificar os textos a frequência das palavras no texto. Ele utiliza a fórmula do Teorema de Bayes para calcular cada probabilidade de cada palavra com relação à classificação resultante. A partir disso se entende o porquê do algoritmo se chamar Ingênuo (*Naive*), ele não considera as correlações entre as palavras, ou seja, elas são tratadas separadamente (FACELLI *et al.*, 2011). Utilizando um exemplo para um melhor entendimento, dado uma fruta classificada como Laranja, o algoritmo não fará a relação se ela é cor laranja e redonda, essas duas características serão calculadas independentemente uma da outra.

Abaixo é apresentada a fórmula definida pelo Teorema de Bayes para calcular essas probabilidades:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Onde P(A|B) é a probabilidade de A acontecer dado B, P(A) é a probabilidade da classe A ocorrer, P(B|A) é a probabilidade de B ocorrer dado A e P(B) é a probabilidade de B ocorrer no conjunto.

2.9. Métricas de avaliação de modelos de aprendizado de máquina

Quando se aplica aprendizado de máquina sobre alguma situação do mundo real, o conhecimento que se tem sobre essa situação é totalmente baseado nos dados utilizados nessa aplicação, os quais posteriormente resultam na geração de um modelo preditivo/classificatório.

Apesar de haver variadas possibilidades de modelos ou técnicas de aprendizado de máquina, não existe uma regra de aplicação para cada situação. O que existe é uma

ideia de que um determinado modelo possa se adequar melhor à determinada situação, baseado nas características dos dados. Além disso, diversos modelos podem ser considerados candidatos à solução de um mesmo problema. Ainda que um algoritmo seja escolhido, existe a possibilidade de realizar ajustes em seus parâmetros, o que leva à obtenção de vários modelos para os mesmos dados. (FACELLI *et al.*, 2011).

Fica evidente que em aprendizado de máquina é necessária a utilização de experimentos para verificar qual modelagem se aplica melhor a uma situação em questão, mesmo que se utilize um ou mais modelos. Para isso, utilizam-se métricas que medem o desempenho desses modelos. A seguir serão exibidas as métricas que são utilizadas para atestar a qualidade de um modelo de aprendizado de máquina para classificação.

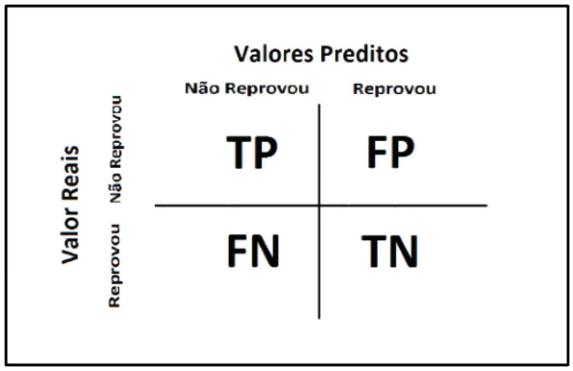
Algumas das principais métricas utilizadas para medir o desempenho de um modelo de aprendizado de máquina são extraídas a partir do que é denominada de matriz de confusão.

A matriz de confusão é formada por:

- **Verdadeiro Positivo** (do inglês *True Positive* - TP): esse valor ocorre quando o resultado da predição do modelo equivale ao valor positivo real esperado. Por exemplo, quando um aluno não reprovou na escola e o modelo previu que ele não reprovou.
- **Falso Positivo** (do inglês *False Positive* - FP): esse valor ocorre quando o resultado da predição do modelo previu um valor positivo quando o mesmo era negativo. O modelo previu que o aluno não reprovou, mas ele havia reprovado.
- **Verdadeiro Negativo** (do inglês *True Negative* - TN): esse valor ocorre quando o resultado da predição do modelo previu um valor negativo quando esse valor negativo era o esperado. O modelo previu que o aluno reprovou e ele realmente reprovou.
- **Falso Negativo** (do inglês *False Negative* - FN): esse valor ocorre quando o resultado da predição do modelo indica um valor negativo quando o valor esperado era um positivo. O modelo previu que o aluno reprovou, quando o aluno na verdade não reprovou.

Segue abaixo uma demonstração de como fica uma matriz de confusão:

Figura 4 - Matrix de confusão



Fonte: Elaborado pelo autor

Com esses valores em mãos, é possível extrair as seguintes métricas:

- **Acurácia**: essa métrica informa de forma geral quantos acertos o modelo teve dentro de todas as possibilidades. Se existem 10 alunos na massa de dados e o modelo acertou 5 de 10 das previsões (tanto negativas quanto positivas), tem-se uma acurácia de 50%. Resume-se na fração entre as previsões acertadas por todas as previsões efetuadas.

$$acc = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Revocação** (em inglês *Recall*): essa métrica mede o desempenho do modelo em prever valores positivos. Utilizando o mesmo exemplo acima dos alunos, quão bom é o modelo em prever alunos que não irão reprovar.

$$recall = \frac{TP}{TP + FN}$$

- **Precisão** (em inglês *Precision*): essa métrica mede quantos resultados positivos realmente foram previstos corretamente pelo modelo.

$$precision = \frac{TP}{TP + FP}$$

- **F_1-score**: essa métrica mescla os valores do revocação com os de precisão para nos dar uma visão melhor do desempenho geral do modelo.

$$fscore = \frac{2 * precision * recall}{precision + recall}$$

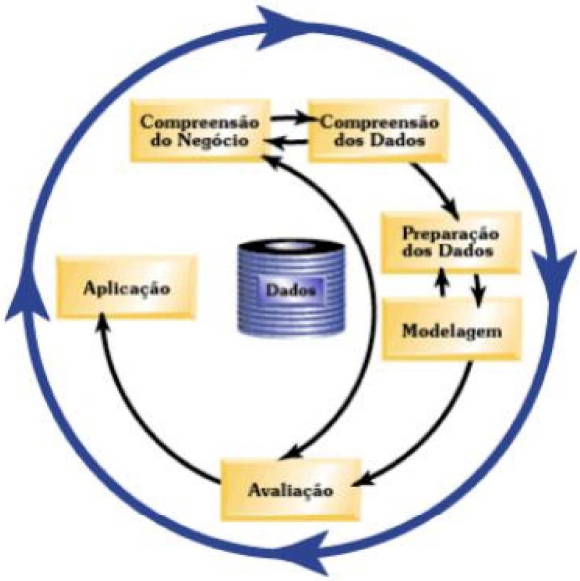
2.10. CRISP-DM

Quando se pretende trabalhar com mineração de textos não se pode somente focar em busca de padrões e descobrimento de conhecimento nos dados, mas também se faz necessário utilizar-se de técnicas de pré-processamento e limpeza desses dados. Para isso, é necessária a utilização de padrões de etapas para serem seguidas a fim de alcançar um objetivo. Neste trabalho foi utilizado o processo chamado CRISP-DM, do

inglês **CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining) (AZEVEDO; SANTOS, 2008).

O CRISP-DM é formado por seis etapas e defini os passos que devem ser seguidos no processo de mineração de textos. A figura a seguir demonstra o funcionamento das etapas desse processo:

Figura 5 - CRISP-DM



Fonte: Imagem adaptada de Azevedo e Santos (2008)

- **Compreensão do negócio:** essa é a fase inicial do CRISP-DM e o seu foco principal é o entendido do objetivo do negócio, ou seja, o problema que se quer resolver. Analisa-se principalmente o objetivo que se quer alcançar e seus critérios de aceitação de sucesso.
- **Compreensão dos dados:** nessa fase o objetivo é inicialmente efetuar a coleta dos dados a serem utilizados no projeto e posteriormente realizar uma análise inicial com o intuito de criar os primeiros insights sobre esses dados utilizando estatística. Essa é uma das fases onde um cientista de dados utiliza a maior parte de seu tempo de projeto. É nessa fase que ele verifica a qualidade dos dados, identifica familiaridade entre os dados e cria *subsets* dos dados do seu interesse a fim de responder hipóteses posteriores.
- **Preparação dos dados:** é a fase na qual os dados serão pré-processados a fim de deixá-los em um formato no qual os algoritmos de aprendizado de máquina propostos poderão utilizá-los. Nessa fase aplicam-se diferentes técnicas como limpeza dos dados, formatação dos dados, seleção dos melhores atributos, integração de dados, entre outras. Tomam-se decisões relativas, por exemplo, como tratar dados faltantes, dados ruidosos, criação de novos atributos utilizados uma conjunção de outros, discretização de dados numéricos, entre outras. Na mineração de textos são executadas técnicas especificas de pré-processamento como remoção de *stopwords*

(artigos, preposições, conjunções, assim como outras palavras auxiliares que não agregam valor ao documento), remoção de caracteres especiais, numeração, acentuação, pontuação, etc.

- **Modelagem:** nessa fase utiliza-se então o resultado (a massa de dados já tratada) que se obteve da fase de preparação dos dados para criar modelos de aprendizado de máquina com o objetivo de encontrar padrões, realizando os ajustes necessários de parametrização a fim de gerar o melhor modelo.
- **Avaliação:** essa fase é responsável por testar a qualidade do modelo (ou dos modelos). Geralmente utiliza-se de métricas para realizar esse comparativo com o objetivo esperado.
- **Aplicação:** a fase de aplicação diz respeito a disponibilização do modelo para um usuário final.

2.11. Trabalhos Relacionados

2.11.1. Análise jurisprudencial com técnica de aprendizado de máquina

O trabalho apresentado por Barros (2018) teve como objetivo a descoberta de conhecimento quanto à tendência de julgamento em processos do Tribunal Regional do Trabalho da 3ª Região com relação à parte favorecida, seja empregado ou empregador.

Para formação da base de dados foram utilizados acórdãos em forma textual, diferentemente do presente trabalho. Assim, o autor extraiu desses acórdãos as informações necessárias para alcançar o seu objetivo. Como, por exemplo, quem entrou com o processo (empresa ou empregado) e o resultado daquele processo.

Como resultado do estudo, o autor conseguiu concluir que, de um modo geral, o Tribunal do Trabalho não julga com tendência sempre para o empregado, pois os números dos resultados mostram que existe um equilibro entre resultados positivos para empregado e empregador. Porém, quando foram realizadas análises utilizando-se especificamente algumas Turmas Recursais, essas mostraram tendência de resultado positivo para empregados mais do que para empresas. Assim como o contrário, algumas turmas recursais deram mais ganho de causa para empresas do que para empregados.

2.11.2. Aprendizado de classificadores das ementas da Jurisprudência do Tribunal Regional do Trabalho da 2ª Região – SP

Fearuche e Almeida (2011) apresentaram um trabalho que teve como finalidade classificar ementas jurisprudências do Tribunal de Justiça do Trabalho quanto à sua categoria (Execução, Previdência Social, Relação de Emprego, entre outras). Assim, há uma relação muito próxima com o presente trabalho, pois ambos utilizam ementas como fonte de dados. A diferença é quanto ao objetivo, pois o presente trabalho tem como objetivo a classificação do resultado de uma jurisprudência, e não quanto à sua categoria. O que também difere deste trabalho são as ferramentas utilizadas.

No estudo de Fearuche e Almeida (2011), foi utilizada como ferramenta para a criação dos modelos o WEKA. Nessa ferramenta, basicamente, os algoritmos já estão implementados, o que somente é necessário é a criação da base de dados a qual o

algoritmo utilizará para treinar. Foram criados 3(três) modelos de aprendizado de máquina, o SVM (Máquina de vetores de suporte – *Support Vector Machines*), Árvore de decisão (sob o algoritmo J4.8) e *Naive Bayes*.

Os resultados obtidos nesse trabalho foram satisfatórios, porém não se conseguiu distinguir qual dos algoritmos teve um melhor desempenho, pois a diferença é muito pequena na fase de treinamento. Quanto à taxa de erro dos modelos quando aplicados à base de teste, constatou-se que o modelo gerado a partir do algoritmo de *Naive Bayes* teve o pior resultado. Para algumas categorias, inclusive, o algoritmo não conseguiu acertar nenhuma vez.

2.11.3. Indexação automática de acórdãos por meio de processamento da linguagem natural.

O trabalho apresentado por Câmara Junior (2007) objetivou o desenvolvimento de um método de indexação automática de documentos jurídicos, neste caso acórdãos, a fim de otimizar a busca por informação, fazendo com que se possa retornar em uma busca somente as informações que realmente sejam úteis para o usuário.

Para a realização desse trabalho o autor utilizou inteligência artificial junto ao processamento de linguagem natural, assim tornando possível a criação de índices melhores para esses documentos. No fim, as suas métricas de validação mostraram que o método de indexação automática criado conseguiu alcançar níveis de performance melhores que uma indexação manual. A vantagem disso é a possibilidade de redução de mão de obra na atividade de indexação, possibilitando o remanejamento de pessoas para outras funções importantes que demandem mão de obra.

Apesar de o trabalho de Câmara Junior (2007) não seguir a mesma abordagem do presente trabalho, ele mostra uma visão diferente de utilização de mineração de texto, fazendo uma abordagem mais morfológica do texto. Isso mostra a variedade de abordagens que se pode utilizar quando se trata de mineração de texto utilizando processamento de linguagem natural.

3. Metodologia

3.1. Delineamento da Pesquisa

Essa pesquisa tem um caráter quantitativo, o qual, segundo Richardson (1999), é caracterizado pelo emprego de quantificação na coleta de informações e no tratamento dessas por meio de técnicas estatísticas, que vão desde as mais simples, como percentual, média, desvio padrão, a mais complexas, como coeficiente de correlação, análise de regressão, entre outras. Com relação ao nível de pesquisa, ela se dará de forma explicativa, que segundo Gil (2002, p. 42) “têm como preocupação central identificar os fatores que determinam ou que contribuem para a ocorrência dos fenômenos”.

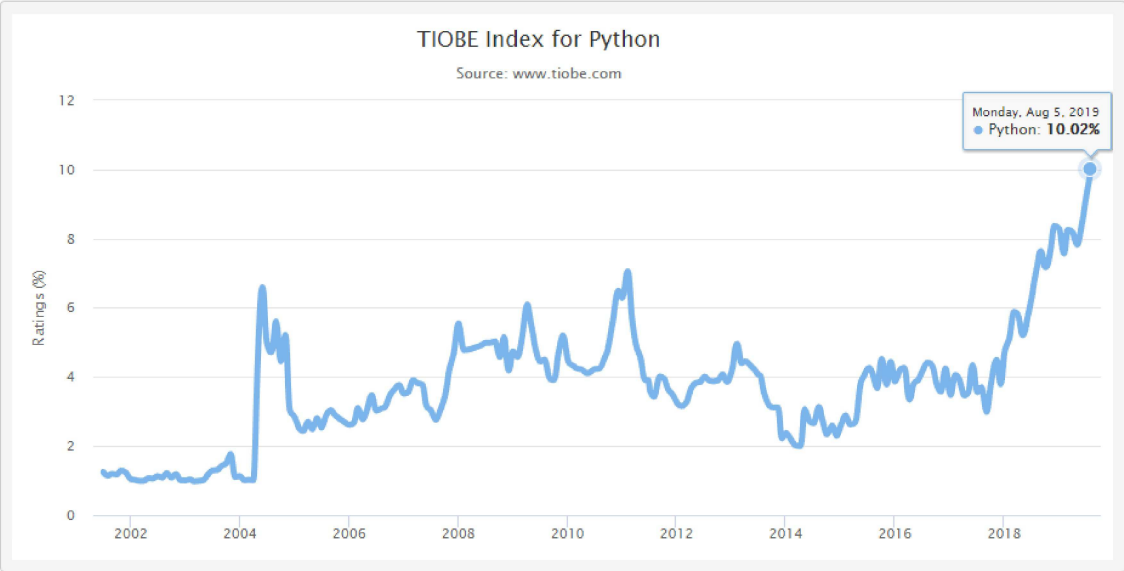
Para a obtenção dos resultados da presente pesquisa foram utilizadas as seguintes técnicas: pesquisa bibliográfica, com o objetivo de entender a área de atuação do estudo e formular um referencial teórico como embasamento; e pesquisa experimental, que visa, segundo Gil (2002, p.47), “determinar um objeto de estudo, selecionar variáveis que seriam capazes de influenciá-lo, definir formas de controle e de observação dos efeitos que a variável produz no objeto”.

Para a parte de mineração de texto foram seguidas as etapas expostas na metodologia apresentada anteriormente denominada CRISP-DM (AZEVEDO; SANTOS, 2008).

3.2. Ferramentas Utilizadas

Para a implementação do projeto foi escolhida como linguagem de programação a linguagem *Python* (PYTHON, 2019), pois ela vem de um crescente nos últimos anos quando se trata de flexibilidade, inteligência artificial e processamento de grandes volumes de dados. Abaixo é possível ver esse crescente pelo gráfico da TIOBE (TIOBE, 2019), a qual é uma empresa responsável por extrair métricas relacionadas a linguagens, dentre essas a popularidade de linguagens de programação.

Figura 6 - Tiobe Index for Python



Fonte: TIOBE (2019).

A partir da escolha da linguagem de programação, buscou-se bibliotecas dentro do universo do Python que fossem voltadas para mineração e manipulação de textos. Com isso foram selecionadas bibliotecas como a *NTLK* (NTLK, 2019), *PANDAS* (PANDAS, 2019) e *scikit-learn* (SCIKIT-LEARN, 2019).

A biblioteca *NTLK* tem como função trabalhar com a linguagem humana. Ela fornece variadas ferramentas de pré-processamento textual como tokenização textual, stemização, lematização, entre outras. A biblioteca *Pandas* tem como função a manipulação de dados, análise, obtenção de dados em diferentes tipos de fontes, etc. Já a biblioteca scikit-learn tem como função a aplicação de modelos de aprendizado de máquina, os quais já foram citados anteriormente.

Para a implementação e execução dos algoritmos foi escolhido o ambiente chamado de *Google Colaboratory* (COLABORATORY, 2019), ou somente Google Colab. Nele é possível criar e executar códigos *Python* sem a necessidade de utilizar a máquina local do indivíduo, pois a carga de trabalho é toda realizada na própria infraestrutura do ambiente o qual se utiliza para desenvolver os códigos *python*.

3.3. Entendimento do Negócio

Com o intuito de se obter um maior entendimento da área jurídica, suas particularidades, como funcionam as instituições, entre outros aspectos, foram realizados dois encontros com profissionais da área, mais precisamente advogados; sendo um encontro com cada profissional, para uma conversa informal possuindo um viés de entrevista exploratória com duração de aproximadamente uma hora.

Dentre os questionamentos, estavam perguntas como:

- Como geralmente você realiza a busca por jurisprudências?
- Você tem uma certa noção de como determinados juízes se comportam com relação a um determinado tema processual?
- Essa noção vem a partir da sua longa experiência?
- Qual a diferença de um Agravo de Instrumento para outro tipo de recurso?
- Quais os cenários que se aplicam um agravo de instrumento no fluxo processual?
- Seria plausível você utilizar palavras chaves relacionadas ao processo para buscar jurisprudências?
- Se você tivesse uma ideia de resultado para um determinado processo, isso seria útil para uma tomada de decisão?

As perguntas foram elaboradas com base na experiência profissional do autor do presente trabalho que atualmente atua como programador de sistemas no Tribunal de Justiça do Rio Grande do Sul.

A partir dessas entrevistas foi possível obter informações a respeito do procedimento que é realizado por um advogado na coleta por jurisprudências para verificação da possibilidade de ganho de causa de um determinado processo. As informações obtidas foram de suma importância para definir de que forma a pesquisa seria conduzida, isto é, utilizando uma abordagem mais voltada para a mineração de texto ou somente utilizando dados tabulares estruturados para a criação dos modelos de aprendizado de máquina.

3.4. Coleta dos dados

Tendo em vista que parte das tarefas de um cientista de dados é realizar a coleta de dados, da forma como a mesma se dará, inicialmente seria desenvolvida uma ferramenta chamada de *Web Scraper*, uma espécie de raspador de dados de sites. Porém, durante o seu desenvolvimento foi verificado que o site do TJRS, no menu jurisprudências, possui uma limitação com relação ao número de jurisprudências retornadas na busca. Mesmo que o site mostre a existência de 100 mil jurisprudências para os filtros informados, a navegação dentre as jurisprudências retornadas somente exibe as primeiras mil jurisprudências.

Diante dessa limitação, foi então realizado um pedido formal via e-mail para o departamento de informática do TJRS, com o objetivo de extrair uma amostra dos dados de jurisprudências diretamente do banco de dados. Com a aprovação do departamento de informática, foi então realizada uma pesquisa no banco de dados utilizando SQL

(*Structured Query Language*). A amostra extraída compreende o período entre 01-2014 e 06-2019.

Os dados extraídos inicialmente contêm 461189 registros contendo as seguintes informações, conforme Tabela 3:

Tabela 3 - Colunas da base de dados

Tipo de dado	Descrição
COD_EMENTA	Numeração única para cada jurisprudência
COD_ORG_JULG	Código referente ao Órgão Julgador que julgou o processo. Por exemplo: Primeira Câmara Cível.
NUMERO_PROCESSO	Número do processo gerado pelo sistema do TJRS
COD_RECURSO	Código referente ao tipo de recurso, nesse caso todos valores serão referentes a Agravo de Instrumento.
DATA_JULG	Data quando ocorreu o julgamento do processo.
COD_RELATOR	Código referente ao Desembargador Relator do processo. Responsável por relatar aos outros desembargadores sobre o que trata o processo.
ORIGEM	Origem diz respeito a cidade a qual originou o recurso. Cidade referente ao processo em primeiro grau.
COD_TIPO_DOCUMENTO	Código referente ao tipo de documento gerado na decisão do recurso. Decisão Monocrática ou Acordão.
DATA_PUBLICACAO	Data a qual foi publicada a jurisprudência no site do TJRS.
TEXTOEMENTA	Texto da ementa de cada processo
NOME_CLASSE_CNJ	É a descrição da classe CNJ do processo, no caso desse projeto "Agravo de Instrumento".
NOME_ASSUNTO_CNJ	É o assunto do qual o processo trata. Exemplo: Telefonia, Pensão alimentícia, entre outros.
NOME_JULG_1GRAU	Nome do Juiz de Direito o qual julgou o processo na primeira instancia.

Fonte: elaborado pelo autor

Verificando a tabela anterior, a resposta que o trabalho busca não está presente. Ou seja, não se tem o resultado do processo exposto diretamente nos dados. O resultado do processo está presente dentro do texto das ementas de cada jurisprudência. Então foi realizado um processo de descoberta da variável "resultado", nesse caso o resultado do recurso. Para isso foram feitas algumas análises manuais para cada tipo de resultado. Na amostra que foi retirada da base de dados do TJRS para o presente trabalho, verificou-se que existem praticamente 4 (quatro) tipos de resultados:

- **Provido**: quando a turma de magistrados que julga o processo aceita aquele pedido feito pela parte que interpôs o recurso.
- **Desprovido**: quando a turma de magistrados que julga o processo rejeita o pedido feito pela parte que interpôs o recurso.

- **Declinação de competência**: quando o Magistrado relator do recurso verifica que aquele pedido que está sendo feito pela parte não deve ser julgado por juízes de segunda instância e sim de uma instância inferior, nesse caso um juiz de primeiro grau.
- **Não conhecido**: significa dizer que um recurso está mal formulado ou falta documentos para o juiz tomar alguma decisão. Ele então não reconhece como um recurso válido. Não chega a verificar o mérito.

Para cada um dos 4 (quatro) tipos de resultados encontrados, notou-se que a forma como um determinado magistrado escreve um mesmo resultado varia. Por exemplo, para um resultado do tipo "Provido" existem, por exemplo, as seguintes formas de redigi-lo:

Tabela 4 - Formas de escrever um resultado

RECURSO PROVIDO
DADO PROVIMENTO
AGRAVO DE INSTRUMENTO PROVIDO
AGRAVO PROVIDO
AGRAVO MONOCRATICAMENTE PROVIDO
SE DÁ PROVIMENTO
DERAM PROVIMENTO
DANDO PROVIMENTO
RECURSO, DE PLANO, PROVIDO
PROVIDO. EM MONOCRÁTICA
DOU PROVIMENTO
EM DECISÃO MONOCRÁTICA, PROVIDO

Fonte: elaborado pelo autor

Com isso, para cada tipo de resultado foram realizadas análises para determinar a maioria das formas de se escrever um determinado resultado. A partir desse processo, conseguiu-se gerar a variável resposta "RESULTADO".

Utilizando a mesma técnica de descoberta da variável resposta "Resultado", foram removidas das ementas essas palavras a fim de não interferir na modelagem dos algoritmos, pois caso fossem deixadas no texto essas tornariam o resultado um falso positivo.

Como o objetivo deste trabalho é baseado somente sobre os resultados "Provido" e "Desprovido", os registros que diferem destes resultados foram descartados. Com isso a base de dados ficou com 323555 registros.

Outra filtragem realizada foi com relação às demais variáveis dos dados. O presente trabalho tem como objetivo realizar uma abordagem sobre o texto da ementa, por isso as demais variáveis da base de dados foram descartadas posteriormente, ficando somente o texto da ementa e o resultado do processo. A Figura 7 apresenta como ficou os dados após essa filtragem.

Figura 7 - Pequena amostra da tabela resultante

	TEXTOEMENTA	RESULTADO
0	execucao sentenca correcao monetaria data cal...	PROVIDO
1	negocios juridicos bancarios acao cominatoria...	DESPROVIDO
2	acao revisional contrato onus prova inscricao...	DESPROVIDO
3	negocios juridicos bancarios execucao titulo ...	DESPROVIDO
4	acao declaratoria cumulada indenizatoria com...	PROVIDO

Fonte: elaborado pelo autor

Inicialmente a ideia era realizar um procedimento de aplicação de aprendizado de máquina levando em consideração outras variáveis, por isso inicialmente a base de dados possuía mais variáveis do que o necessário para o projeto, como por exemplo, a origem do processo, julgador em primeiro grau, relator do processo, etc. Porém, durante o desenvolvimento da modelagem do algoritmo de aprendizado de máquina, percebeu-se que as variáveis utilizadas não estavam tendo influência no resultado dos processos. Como uma das funções de um cientista de dados é fazer análises de formas de resolução de um problema proposto, tomou-se a decisão então de fazer uma análise das ementas.

3.5. Pré-processamento dos dados

Para mineração de textos geralmente se faz necessário alguns tratamentos que precisam ser realizados, isso porque alguns algoritmos de aprendizado de máquina se tornam mais eficientes na sua capacidade de classificação. Para isso utilizou-se algumas técnicas para realizar essa limpeza.

3.5.1. Limpeza inicial

Para trabalhar com textos, inicialmente se faz necessário passar todos os textos para um formato de letras minúsculas. Posteriormente é realizada uma limpeza de caracteres que não são necessários na classificação textual como "/(){}\[\]\|@". Após a remoção dos símbolos/caracteres é removida toda a pontuação do texto como pontos, vírgulas, ponto e vírgula, etc.

3.5.2. Remoção de Stopwords

Para a classificação de textos geralmente é necessário somente palavras que tem algum peso no algoritmo, palavras que possam estar presentes em todos os textos acabam possuindo uma má influência no algoritmo, podendo até atrapalhar na eficácia do mesmo. Essas palavras são chamadas de *stopwords.* Abaixo alguns exemplos de *stopwords* que foram removidas dos textos:

'a', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', aquilo', 'as', 'até', 'com', 'como', 'da', 'das', 'de', 'dela', delas', 'dele', 'deles', 'depois', 'do', 'dos', 'e', 'ela', 'há', 'hão', 'isso', 'isto', 'já', 'lhe', 'lhes', 'mais', 'mas', 'me', 'mesmo', 'meu', 'meus', 'minha', 'minhas', 'muito', 'na', 'nas', 'nem', 'no', 'nos', 'nossa', 'nossas', 'o', 'os', 'ou', 'para', 'pela', 'pelas', 'pelo', 'pelos', 'por', 'qual', 'quando', 'que', entre outras.

3.5.3. Stemização (*Stemming*)

Para uma melhor análise dos algoritmos de aprendizado de máquina é realizada a normalização das palavras, isto é, palavras que possuem flexão serão consideradas para os algoritmos a mesma palavra. Isso é chamado de *stemizar* a palavra, obter seu radical, o seu tronco (do inglês, *stem*). Por exemplo, as palavras livro, livrinho, livreiro e livreco tem como radical **livr**. Quando é realizado esse processo no texto, todas as palavras estão sendo reduzidas ao seu radical.

3.5.4. Exemplo do Pré-processamento

Abaixo segue um exemplo de como fica uma ementa após passar pelo processo de pré-processamento textual utilizado no projeto.

Ementa extraída do site do TJRS:

AGRAVO DE INSTRUMENTO. EXECUÇÃO FISCAL. PEDIDO DE CONSULTA AO SISTEMA INFOJUD. POSSIBILIDADE. PRECEDENTES DESTA CORTE. Deve ser deferida a pretensão do exequente, pois tem ele o direito de utilizar os meios disponíveis na busca de bens passíveis de constrição. Não se deve obstar o direito do Ente Público de ver adimplido seu crédito tributário. AGRAVO DE INSTRUMENTO PROVIDO, EM DECISÃO MONOCRÁTICA.

Resultado da aplicação da limpeza e pré-processamento textual:

agrav instrument execuca fiscal ped consult sistem infojud possibil preced dest cort dev ser defer pretensa exequ poi direit utiliz mei disponi busc bem passi constrica nao dev obst direit ent publ ver adimpl credit tributari agrav instrument prov decisa monocra

3.6. Modelagem

Não existe nenhum tipo de mágica na criação de modelos de aprendizado de máquina. O que existe é algoritmos que são conhecidos por se saírem melhores em alguns aspectos diante de um determinado desafio. A função do cientista de dados é realizar testes com variados algoritmos para verificar qual desses terá um melhor resultado diante do projeto o qual ele esteja trabalhando. Por isso, nesse projeto foram utilizados diferentes algoritmos de aprendizado de máquina a fim de testar qual deles iria ter uma melhor performance. Dentre esses algoritmos foram utilizados os algoritmos de KNN, Floresta Aleatória e *Naive Bayes*. Lembrando que mesmo utilizando, aparentemente, somente 3 (três) algoritmos, quando se alteram seus parâmetros, pode-se considerar um novo modelo.

Durante a modelagem, os dados são divididos em dados de treino, os quais serão submetidos ao modelo para fazer o seu treinamento, e os dados de teste, os quais são responsáveis por gerar as métricas de avaliação. A técnica escolhida para realizar essa divisão foi a técnica conhecida como **Hold-out** (KOHAVI, 1995), a qual consiste em dividir o conjunto de dados baseado em uma porcentagem pré-definida. Nesse caso foi utilizado 75% da base de dados como treinamento e 25% como teste. Com o intuito de balancear os dados, do total de 323555 ementas, foram retiradas 100 mil amostras de cada resultado, contemplando 200 mil amostras no total.

Antes de submeter os dados aos algoritmos se faz necessário utilizar as técnicas já citadas anteriormente como o pré-processamento, responsável pela limpeza e

normalização das palavras, e a vetorização das palavras a fim de deixar os dados estruturados em uma forma que os algoritmos consigam trabalhar com os mesmos.

Durante a modelagem, também é possível utilizar um número limitado de palavras nos algoritmos. Isso serve para testar se é necessário utilizar todas as palavras para classificar uma ementa, ou se possuir as N palavras mais utilizadas já seria necessário para uma boa classificação.

3.6.1. Experimentos Realizados

Partindo do pressuposto de que quando são alterados os parâmetros de um algoritmo, ele se torna um novo, a Tabela 5 apresenta os 11 (onze) modelos criados, na sequência em que foram testados, a fim de verificar qual teria um melhor resultado. A tabela é composta pelo algoritmo utilizado, o tipo de vetorização das palavras e o total de palavras (características, do inglês *features*) utilizadas.

Tabela 5 - Modelos aplicados

Algoritmo	Tipo de Vetorização	Número de palavras
Naive Bayes 1	Word count	Todas disponíveis
Naive Bayes 2	TF-IDF	Todas disponíveis
Naive Bayes 3	Word count	5 mil
Naive Bayes 4	TF-IDF	5 mil
KNN 1 (k = 5)	Word count	Todas disponíveis
KNN 2 (k = 10)	Word count	Todas disponíveis
KNN 3 (k = 5)	TF-IDF	Todas disponíveis
RF (n=100, entropia)	Word count	Todas disponíveis
RF (n=150, entropia)	Word count	Todas disponíveis
RF (n=100, gini)	Word count	Todas disponíveis
RF (n=150, gini)	Word count	Todas disponíveis

Fonte: elaborado pelo autor

4. Resultados

Para obter a dimensão da quantidade de dados utilizado nesse trabalho pode-se mensurar a quantidade de palavras existentes na amostra. O total de palavras tem aproximadamente 14 milhões, com uma média de 70 palavras por ementa. Abaixo seguem duas figuras que representam as nuvens de palavras para cada resultado das ementas, o qual traduz quais palavras foram mais utilizadas para cada resultado.

Figura 8 - Nuvem de palavras para o resultado Provido

Fonte: elaborado pelo autor

Figura 9 - Nuvem de palavras para o resultado Desprovido

Fonte: elaborado pelo autor

Nas imagens anteriores fica claro que para os dois resultados as palavras com maior ocorrência são iguais. Isso é comum em trabalhos que envolvam mineração de texto. Para exemplificar de uma forma didática, o exemplo de análise de filmes citado anteriormente nesse artigo reflete esse comportamento da repetição de palavras, pois nas opiniões dos usuários há uma grande chance de sempre aparecer a palavra “filme”, tanto para análises positivas quanto para análises negativas. O que vai decidir para qual resultado aquele filme ou aquela jurisprudência pertencerá são palavras que aparecem com mais frequência para somente um desses resultados.

4.1. Quanto ao tamanho em disco

A Tabela 6 tem como objetivo comparar o quanto de espaço em disco cada modelo ocupa. Essa informação pode ser utilizada quando houver necessidade de verificar espaço *versus* performance.

Tabela 6 - Tabela comparativa de espaço em disco dos modelos

MODELO	Tamanho em disco
Naive Bayes 1	143 Kilobytes
Naive Bayes 2	506 Kilobytes
Naive Bayes 3	63 Kilobytes
Naive Bayes 4	148 Kilobytes

KNN 1 (k = 5)	16 Megabytes
KNN 2 (k = 10)	16 Megabytes
KNN 3 (k = 5)	69 Megabytes
RF 1 (n=100, entropia)	82 Megabytes
RF 2 (n=150, entropia)	123 Megabytes
RF 3 (n=100, gini)	83 Megabytes
RF 4 (n=150, gini)	124 Megabytes

Fonte: elaborado pelo autor

4.2. Quanto ao tempo de treinamento

Também foi realizada uma comparação quanto ao tempo que cada modelo leva para realizar o treinamento sobre a base de treino. Na Tabela 7 é possível verificar essa comparação.

Tabela 7 - Tabela comparativa entre os tempos de treinamento de cada modelo

MODELO	**Tempo de treinamento**
Naive Bayes 1	0.426s
Naive Bayes 2	0.432s
Naive Bayes 3	0.442s
Naive Bayes 4	0.392s
KNN 1 (k = 5)	0.199s
KNN 2 (k = 10)	0.159s
KNN 3 (k = 5)	0.171s
RF 1 (n=100, entropia)	643.091s
RF 2 (n=150, entropia)	958.223s
RF 3 (n=100, gini)	653.743s
RF 4 (n=150, gini)	970.125s

Fonte: elaborado pelo autor

Observando a Tabela 7, percebe-se uma grande diferença entre os tempos de treinamento dos modelos de Floresta Aleatória para os demais. Para a utilização dos modelos esse tempo não necessariamente precisa ser levado em conta, pois o modelo posteriormente pode ser exportado já treinado, assim, não necessitando desse tempo novamente para treinamento. Embora esses tempos dos modelos de Floresta Aleatória sejam muito mais expressivos que os demais, eles se justificam pela performance quanto às métricas de assertividade, as quais podem ser visualizadas na subseção seguinte.

4.3. Quanto às métricas avaliativas

Seguindo a metodologia CRISP-DM, chega-se a etapa na qual é necessário avaliar os modelos utilizados, a fim de mensurar qual destes teve uma melhor performance diante dos dados disponíveis. A Tabela 8 tem como objetivo apresentar essa comparação visualmente. Isso leva a uma possibilidade de escolha do melhor algoritmo, a fim de utilizá-lo na etapa final da metodologia CRISP-DM, a aplicação.

Tabela 8 - Tabela comparativa entre as métricas de performance de cada modelo

MODELO	Acurácia	Revocação	Precisão	F1-Score
Naive Bayes 1	72,39%	72,39%	72,40%	72,39%
Naive Bayes 2	72,11%	72,11%	72,12%	72,11%
Naive Bayes 3	72,50%	72,49%	72,62%	72,46%
Naive Bayes 4	72,14%	72,13%	72,27%	72,10%
KNN 1 (k = 5)	80,23%	80,23%	80,33%	80,22%
KNN 2 (k = 10)	79,42%	79,40%	80,06%	79,30%
KNN 3 (k = 5)	79,75%	79,75%	79,76%	79,75%
RF 1 (n=100, entropia)	87,99%	87,98%	88,24%	87,97%
RF 2 (n=150, entropia)	88,10%	88,09%	88,35%	88,08%
RF 3 (n=100, gini)	88,00%	87,99%	88,24%	87,97%
RF 4 (n=150, gini)	**88,16%**	**88,15%**	**88,39%**	**88,14%**

Fonte: elaborado pelo autor

Após a aplicação dos modelos na fase de testes, e a utilização de ferramentas que consigam extrair as métricas de performance dos modelos, baseado na matriz de confusão, fica claro que o modelo de Floresta Aleatória tem uma vantagem de performance em comparação aos outros modelos. Também é possível notar na Tabela 8 que todas as variações de cada modelo, para essa aplicação, não obtiveram uma diferença significativa de ganho de performance.

4.4. Aplicação do Modelo

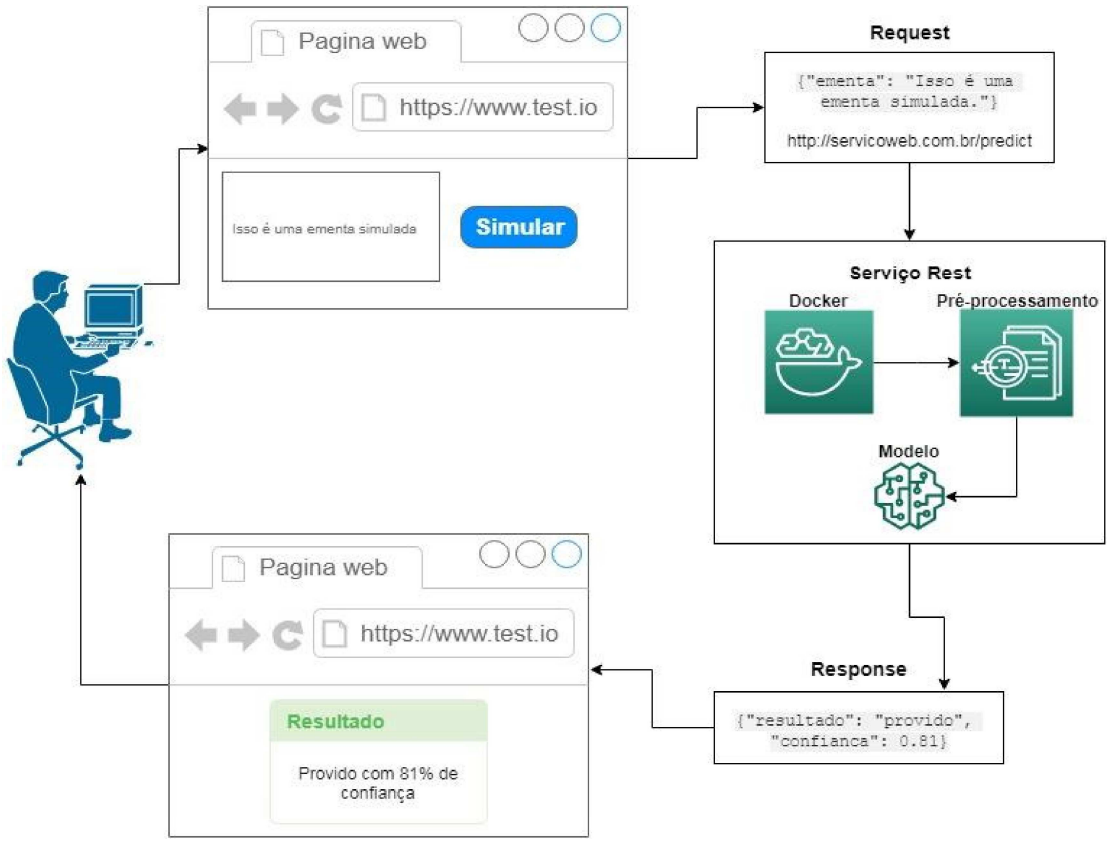
Partindo dos resultados anteriores apresentados e fechando o ciclo da metodologia CRISP-DM, escolheu-se como modelo a ser aplicado, o modelo de Floresta Aleatória, mais conhecido na comunidade internacional como *Random Forest* (BRIEMAN, 2001). Apesar de seu tempo de treinamento ser muito elevado se comparado a de seus concorrentes, o ganho com performance de acertos justifica sua escolha.

A fim de satisfazer então a última etapa da metodologia CRISP-DM, foi criado um serviço web com o objetivo de disponibilizar seu consumo por outras aplicações. Basicamente o serviço consiste em receber um texto, que pode ser redigido por um advogado, simulando uma ementa, com algumas palavras chaves relacionadas ao seu processo. Posteriormente o sistema irá tratar esse texto passando pelo processo de pré-processamento já discutido em tópicos anteriores, e realizar a predição utilizando o modelo já salvo e previamente treinado. Feito isso, o sistema retorna o resultado para aquela 'Ementa Simulada' e um valor referente à confiança daquela predição.

A ideia é realizar o consumo desse serviço através de uma aplicação com interface amigável, ainda não desenvolvida, onde o usuário faria a entrada do texto, citado anteriormente, e receberia a resposta do possível resultado visualmente.

A Figura 10 demonstra uma estrutura básica do funcionamento do serviço web. Um usuário, por exemplo, um advogado, submete uma ementa simulada através de alguma interface disponibilizada, seja um aplicativo mobile ou uma página na web, então é criada uma requisição para o serviço web. Essa requisição contendo a ementa simulada é interpretada pela aplicação na qual é aplicado um pré-processamento nesse texto, como já descrito anteriormente nesse artigo. Na sequência, esse texto é submetido ao modelo já treinado previamente e, então, o resultado da classificação da ementa simulada é devolvido ao usuário.

Figura 10 - Diagrama do Serviço Web



Fonte: elaborado pelo autor.

Como tecnologias utilizadas estão o *Firefly* (FIREFLY, 2019), framework desenvolvido em *Python* para criação de funções *python* como um serviço web *Rest*; *Docker* (DOCKER, 2019), para empacotar o serviço em um container isolado; e um serviço em Nuvem (*Amazon AWS, Heroku, Google Cloud*, entre outras) para a hospedagem da aplicação contendo interface e o serviço web.

4.5. Testes Isolados

Com o intuito de aferir a precisão do modelo escolhido, foram retiradas do site do TJRS duas ementas, uma para cada resultado, fora da amostra escolhida, ou seja, totalmente isoladas. Essas ementas foram tratadas no mesmo processo utilizado nas ementas usadas para gerar o modelo, inclusive retirando o resultado do processo das mesmas. Segue abaixo a predição do modelo para as duas ementas.

Ementa 1 (Resultado Provido):

CONDOMÍNIO. AÇÃO DE COBRANÇA DE COTAS CONDOMINIAIS. GRATUIDADE DA JUSTIÇA INDEFERIDA. PESSOA JURÍDICA. PRESENTE PROVA ACERCA DA NECESSIDADE DO BENEFÍCIO. SÚMULA 481 DO SUPERIOR TRIBUNAL DE JUSTIÇA. DEFERIMENTO. DECISÃO AGRAVADA REFORMADA. I. O benefício da gratuidade da justiça deve ser concedido à pessoa natural ou jurídica, brasileira ou estrangeira, com insuficiência de recursos para pagar custas, despesas processuais e até honorários advocatícios, nos termos do artigo 98 do CPC. II. Nos termos do previsto na Súmula 481 do Superior Tribunal de Justiça faz jus ao benefício

da justiça gratuita a pessoa jurídica com ou sem fins lucrativos que demonstrar sua impossibilidade de arcar com os encargos processuais. III. No caso, presente prova no sentido da necessidade quanto ao pagamento das custas e honorários que a parte eventualmente venha a suportar, impõe-se a concessão da gratuidade da justiça.

Aplicação do pré-processamento:

condomini aca cobranc *cot condomin gratu jus indefer pesso jurid pres prov acerc necess benefici sumul 481 superi tribun jus defer decisa agrav reform i benefici gratu jus dev ser conced pesso natur jurid brasil estrang insuficienc recurs pag cust desp process ate honorari advocatici term artig 98 cpc ii term previst sumul 481 superi tribun jus faz ju benefici jus gratuit pesso jurid fim lucr demonstr impossibil arc encarg process iii cas pres prov sent necess quant pag cust honorari part event venh suport impo concessa gratu jus*

O modelo teve como resultado: ['PROVIDO', 0.89]. Isso significa que o modelo acertou o resultado dessa ementa com 89% de confiança.

Ementa 2 (Resultado Desprovido):

NEGÓCIOS JURÍDICOS BANCÁRIOS. AÇÃO REVISIONAL DE CONTRATOS. GRATUIDADE DA JUSTIÇA. AUSENTE PROVA ACERCA DA NECESSIDADE DO BENEFÍCIO. INDEFERIMENTO. DECISÃO AGRAVADA MANTIDA. I. Consoante redação do artigo 98 do Novo Código de Processo Civil, tem direito à gratuidade da justiça a pessoa natural ou jurídica, brasileira ou estrangeira, com insuficiência de recursos para pagar custas, despesas processuais e honorários advocatícios. II. Ausente prova ou indício no sentido da necessidade quanto ao pagamento das custas e honorários que a parte eventualmente venha a suportar, uma vez que contracheques acostados comprovam, de maneira inequívoca, que o recorrente percebe renda próxima do dobro estabelecido na Resolução n. 49ª do CETJRS. Decisão agravada mantida.

Aplicação do pré-processamento:

negoci jurid bancari aca revis contrat gratu jus ausent prov acerc necess benefici indefer decisa agrav mant i conso redaca artig 98 nov codig process civil direit gratu jus pesso natur jurid brasil estrang insuficienc recurs pag cust desp process honorari advocatici ii ausent prov indici sent necess quant pag cust honorari part event venh suport vez contrachequ acost comprov man inequivoc recorr perceb rend prox dobr estabelec resoluca n 49a cetjr decisa agrav mant

O modelo obteve como resultado: ['DESPROVIDO', 0.71]. Isso significa que o modelo acertou o resultado novamente com uma confiança de 71%.

4.6. Análise dos Resultados e Discussão

Um fato curioso que pode ser observado nos resultados é com relação à discrepância do tempo de treinamento de cada modelo. Conforme pode ser visualizado na Tabela 7, os modelos de Floresta Aleatória chegam a ter um custo de tempo aproximado de 3370 vezes maior na fase de treinamento que os modelos de KNN, e aproximadamente 1500 vezes maior que os modelos de Bayes Ingênuo.

Porém, esses tempos se justificam depois na Tabela 8, a qual mostra que os modelos de Floresta Aleatória ficaram 8% melhores nas métricas avaliativas com

relação aos modelos de KNN, e 17% melhores com relação aos modelos de Bayes Ingênuo.

Olhando por uma ótica mais criteriosa, os modelos de Floresta Aleatória, quando alterado o parâmetro referente ao número de árvores que compunham a Floresta, nota-se que há um acréscimo de aproximadamente 50% no seu tempo de treinamento. Esse acréscimo pode ser observado na Tabela 7. Porém se observada a Tabela 8 as métricas praticamente se mantém iguais. O que leva a considerar que não se justifica o incremento desse parâmetro, já que o resultado do modelo é praticamente igual, porém com um tempo maior de treinamento.

5. Conclusão

Esse estudo objetivou compreender e aplicar técnicas de inteligência artificial na área jurídica, a fim de otimizar e melhorar a função de um ator jurídico durante a sua atuação. Para isso, durante o estudo foi necessária a inclusão de uma etapa de entendimento do negócio, para compreender o que um advogado realiza quando há a necessidade de utilizar-se de jurisprudências para embasar suas teses. Assim, uma das motivações de realização dessa pesquisa, foi a possibilidade de melhorar o processo de trabalho de uma área ainda em desenvolvimento quando envolve aplicação de inteligência artificial.

De um modo geral, os objetivos definidos para a pesquisa foram alcançados. Através do presente estudo foi gerado um modelo de aprendizado de máquina utilizando mineração de texto a fim de classificar ementas de jurisprudências quanto ao seu resultado. Utilizando um embasamento teórico foi possível aplicar variados modelos de aprendizado de máquina objetivando medir seus desempenhos para a escolha do mais satisfatório. Com as métricas obtidas, foi escolhido então o modelo gerado a partir do algoritmo de Floresta aleatória, o qual obteve uma acurácia de aproximadamente 87%.

Para chegar nesse objetivo existiram algumas etapas que possibilitaram isso. Dentre essas etapas, a mais importante foi o entendimento do mundo jurídico e suas particularidades. Posteriormente, foi realizada a coleta de dados e seu pré-processamento para, assim, ser possível a utilização desses dados pelos algoritmos de aprendizado de máquina. Os algoritmos utilizados foram os mais conhecidos na literatura para classificar textos (Floresta Aleatória, Bayes Ingênuo e KNN).

Pode-se concluir então que o trabalho obteve êxito no alcance dos seus objetivos. Conseguiu-se gerar um modelo que foi satisfatório para o estudo se observadas todas as métricas. Apesar disso, ainda se vislumbram possíveis melhoras nos resultados, pois há muitas oportunidades dentro da área de inteligência artificial aplicadas à área jurídica.

Notou-se ao longo da etapa de pré-processamento, na criação das bases de treino/teste, que ainda existem erros durante a etapa de definição da variável resposta, nesse caso da variável “RESULTADO”, pois alguns cenários não foram cobertos, ocasionando na geração de uma variável resposta incorreta. Por exemplo, uma ementa que teria um resultado “Provido”, foi definida como “Desprovido”. Uma sugestão para trabalhos futuros seria encontrar uma solução melhor para esse problema identificado.

Outra possibilidade é aplicar redes neurais, área essa que compreende o chamado aprendizado profundo. No artigo de Lai *et al.* (2015), eles comparam a

aplicação de redes neurais na classificação de texto com métodos tradicionais, como os utilizados no presente trabalho. Nos resultados apresentados fica evidente a melhora na classificação dos textos, pois redes neurais levam em consideração a semântica das palavras no texto o qual elas pertencem (LAI *et al.*, 2015).

Voltando para os métodos tradicionais, também é possível no momento em que se está vetorizando o texto, a utilização dos chamados n-gramas, isto é, não mais se utiliza palavra por palavra, mas sim combinações de palavras. Por exemplo, quando o texto "O processo foi julgado como procedente" é vetorizado para aí sim ser utilizado pelos algoritmos, ele iria ser separado por palavras como: "O", ''processo", "foi", "julgado", "como", "precedente", estes chamados de unigramas. Porém quando se utiliza n-gramas, nesse caso a vetorização cria combinações das palavras subsequentes, isto é, utilizando a mesma frase anterior, a vetorização ficaria assim: "O processo", "processo foi", "foi julgado", "julgado como", "como precedente".

No exemplo anterior foi utilizado o chamado bigrama, combinação de 2 palavras, mas pode ser utilizado quantas forem necessárias. O problema disso é que pode aumentar a dimensionalidade do conjunto de características que será utilizado pelo algoritmo, degradando assim seu tempo de treinamento. Porém, o ganho em performance de classificação pode compensar.

Dessa forma, pode-se perceber que há uma gama de possibilidades a fim de melhorar os resultados apresentados nesse trabalho.

6. Referências

ARANHA, Christian; PASSOS, Emmanuel. **A Tecnologia de Mineração de Textos**. RESI-Revista Eletrônica de Sistemas de Informação. 2006.

AZEVEDO, Ana Isabel Rojão Lourenço; SANTOS, Manuel Filipe. **KDD, SEMMA and CRISP-DM: a parallel overview**. 2008.

BARROS, Rhuan Paulo Lopes. **Análise Jurisprudencial com técnica de aprendizado de máquina.** Porto Alegre. 2018.

BRASIL. **Código de Processo Civil Brasileiro**. Brasília, DF: Senado, 2015.

BREIMAN, Leo. (1996). **Bagging predictors**. Machine Learning, 24:123–140.

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

CAMARA JUNIOR, A. T. **Indexação automática de acórdãos por meio de processamento de linguagem natural**. Dissertação de Mestrado, Departamento de Ciência da Informação e Documentação, Universidade de Brasília, Brasília, 2007. Disponível em: <http://enancib.ibict.br/index.php/enancib/viiienancib/paper/viewFile/2818/1946>. Acesso em: 17 de maio de 2019.

COELHO, Alexandre Zavaglia. **As 7 tendências para o uso de inteligência artificial no Direito em 2018**. 2018.

COLABORATORY. **Welcome to Colaboratory**. Disponível em: <https://colab.research.google.com>. Acesso em: 10 de Julho de 2019.

Conselho Nacional de Justiça (CNJ). **Primeira instância, segunda instância... Quem é quem na Justiça brasileira?**. 2012. Disponível em <https://cnj.jusbrasil.com.br/noticias/100111134/primeira-instancia-segunda-instancia-quem-e-quem-na-justica-brasileira>. Acesso em: 29 de maio de 2019.

DOCKER. **Docker platform**. Disponível em: <https://www.docker.com/>. Acesso em: 10 de Julho de 2019.

FERAUCHE, Thiago; ALMEIDA, Maurício Amaral de. **Aprendizado de classificadores de ementas da jurisprudência do Tribunal Regional do Trabalho da 2ª Região–SP**. In: VI WorkShop de Pesquisa do Centro Estadual de Educação Tecnológica Paula Souza–SP–Brasil. 2011.

FACELLI, K., LORENA, A. C., GAMA, J., & CARVALHO, A. C. P. L. F. **Inteligência Artificial: Uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC. 2011.

FELDMAN, Ronen; SANGER, James. **The text mining handbook: advanced approaches in analyzing unstructured data**. Nova York: Cambridge, 2006.

FIREFLY. **Function as a service framework**. Disponível em: <https://firefly-python.readthedocs.io/en/latest/>. Acesso em: 10 de Julho de 2019.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 4. ed. São Paulo: Atlas, 2002.

ISAYED, Anwar. **Classification of Error Related Potential (ErrP) in P300-Speller.** Palestine, 2015.

KOHAVI, R. **A study of cross-validation and bootstrap for accuracy estimation and model selection**. In: International joint Conference on artificial intelligence. [S.l.: s.n.], 1995. v. 14, p. 1137–1145.

LAI, Siwei; XU, Liheng; LIU, Kang; ZHAO, Jun. **Recurrent convolutional neural networks for text classification**. In Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.

MENEZES, Daniel; BARROS, Gisele Porto. **Breve análise sobre a Jurimetria, os desafios para sua implementação e as vantagens correspondentes**. Revista Duc In Altum Cadernos de Direito, Recife, vol. 9, nº19, p. 45-83, set.-dez. 2017.

MITCHEL, Tom. **Machine Learning**. McGraw Hill. 1997.

MONTENEGRO FILHO, Misael. **Curso de Direito Processual Civil***, 12ª edição.* 2016.

NETTO, Ernesto. **A influência da jurisprudência no direito brasileiro**. Disponível em <https://www.direitonet.com.br/artigos/exibir/5872/A-influencia-da-jurisprudencia-no-direito-brasileiro-Parte-I>. Acesso em: 9 junho 2019.

NLTK. **Natural Language Toolkit**. Disponível em: <http://www.nltk.org>. Acesso em: 14 de Julho de 2019.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial**. 3ED. [s.l.] : Rio de Janeiro, Elsevier, 2013. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=cat07348a&AN=urds.9788535251418&lang=pt-br&site=eds-live>. Acesso em: 9 junho 2019.

PANDAS. **Python Data Analysis Library**. Disponível em: <https://pandas.pydata.org>. Acesso em: 10 de Julho de 2019.

PYTHON. **Python**. Disponível em: <https://www.python.org>. Acesso em: 28 de Julho de 2019.

RICHARDSON, Robert Jarry. **Pesquisa social: métodos e técnicas**. 3. ed. São Paulo: Atlas, 1999.

SCIKIT-LEARN. **scikit-learn Machine Learning in Python**. Disponível em: <https://scikit-learn.org/stable>. Acesso em: 10 de Julho de 2019.

Superior Tribunal Federal (STF). 2019. **Glossário Jurídico**. Disponível em <http://www.stf.jus.br/portal/glossario/>. Acesso em: 30 de maio de 2019.

TIOBE. **Tiobe Index for Python**. Disponível em: <https://www.tiobe.com/tiobeindex/python>. Acesso em: 15 de Julho de 2019.