Comparação de métodos de mineração de texto para classificação de documentos jurídicos

Bruno A. Medeiros[1], Eric C. Marcelino da Silva[1]

[1]Curso de Ciência da Computação - Universidade do Sul de Santa Catarina (UNISUL)
Tubarão – SC – Brasil

{bruno.am19, ericcristhiano}@gmail.com

Abstract. *Due to the increasing digitalization of court documents and eletronics lawsuits, it can be said that there was a significant increase in the amount of unstructured legal texts available, creating the possibility to use the mining of data for extraction of important informations. This article is based in describing the stages of the text mining process and the analysis of different methods of classification, using learning algorithms from supervised machines in jurisprudence, in order to evaluate the method that better stands out in this gathering of amplified diverse data.*

Resumo. *Devido à crescente digitalização de documentos e processos eletrônicos, pode-se afirmar que houve um aumento significativo da quantidade de textos jurídicos não estruturado disponível, possibilitando a utilização da mineração de dados para a extração de informações relevantes. Este artigo propõe-se a descrever as etapas do processo de mineração de texto e a análise de diferentes métodos de classificação, utilizando algoritmos de aprendizado de máquina supervisionados a partir de jurisprudências, com o objetivo de avaliar o método que melhor se enquadra neste conjunto de dados amplamente diversificado.*

1. Introdução

O processo de digitalização de documentos judiciais tornou-se uma necessidade devido ao grande volume de processos que tramitam diariamente no Brasil. Uma reportagem apresentada pelo Portal de Notícias G1[1] em 2009 (ESTADO, 2019) informou que o peso de cerca de dois milhões de processos, num total de 100 toneladas, comprometeu a estrutura física de um fórum de São Paulo. A mesma reportagem apresentou uma pesquisa realizada pelo Conselho Nacional de Justiça (CNJ), chamada Justiça em Números, onde segundo o órgão, no fim de 2018 havia 78,69 milhões de ações inconclusas tramitando nos tribunais. Nesta última apuração foram ajuizados 28,1 milhões de novos processos. O estoque total de processos que, em 2018, aguardava por uma decisão judicial, aumentou cerca de 30% no período entre 2009 e 2019; durante este mesmo tempo, o total de ações em tramitação judicial teve um incremento substancial de 60,7 milhões para 78,69 milhões.

Com estas informações levantadas, uma das soluções encontradas foi a digitalização de documentos. O uso deste artifício criou um conjunto de dados que possibilita a exploração para diversos fins, que após processados e classificados são comumente chamados de jurimetria.

[1] G1 é um portal de notícias brasileiro

Este artigo se propõe a desenvolver o processamento de um conjunto de dados, aplicar os métodos de mineração de texto mais utilizados para este tipo de problema, analisando e apresentando a eficácia de cada um deles, através de modelos de aprendizado de máquina supervisionados, treinados a partir de jurisprudências obtidas na ferramenta de busca do Tribunal de Justiça de Santa Catarina (TJSC). Para que seja possível aplicar o método com menor custo computacional e com melhor resultado na obtenção de informações relevantes.

A segunda seção deste artigo apresenta alguns trabalhos relacionados que trazem a aplicação de técnicas de mineração de texto, no âmbito de classificação, agrupamento e aplicação para diferentes situações relacionadas ao Direito. Na terceira seção são apresentados os conceitos abordados na elaboração deste projeto. Materiais, ferramentas e métodos aplicados são apresentados na seção 4. A seção 5 é composta pelos resultados obtidos com a aplicação dos modelos estudados. Por fim, a seção 6 contém as considerações finais e a conclusão do trabalho desenvolvido.

2. Trabalhos Relacionados

Araújo Neto (2015) desenvolveu uma ferramenta apta a agrupar e categorizar automaticamente atos processuais digitais. As técnicas empregadas para tal foram pesquisa harmônica, algoritmo genético e *K-means*[2]. Utilizaram-se os índices de dissimilaridade, distância Euclidiana, do Coseno e de Hamming e, posteriormente, C³M, por categorizar-se como um problema de otimização. Os resultados obtidos extraídos após a utilização das técnicas indicam que dentre os métodos de agrupamento de pesquisa harmônica, o algoritmo genético e C³M, o *K-means* foi considerado o algoritmo mais apropriado para resolver o problema de particionamento de texto.

Ticom (2007) desenvolveu uma dissertação com o objetivo de apresentar resultados na aplicação das técnicas de mineração de dados em textos não estruturados utilizando-se das metodologias probabilística, linear por ordenação e de indução de regras na categorização de textos. Além disso, utilizou sistemas especialistas em sentenças judiciais da área trabalhista. Os resultados obtidos mostraram que para situações singulares existem métodos diferentes que são mais apropriados. Em aspectos gerais, o método Linear, por exemplo, obteve melhor desempenho em relação aos demais, no entanto, para um determinado indicador, o método *Naive Bayes* se sobressaiu. Como se utilizou um grande volume de dados, o método *K-nearest-neighbors*[3], com as medidas de distância euclidiana, manhattan, camberra e minimax, não completou o processamento necessário, pois este método na época exigia uma grande capacidade de recurso de máquina.

Barros (2019), teve por objetivo realizar a análise de documentos judiciais da área trabalhista utilizando-se de técnicas de mineração de dados, com o intuito de analisar a opinião da turma de julgamento com base nas suas decisões proferidas e classificando-as em relação à parte favorecida, sendo ela empregador ou empregado. Previamente, foi aplicado um algoritmo para a remoção de documentos em que empregado e empregador recorreram. Estes processos seriam inconclusivos e prejudicariam a acurácia do modelo. Os documentos extraídos foram os acórdãos judiciais. Após a anotação manual desses documentos, fez-se alimentação de um algoritmo de aprendizado de máquina baseado em redes Bayesianas com a biblioteca *Scikit-Learning*. Além disso, as decisões foram processadas com extração de

[2] K-means é um método de Clustering utilizado em mineração de texto
[3] K-nearest-neighbors é um método não paramétrico usado para classificação e regressão

features através de índice TF-IDF [4] e Redes Bayesianas. Utilizando este método para classificação da parte vencedora entre empregador e empregado, o algoritmo alcançou 92% de acurácia.

Oliveira (2015) utilizou em sua dissertação a mineração de dados e o Processamento de Linguagem Natural com o objetivo de classificar documentos jurídicos. Para realizar essa classificação utilizou-se do classificador *Naive-Bayes* por conta de sua aplicação em outros projetos com temas jurídicos. Foram realizados testes com esse classificador em petições de origem Penal, Trabalhista, Civil, Ação de Despejo e de Cobrança. Os resultados sofreram bastante variações dependendo do tipo de classificação, da quantidade de documentos testados e da base de documentos treinados. O resultado final, considerando a média total de acertos do classificador, foi de cerca de 80%.

3. Contextualização

3.1. Jurisprudência e Conceitos Relacionados ao Direito

3.1.1. Conceito

Jurisprudência é o conjunto das decisões dos tribunais, no exercício da aplicação da lei. Representa a visão do tribunal, em determinado momento, sobre as questões legais levadas a julgamento (TSE, 2019).

Ao longo da história o vocábulo sofreu variações, de origem latina, empregado na Roma antiga para designar a Ciência do Direito, mas atualmente é aplicado com pouca frequência. Para Nader (2001) atualmente o vocábulo é adotado para indicar os precedentes judiciais, ou seja, a reunião de decisões judiciais a respeito do mesmo assunto, interpretadoras do Direito vigente.

Então, pode-se resumir que, a jurisprudência constitui a definição do Direito elaborada pelos tribunais com intuito de tornar o conhecimento mais acessível sendo formada por casos em que foi decidida a maneira mais adequada de cumprir a norma jurídica.

3.1.2. Aplicação

Para fundamentar uma pretensão judicial, ou seja, exigir algo que satisfaça um interesse legítimo, econômico ou moral, os advogados identificam sentenças proferidas pelos tribunais, com pertinência ao caso específico. Por exemplo, em termos práticos é usual que nas petições advogados utilizem jurisprudências como argumentações favoráveis em suas teses, pois pode ser útil que o magistrado saiba da existência de outras decisões que tratem do mesmo assunto, tornando-se base para a pesquisa. Ressalta-se que a existência de outras decisões similares, pode não ser suficiente para influenciá-lo (DIMOULIS, 2013).

[4] Term Frequency / Inverse Document Frequency é uma técnica para classificar palavras em uma coleção de documentos por ordem de relevância

3.2. KDT - *Knowledge Discovery from Text*

Gomes (2013) define KDT ou Descoberta de Conhecimento em Texto, como a área onde são aplicadas técnicas e ferramentas computacionais com o objetivo de auxiliar na busca de conhecimento novo e útil em coleções textuais. Os processos a serem executados não são triviais, pois é necessário um alto grau de complexidade no tratamento das informações coletadas.

O KDT combina muitas técnicas de extração de informação, recuperação de informação, processamento de linguagem natural e classificação de documentos utilizando os métodos de mineração de dados. O uso principal é a extração de conhecimento anteriormente desconhecido em um volume de texto.

3.3. Mineração de Texto

A mineração de texto está inserida no ambiente da mineração de dados, sendo empregada quando um alto volume de informações de texto, armazenadas de forma não estruturada, é dificilmente analisado; exigindo a aplicação de métodos específicos de processamento para extrair padrões úteis.

Mineração de texto é um processo de análise utilizado para encontrar aspectos ocultos e extrair informações relevantes com técnicas de extração de dados e processamento de linguagem natural (PLN) e, as combina com mineração de dados, aprendizado de máquina e estatística (SANTOS, 2013).

3.4. Aprendizado de Máquina Supervisionado

O aprendizado de máquina supervisionado é um dos tipos mais comuns de aprendizado de máquina. É utilizado sempre que se precisa prever um determinado resultado de uma determinada entrada, usando pares de entrada e saída.

3.4.1 Classificação e Regressão

Existem dois tipos principais de problemas de aprendizado de máquina supervisionados, denominados classificação e regressão (MÜLLER; GUIDO, 2016).

Na classificação a tarefa é identificar, por semelhança, objetos entre diversas categorias pré-definidas. A classificação às vezes é separada em classificação binária, que é o caso especial de distinguir entre exatamente duas categorias, e classificação multiclasse, que é a classificação entre mais de duas categorias. Citamos exemplos: detecção de mensagens de *spam* em e-mails baseadas no cabeçalho e conteúdo da mensagem, classificação de galáxias baseada em seus formatos, entre outros.

Para tarefas de regressão, o objetivo é prever um número contínuo, ou um número de pontos flutuantes em termos de programação (ou números reais em termos matemáticos). Um exemplo de tarefa de regressão é prever a renda anual da pessoa a partir de sua educação, idade e local onde mora.

Uma forma fácil de distinguir tarefas de classificação e regressão é perguntar se existe algum tipo de continuidade na saída. Se houver continuidade entre os resultados possíveis, é um problema de regressão.

3.5 TF-IDF

TF-IDF significa *term frequency-inverse document frequency*, e o peso TF-IDF é um peso frequentemente usado na recuperação de informações e na mineração de texto. Müller e Guido (2016) explicam que este peso é uma medida estatística usada para avaliar a importância de uma palavra para um documento em uma coleção ou *corpus*. A importância aumenta proporcionalmente ao número de vezes que uma palavra aparece no documento, mas é compensada pela frequência da palavra no *corpus*. Variações do esquema de ponderação TF-IDF são frequentemente usadas pelos mecanismos de busca como uma ferramenta central na pontuação e classificação da relevância de um documento, dada uma consulta do usuário. A pontuação da palavra w no documento d implementado, é fornecida por:

$$tfidf(w,d) = tf * log(\frac{N + 1}{N_w + 1}) + 1$$

Onde N é o número de documentos no conjunto de treinamento, N_wé o número de documentos no conjunto de treinamento em que a palavra w aparece e tf (o termo frequência) é o número de vezes que a palavra w aparece no documento de consulta d.

O TF-IDF pode ser usado com êxito para filtragem de *stop words* em vários campos de assunto, incluindo resumo e classificação de texto.

3.5.1 Vetores de Contagem TF-IDF

Vetores TF-IDF podem ser gerados em diferentes níveis de *tokens* (palavras, caracteres, n-gramas).

- Nível de palavra: Matriz representando a pontuação TF-IDF de todo termo em diferentes documentos.
- Nível de N-Grama: N-gramas são a combinação de N termos juntos. Essa matriz representa a pontuação TF-IDF de N-gramas. É uma forma de unir palavras para adicionar mais contexto nos recursos.
- Nível de Caractere: Matriz representando a pontuação TF-IDF de cada carácter n-grama no corpus.

3.6. Modelos para Classificação

3.6.1 *Support Vector Machine - SVM*

Este é um dos classificadores mais populares do tipo linear. As SVMs implementam a ideia de construir um hiperplano com base no mapeamento dos vetores de entrada em um espaço de características com uma grande quantidade de dimensões. As SVMs são embasadas pela teoria de aprendizado estatístico, desenvolvida por Vapnik (1992).

De acordo com o estudo do Joachims (1998), SVMs conseguem consistentemente bom desempenho em tarefas de categorização de texto, superando substancialmente e significativamente os métodos existentes. Com sua capacidade de generalizar bem em espaços de recursos de alta dimensão, os SVMs eliminam a necessidade de seleção de recursos, facilitando consideravelmente a aplicação da categorização de texto.

3.6.2 Regressão Logística

Apesar do nome, regressão logística é um algoritmo de classificação bastante utilizado na área de estatística há um longo tempo, que começou a ser utilizado na área de aprendizado de máquina recentemente, devido à próxima relação com o SVM. Ele analisa o relacionamento entre diversas variáveis independentes e uma variável dependente categórica, estimando a probabilidade de ocorrência de um evento ajustando os dados a uma curva logística, usada para prever um resultado binário (1 / 0, Sim / Não, Verdadeiro / Falso) (SANTOS, 2013).

Levando em consideração os estudos apresentados no artigo de Zhang e Oles (2001), onde a aplicação do modelo de regressão logística pode funcionar tão bem quanto uma máquina de vetor de suporte (SVM), foi o motivo para a implementação e análise dos resultados utilizando este modelo.

3.6.3. Modelo Naive Bayes

Segundo Tan, Steinbach e Kumar (2009), esse algoritmo é utilizado em aplicações cujo relacionamento entre o conjunto de atributos e a variável classe é não determinístico, ou seja, o rótulo da classe de um registro de teste não pode ser previsto com certeza, embora seu conjunto de atributos seja idêntico a alguns exemplos de treinamento.

Existem dois modelos de classificadores Naive Bayes, um modelo que especifica um documento representado por um vetor de atributos binários, indicando quais palavras ocorrem e não ocorrem no documento. Isso descreve uma distribuição baseada em um modelo de evento Bernoulli multivariável. O segundo modelo especifica que um documento é representado pelo conjunto de ocorrências de palavras do documento, chamado de modelo de evento multinomial.

De acordo com as conclusões de Mccallum e Nigam (1998), o modelo multinomial é encontrado quase uniformemente melhor que o modelo multivariado de Bernoulli. Diante destas conclusões, optamos por implementar neste projeto o método multinomial.

3.6.4. Floresta Aleatória

Floresta Aleatória é uma coleção projetada especificamente para árvores de decisão, onde cada árvore é um pouco diferente da outra. A ideia é combinar as previsões feitas por várias árvores de decisão, geradas com base nos valores de um conjunto independente de vetores aleatórios (MÜLLER; GUIDO, 2016).

Provando ser eficaz em uma ampla gama de conjuntos de dados para classificação e regressão, pois, construindo muitas árvores que funcionem bem e superestimam de maneiras diferentes, podemos reduzir a quantidade de sobreajustes, mantendo os seus resultados, sendo eficaz para o conjunto de dados apresentado neste artigo.

4. Materiais e Métodos

4.1. Fonte de Dados

Para a obtenção dos textos utilizados nos testes foi empregada a ferramenta de busca online do site do Tribunal de Justiça de Santa Catarina. Na realização deste estudo, os documentos escolhidos foram as jurisprudências disponíveis no TJSC. Os resultados da busca são disponibilizados em diversos formatos, o PDF é um deles, sendo escolhido para

ser analisado pela aplicação desenvolvida. Foram baixados cerca de 60 mil links da jurisprudência em PDF, para posteriormente serem filtrados e utilizados para a aplicação da mineração de texto.

4.2. Metodologia

As etapas para a execução deste estudo podem ser definidas em (1) Coleta dos dados, (2) Preparação dos dados, (3) Seleção e aplicação dos algoritmos (4) Avaliação dos resultados, conforme ilustrado na figura 1.

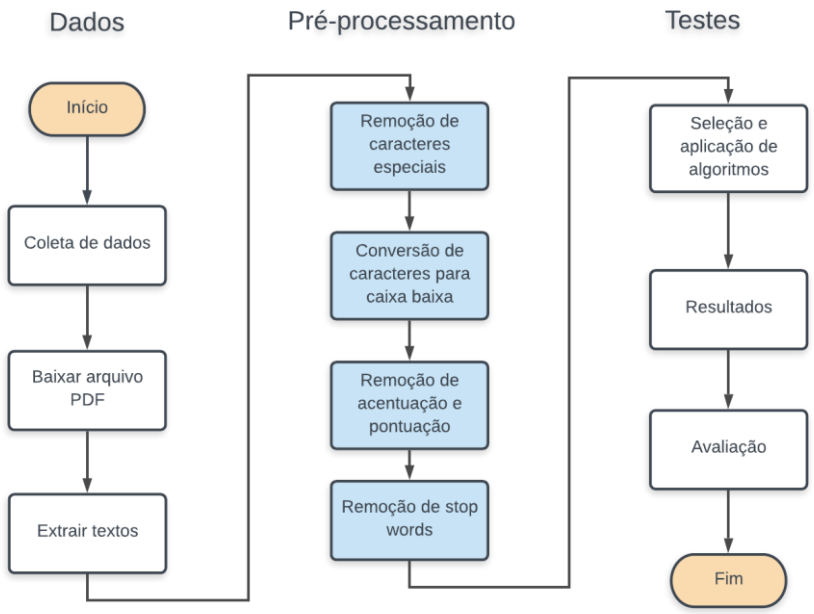

Figura 1. Etapas da elaboração do estudo
Fonte: Os autores

4.2.1 Coleta dos Dados

O banco de dados inicial foi criado a partir de um *web crawler,* ou rastreador da rede, para coletar as informações das jurisprudências disponíveis no site do Tribunal de Justiça de Santa Catarina, baseando-se nas sugestões fornecidas por um profissional da área de direito. Os dados armazenados foram: a identificação do processo, a origem, o órgão julgador, a data, a classe e o endereço para *download* na íntegra em formato PDF.

Na primeira fase foram obtidos cerca de 60 mil registros de jurisprudências. Para tornar o processo mais ágil, resolveu-se filtrar as jurisprudências pelas classes que possuíam mais documentos disponíveis. Sendo assim, uma nova base de dados foi criada contendo cinco classes que possuem a maior quantidade de registros disponíveis: Agravo de Instrumento, Apelação Criminal, Apelação Cível, Embargos de Declaração e Recurso Especial.

4.2.2 Preparação dos Dados

Após obter essa segunda base de dados com os resultados mais concisos, foi criado e executado um *script* para percorrer o banco de dados linha por linha e realizar o *download*

dos arquivos em PDF. Em seguida, realizou-se a conversão do PDF para texto e foram aplicadas as técnicas de pré-processamento.

As técnicas de pré-processamento de texto aplicadas foram: (1) Remoção de caracteres especiais, (2) Conversão de todos os caracteres para caixa baixa (3) Remoção de caracteres de acentuação e pontuação e (4) Remoção de *stop words*.

Posteriormente, os textos extraídos do PDF e os textos já pré-processados foram salvos em dois novos campos no banco de dados, para que estivessem disponíveis para serem aplicados nos algoritmos de mineração.

4.2.3 Seleção e Aplicação dos Algoritmos

Uma vez que todos os dados estavam inseridos no banco de dados, iniciou-se a aplicação dos algoritmos e das diferentes técnicas para realizar o comparativo e observar qual entregaria o melhor resultado frente aos documentos apresentados.

Após uma pesquisa intensa sobre os melhores métodos para a mineração de textos as técnicas Floresta Aleatória, SVM, Regressão Logística e Naive Bayes foram as selecionadas, utilizando para cada método os vetores de contagem TF-IDF em nível de palavra, TF-IDF em nível de caractere, TF-IDF em nível de N-grama e contador de palavras.

4.2.4 Avaliação da Classificação Multiclasse

A métrica mais comumente utilizada para conjuntos de dados desequilibrados nas configurações de várias classes é a versão multiclasse do f-score. A ideia por trás do f-score multiclasse é calcular utilizando a abordagem *one-vs.-rest*. Na abordagem *one-vs.-rest*, é aprendido um modelo binário para cada classe que tenta separar essa classe de todas as outras classes, resultando em tantos modelos binários quanto existem classes. Para fazer uma previsão, todos os classificadores binários são executados em um ponto de teste. O classificador que tem a pontuação mais alta em sua classe única "vence" e esse rótulo de classe é retornado como predição (MÜLLER; GUIDO, 2016).

Em seguida, esses *f-scores* por classe são calculados usando uma das seguintes estratégias:

- Média macro: calcula os *f-scores* não ponderados por classe, dando o mesmo peso a todas as classes, independentemente do tamanho.
- Média ponderada (*weighted*): calcula a média dos f-scores por classe, ponderados por seu apoio. É isso que é exposto no relatório de classificação.
- Média micro: calcula o número total de falsos positivos, falsos negativos e positivos verdadeiros em todas as classes e, em seguida, calcula a precisão, *recall* e *f-score* usando essas contagens.

Considerando que o conjunto de dados analisado é desequilibrado, a melhor estratégia para calcular o f-score foi através da média micro.

5. Resultados e Discussões

Dentre as técnicas analisadas, a Floresta Aleatória utilizando vetor TF-IDF em nível de palavra foi a destaque por atingir o f_1-*score* de 79.2%. Os resultados da matriz de confusão podem ser vistos logo abaixo na figura 2.

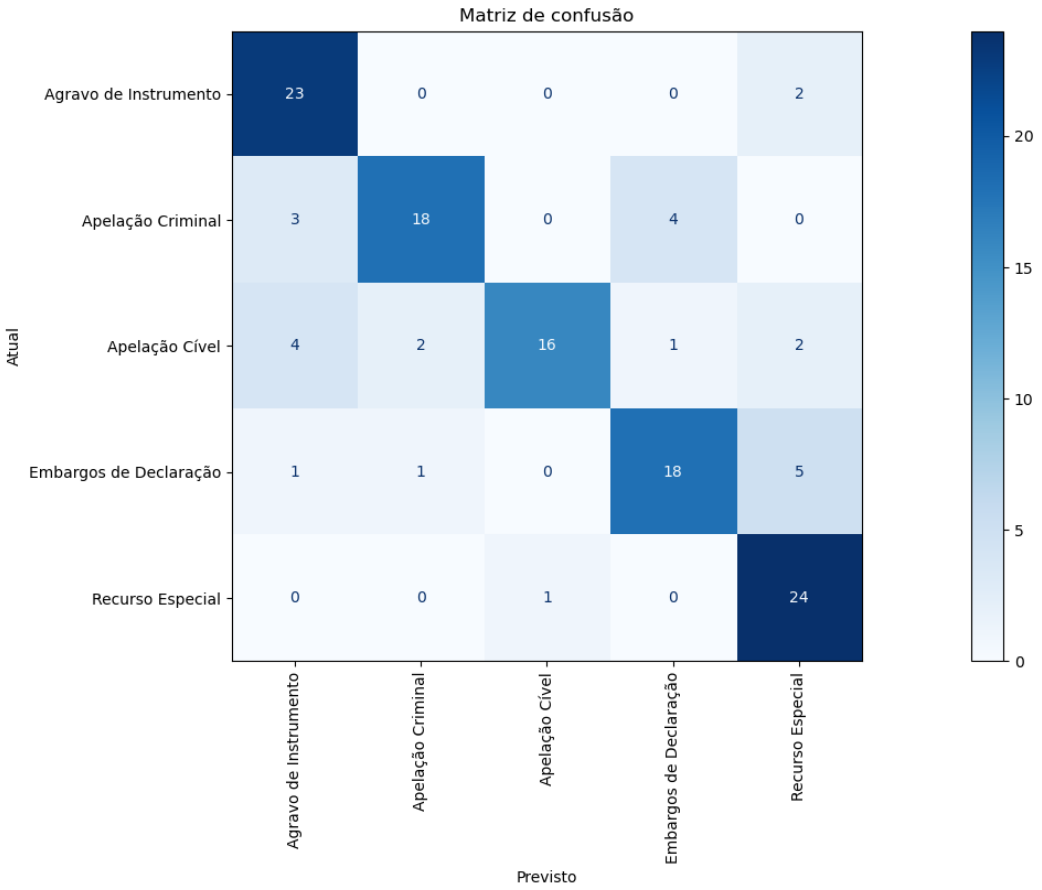

Figura 2. Matriz de Confusão da Floresta Aleatória
Fonte: Os autores

Conforme ilustrado na figura 3, os resultados obtidos utilizando as métricas selecionadas foram muito bons, especialmente devido ao conjunto de treinamento selecionado.

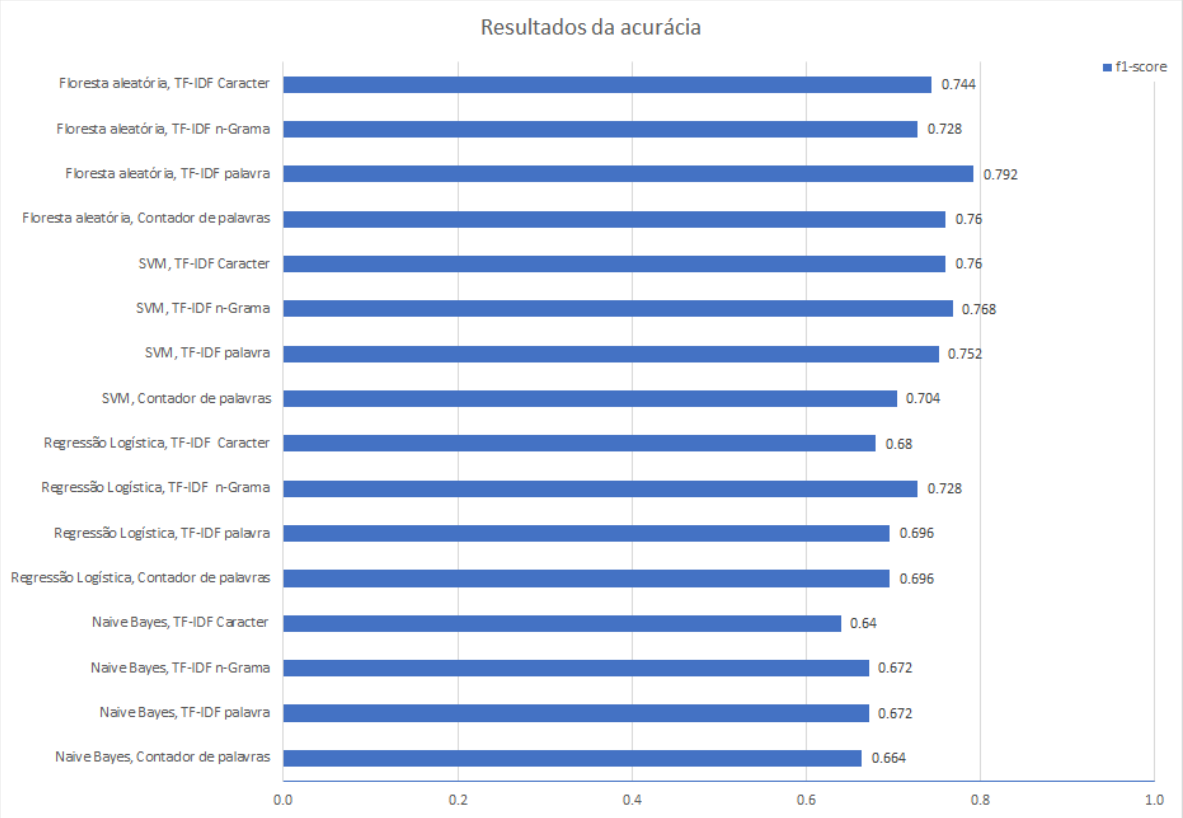

Figura 3. Resultado da acurácia de todos os métodos
Fonte: Os autores

Observa-se que poucos erros ocorreram no processo de classificação, sendo considerado um resultado promissor. A seção 5.1, mostra de uma forma mais detalhada os demais classificadores analisados. Abaixo, na figura 4, é possível verificar o relatório de execução do algoritmo com mais detalhes.

```
Training:
RF, WordLevel TF-IDF
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                      n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
train time: 0.459s
test time:  0.024s
accuracy:   0.792
f1_score_macro:   0.789
f1_score_micro:   0.792

classification report:
                       precision    recall  f1-score   support

 Agravo de Instrumento       0.74      0.92      0.82        25
     Apelação Criminal       0.86      0.72      0.78        25
        Apelação Cível       0.94      0.64      0.76        25
Embargos de Declaração       0.78      0.72      0.75        25
      Recurso Especial       0.73      0.96      0.83        25

              accuracy                           0.79       125
             macro avg       0.81      0.79      0.79       125
          weighted avg       0.81      0.79      0.79       125
```

Figura 4. Relatório de execução do algoritmo
Fonte: Os autores

O SVM não pode deixar de ser citado, dado que obteve um bom f_1-*score* de 76,8%. No entanto, seu tempo para treinamento foi o dobro da Floresta Aleatória. Enquanto o SVM teve um tempo de treinamento de 0.921s e teste de 0.199s, a Floresta Aleatória teve um tempo de treinamento de 0.459s e teste de 0.024s. Para esse tipo de cenário, o tempo de execução é um fator importante e que deve ser levado em consideração, imagine se o tempo de execução da Floresta Aleatória for 10 dias, e o SVM for 20 dias, é um valor que preocupa.

No gráfico abaixo, ilustrado pela figura 5, é possível visualizar que o método Naive Bayes teve o menor tempo de treinamento de 0.007s e teste de 0.003s, porém a pontuação máxima que o método obteve foi de 67.2% ficando abaixo dos 79.2% da Floresta Aleatória. A Regressão Logística teve o pior resultado dentre os quatro classificadores, com um *score* baixo, de 72.8%, e um tempo de execução relativamente alto, de 0.162s. Observando o SVM nota-se que esse método tem o maior tempo de treinamento e de teste, sendo a média dos tempos de 0.960s para treinamento e 0.206s de teste, alcançou um score melhor que o Naive Bayes; mas o custo de processamento torna-se inadequado para um conjunto de dados muito volumoso. Por fim, como citado no início, a Floresta Aleatória, apesar do seu custo de treinamento teve o melhor desempenho neste estudo.

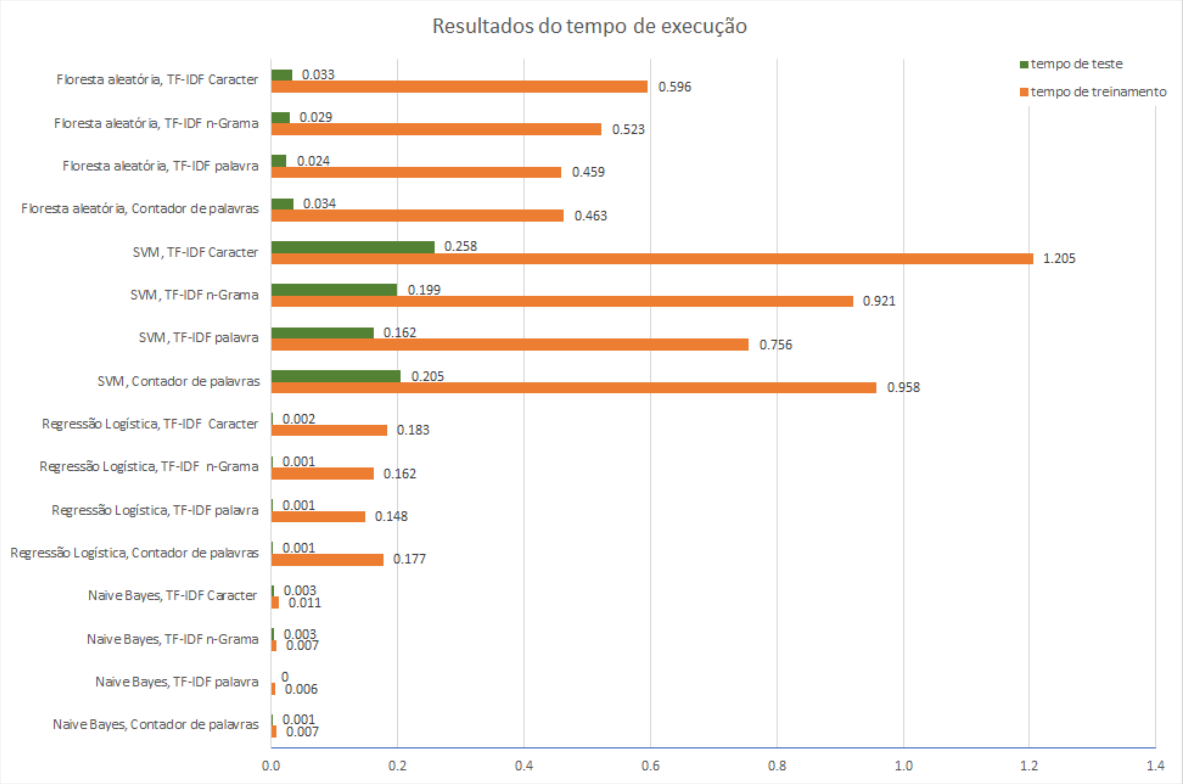

Figura 5. Resultado dos tempos de execução de todos os métodos
Fonte: Os autores

5.1 Outros resultados

Além do melhor resultado da Floresta Aleatória, nesta seção são mostrados os valores registrados por cada classificador com seus respectivos vetores de contagem.

5.1.1 Floresta aleatória

Os parâmetros utilizados para o classificador Floresta Aleatória foram:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
```

```
min_weight_fraction_leaf=0.0, n_estimators=100,
n_jobs=None, oob_score=False, random_state=0, verbose=0,
warm_start=False)
```

Tabela 1. Resultados obtidos da Floresta Aleatória com os demais vetores de contagem

Vetor de Contagem	Tempo de treino	Tempo de teste	Acurácia	f_1-macro	f_1-micro
Contador de palavras	0.463s	0.034s	0.760	0.756	0.760
Caracter	0.596s	0.033s	0.744	0.738	0.744
n-Grama	0.523s	0.029s	0.728	0.724	0.728

5.1.2 SVM

Para o classificador SVM, os parâmetros configurados foram: `SVC(C=100, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,`

```
decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
max_iter=-1, probability=False, random_state=42, shrinking=True, tol=0.001,
verbose=False)
```

Tabela 2. Resultados do SVM obtidos

Vetor de Contagem	Tempo de treino	Tempo de teste	Acurácia	f_1-macro	f_1-micro
Contador de palavras	0.958s	0.205s	0.704	0.699	0.704
TF-IDF Caracter	1.205s	0.258s	0.760	0.758	0.760
TF-IDF n-Grama	0.921s	0.199s	0.768	0.766	0.768
TF-IDF palavra	0.756s	0.162s	0.752	0.749	0.752

5.1.3 Regressão Logística

Utilizamos para a Regressão Logística a seguinte configuração: `LogisticRegression(C=100, class_weight=None, dual=False, fit_intercept=True,`

```
intercept_scaling=1, l1_ratio=None, max_iter=10000,
multi_class='auto', n_jobs=None, penalty='l1',
random_state=None, solver='liblinear', tol=0.0001, verbose=0,
warm_start=False)
```

Tabela 3. Resultados obtidos com a Regressão Logística

Vetor de Contagem	Tempo de treino	Tempo de teste	Acurácia	f_1-macro	f_1-micro
Contador de palavras	0.177s	0.001s	0.696	0.691	0.696
TF-IDF Caracter	0.183s	0.002s	0.680	0.677	0.680
TF-IDF n-Grama	0.162s	0.001s	0.728	0.726	0.728
TF-IDF palavra	0.148s	0.001s	0.696	0.692	0.696

5.1.4 Naive Bayes

No Naive Bayes, foram usados os parâmetros a seguir: `MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)`

Tabela 4. Resultados do Naive Bayes

Vetor de Contagem	Tempo de treino	Tempo de teste	Acurácia	f_1-macro	f_1-micro
Contador de palavras	0.007s	0.001s	0.664	0.657	0.664
TF-IDF Caracter	0.011s	0.003s	0.640	0.631	0.640
TF-IDF n-Grama	0.007s	0.003s	0.672	0.665	0.672
TF-IDF palavra	0.006s	0.000s	0.672	0.666	0.672

6. Conclusão

Com a crescente quantidade de informações digitais no mundo, pode-se observar a dificuldade para realizar uma classificação manual de todas essas informações.

Desse modo, a análise proposta neste estudo mostra que os classificadores e a aprendizagem de máquina podem ser muito úteis para essa tarefa. Através da utilização da técnica aprendizado de máquina supervisionado é possível apresentar informações relevantes sem a necessidade de ler manualmente todos os documentos.

Os resultados apresentados diferem dos trabalhos relacionados demonstrados na seção 2, mostrando que, apesar de serem todos documentos jurídicos, a forma de aplicar as técnicas muda de acordo com o objetivo. Destaca-se a variedade de classificadores testados e a maior quantidade de classes quando comparado com os projetos citados.

Além de uma boa precisão na classificação, devido a quantidade de classes existentes, outro fator primordial apresentado, foi o tempo computacional utilizado para o treinamento de cada modelo na aplicação prática desses algoritmos, visto que se o tempo for muito alto, pode tornar inviável sua utilização. Pode-se concluir também que os classificadores apresentados obtiveram um bom resultado para o conjunto avaliado. Ao serem aplicados em outros cenários e com diferentes conjuntos podem se fazer necessários processos diferenciados na execução e aplicação dos mesmos.

É necessário levar em consideração que um aumento do tamanho da amostra não significa que haverá uma influência positiva no resultado, pois aumentando o número de documentos e de suas classes pode-se acabar reduzindo a efetividade dos algoritmos apresentados, necessitando de uma nova parametrização para se atingir o objetivo.

Como trabalho futuro, pretende-se utilizar o resultado da classificação e integrar a um sistema de pesquisa inteligente baseado no conteúdo extraído de cada documento. Melhorar o pré-processamento dos documentos para aumentar a acurácia da classificação e explorar outros métodos de classificação que utilizam redes neurais.

Referências

ARAÚJO NETO, Alfredo Silveira. Utilização de técnicas de mineração de texto para organização não supervisionada de atos processuais digitais. 2015. 178 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Estadual do Ceará, Fortaleza, 2015.

BARROS, Rhuan. Análise jurisprudencial com técnica de aprendizado de máquina: qual a tendência de opinião de cada turma do TRT da 3ª Região. 2017. Disponível em: <https://rhuanlopesbarros.jusbrasil.com.br/artigos/529722364/analise-jurisprudencial-com-tecnica-de-aprendizado-de-maquina>. Acesso em: 06 nov. 2019.

DIMOULIS, Dimitri. Manual de introdução ao estudo do direito. 5. ed. São Paulo: Revista dos Tribunais, 2013.

ESTADO, Agência. Processos provocam rachadura em prédio de fórum em SP. 2009. Disponível em: <http://g1.globo.com/Noticias/SaoPaulo/0,,MUL1040240-5605,00-PROCESSOS+PROVOCAM+RACHADURA+EM+PREDIO+DE+FORUM+EM+SP.html>. Acesso em: 06 nov. 2019.

GOMES, Neide de Oliveira. Categorização de Textos - Estudo de Caso: documentos de pedidos de patente no idioma português. 2013. 294 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2013.

JOACHIMS Thorsten. (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg.

MCCALLUM, Andrew, NIGAM, Kamal. A comparison of event models for naive bayes text classification. AAAI 1998 (1998).

MÜLLER, Andreas C.; GUIDO, Sarah. Introduction to Machine Learning with Python: a guide for data scientists. Sebastopol: O'reilly, 2016.

NADER, Paulo. Introdução ao estudo do direito. 21. ed. Rio de Janeiro: Forense, 2001.

OLIVEIRA, Cristiano Cesar da Silva. Categorização automática de documentos jurídicos utilizando o classificador naive-bayes. 2015. 42 f. TCC (Graduação) - Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas, Instituto Federal de Educação, Ciência e Tecnologia - Ifsp, Campos do Jordão, 2015.

SANTOS, Fernando. Mineração de opinião em textos opinativos utilizando algoritmos de classificação. 2013. 60 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2013.

TAN, Pang-ning; STEINBACH, Michael; KUMAR, Vipin. Introdução ao Data Mining: mineração de dados. Rio de Janeiro: Ciência Moderna, 2009.

TICOM, Antonio; MELLO Alexandre. Aplicação de Mineração de Textos e Sistemas Especialistas na Liquidação de Processos Trabalhistas Especialistas. 2007. 101 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

TSE. Perguntas frequentes. Disponível em: <http://www.tse.jus.br/jurisprudencia/perguntas-frequentes>. Acesso em: 06 nov. 2019.

ZHANG, Tong; OLES, Frank.J. Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4, 5–31 (2001). https://doi.org/10.1023/A:1011441423217