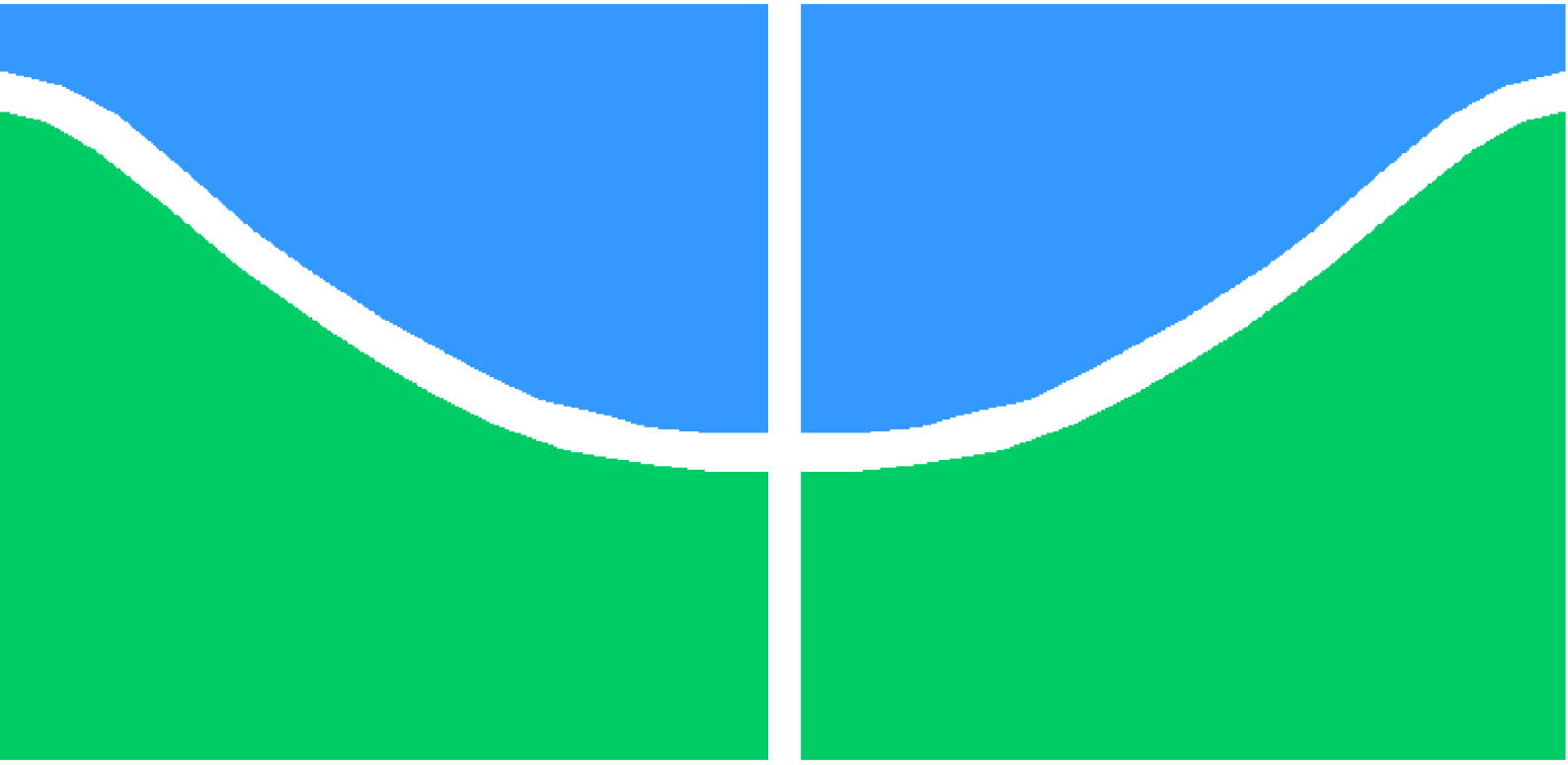

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Utilizando Text Mining na Taxonomia Processual

Autor: Gustavo Rodrigues Coelho
Orientador: Professor Dr. Ricardo Matos Chaim

Brasília, DF
2018

Gustavo Rodrigues Coelho

Utilizando Text Mining na Taxonomia Processual

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Professor Dr. Ricardo Matos Chaim

Brasília, DF

2018

Gustavo Rodrigues Coelho
Utilizando Text Mining na Taxonomia Processual/ Gustavo Rodrigues Coelho. – Brasília, DF, 2018-
42 p. : il. (algumas color.) ; 30 cm.

Orientador: Professor Dr. Ricardo Matos Chaim

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2018.
Utilizando Text Mining na Taxonomia Processual

CDU 02:141:005.6

Gustavo Rodrigues Coelho

Utilizando Text Mining na Taxonomia Processual

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 01 de Julho de 2018:

Professor Dr. Ricardo Matos Chaim
Orientadora

Convidado 1

Convidado 2

Brasília, DF
2018

Dedicamos este trabalho às pessoas que sempre estiveram ao nosso lado incentivando e nos ajudando nos momentos mais difíceis desta jornada.

Agradecimentos

Meus sinceros agradecimentos a todos que me ajudaram de alguma forma nessa jornada. Primeiramente a minha avó materna que me ensinou nos primeiros anos de vida escolar. Os mais importantes para qualquer criança. Em segundo, aos meus pais que sempre me apoiaram na jornada de sair de casa para estudar. Ao meu irmão Gabriel que por tantas vezes me suportou e nunca desistiu de mim. E mais recentemente a Sara que tem feito parte da minha vida.

Agradeço, também, aos vários amigos que conheci nessa caminhada: no ciências sem fronteiras, na universidade e na escola.

A todos o meu muito obrigado.

"Significa simplesmente que, em cada momento da história,

as pessoas têm as suas sensibilidades."

(Fernando Henrique Cardoso)

Resumo

O trabalho apresenta uma avaliação de métodos supervisionados de classificação utilizando como entrada processos judiciais. O sistema judicial brasileiro recebe milhões de processos por ano, que possuem uma variedade de informações que interessam diversos atores inclusive o poder executivo. A tarefa de classificar esses processos para análise em pesquisas específicas é uma tarefa hercúlea sendo uma grande oportunidade para o uso de algoritmos inteligentes. O trabalho utilizou os algoritmos knn e naive bayses para classificar os processos judiciais e avaliou a performance dos dois algoritmos. O trabalho resultou em valores adequados para os dois algoritmos podendo os dois serem usados para a classificação de processos judiciais.

Palavras-chaves: Classificação;Text Mining;IPEA;Processos Judiciais;

Abstract

The paper presents an evaluation of supervised learning methods of classification using as input lawsuits. The Brazilian judicial system receives millions of cases per year, which have a variety of information that interests research papaer and stakeholders including the executive power. The task of classifying these processes for analysis in specific searches is a Herculean task being a great opportunity for the use of intelligent algorithms. The work utilized the knn and naive bayses algorithms to classify the judicial processes and evaluated the performance of the two algorithms. The work resulted in adequate values for the two algorithms, both of which can be used to classify lawsuits. **Key-words**: Classification;Text Mining;IPEA;Judicial Processes;

Lista de ilustrações

Figura 1 – O processo principal da abordagem GQM 15
Figura 2 – Exemplo de categorização pelo KNN 24
Figura 3 – Processo do experimento 30
Figura 4 – Erro KNN para K para 3000 amostras 35
Figura 5 – Erro KNN para K com 3000 amostras 36
Figura 6 – Matriz de confusão Naive Bayses para 2000 amostras 36
Figura 7 – Curva ROC para 2000 amostras 37
Figura 8 – Erro KNN para K com 3000 amostras 37
Figura 9 – Matriz de confusão KNN para 3000 amostras 38
Figura 10 – Matriz de confusão Naive para 3000 amostras 38
Figura 11 – Erro KNN para K de 0 a 20 39

Lista de tabelas

Tabela 1 – Matriz de confusão 28
Tabela 2 – Primeiro objetivo 29
Tabela 3 – Segundo objetivo 29
Tabela 4 – Cronograma do desenvolvimento do trabalho 32
Tabela 5 – Classe de processo encontradas 34

Lista de abreviaturas e siglas

IPEA	Instituto de Pesquisa Econômica Aplicada
CNJ	Conselho Nacional de Justiça
KNN	K Nearest Neighbor

Sumário

1 INTRODUÇÃO 13
1.1 Objetivos 14
1.1.1 Objetivo Geral 14
1.1.2 Objetivos Específicos 14
1.2 Metodologia de Pesquisa 15
1.3 Organização do Trabalho 16

2 REFERENCIAL TEÓRICO 17
2.1 O sistema judiciário brasileiro 17
2.2 Taxonomia Processual 19
2.3 Text mining 20
2.3.1 Bag of words 20
2.3.2 Stop Words 21
2.3.3 Stemming 21
2.3.4 Lemmatization 22
2.3.5 Term Frequency(TF) 22
2.4 Aprendizado de máquina 23
2.4.1 Knearest Neighbor Classification (KNN) 24
2.4.2 Naive Bayes 26
2.4.3 Avaliação 27

3 METODOLOGIA 29
3.1 GQM 29
3.2 Processo 30
3.3 Cronograma 31

4 RESULTADOS 33
4.1 Coleta de dados 33
4.2 Resultados 33
4.2.1 Resultados para 2000 amostras 35
4.2.2 Resultados para 3000 amostras 37

5 CONCLUSÃO 40

REFERÊNCIAS 41

1 Introdução

O Instituto de Pesquisa econômica Aplicada (IPEA) é uma fundação pública vinculada ao ministério do planejamento, desenvolvimento e gestão que tem por missão desenvolver pesquisas sobre a realidade brasileira para fomentar o debate e as políticas públicas do governo federal. O órgão analisa, coleta e cruza dados dos três poderes gerando trabalhos que subsidiam políticas públicas do poder executivo.

Os efeitos das decisões dos poderes legislativo e judiciário no poder executivo podem ser sentidos de diversas maneiras como no aumento de gastos públicos ou na reformulação de programas para se adequar a novas regras. As decisões judiciais obrigando o poder executivo a conceder remédios de alto custo são um exemplo de impacto direto no planejamento financeiro de curto prazo dos entes federados. O estudo e monitoramento de dados dos outros poderes são demandados ao IPEA para avaliar impactos em políticas públicas. O poder judicial tem um papel importante nessas demandas devido ao grande número de processos existentes, - já atinge uma marca recorde de 80 milhões de processos em estoque de acordo com o relatório Justiça em números do Conselho Nacional de Justiça (CNJ) de 2017 ano base 2016 - e não sabendo a administração pública quantos podem gerar efeitos direto e indiretos em suas atividades.

A taxonomia processual é feita de forma subjetiva pelos servidores dos tribunais baseada nas classes unificadas de processos do CNJ. Esse método não abrange temas mais específicos como casos para uma determinada lei, além de não permitir a separação por outros elementos do contexto do processo por exemplo, não seria possível encontrar os condenados por furto dentro de uma faixa etária. Uma informação como essa poderia ser valiosa para o planejamento de políticas públicas voltadas para a redução dos furtos para uma determinada área geográfica. A classificação processual de forma computacional é uma forma de melhorar as pesquisas na área judicial brasileira gerando valor para o IPEA e a administração pública.

Na computação segundo (HAN JIAWEI; KAMBER; PEI, 2012), a descoberta de conhecimento através da exploração de uma base de dados pode ser definida como mineração de dados. Os autores ainda sugerem uma analogia ao processo físico de separar minerais em uma compania de mineração, onde de uma porção bruta de minérios é possivel encontrar pepitas de ouro. Seguindo a definição dos autores, um bom exemplo desse processo seria uma grande empresa de supermercados pode descobrir que clientes acima de 40 anos comem menos açúcares do que um cliente de 25 cruzando suas vendas com a idade dos seus clientes. A fonte a ser analisada pode estar em diferentes formatos: estruturados, como no exemplo, ou não estruturados, como no caso de processos judiciais.

O fato de não estar estruturados exige uma abordagem diferente para o tratamento.

Os paradigmas de processamento de dados não estruturados em linguagem natural podem ser divididos em dois grupos: linguistico ou estatístico (IJSMI, 2017). No primeiro caso a posição e classe gramatical das palavras tem valor e alteram os resultados, na segunda forma as análises são feitas considerando a ocorrência da palavra em um dado texto. Os dois casos podem utilizar algoritmos de aprendizagem de máquina para classificar ou predizer algum tipo de conhecimento sobre o texto. As técnicas para a análise textual serão utilizadas para a classificação dos processos judiciais. A verificação da eficácia dessas técnicas no caso judicial será o foco desse trabalho para gerar um aprimoramento nos trabalhos do IPEA em relação a processos judiciais.

1.1 Objetivos

1.1.1 Objetivo Geral

O objeto deste trabalho é verificar a eficácia de algoritmos de classificação para o caso de processos judiciais.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, foram definidos os seguintes objetivos específicos:

- Realizar uma Revisão da literatura acerca dos métodos de análise de textos (text mining);
- Realizar uma Revisão da literatura a cerca dos métodos de aprendizagem de máquina;
- Selecionar algoritmos para classificação;
- Adquirir os dados (processos judiciais) dos diversos entes produtores do sistema judicial;
- Pré-processamento dos dados;
- Aplicação de métodos de aprendizagem
- Análise dos resultados;
- Elaborar a proposta de solução.

1.2 Metodologia de Pesquisa

O conhecimento científico difere-se do conhecimento popular devido a utilizar métodos reproduzíveis tornando-o verificável. Os métodos gerais ou de abordagem oferecem ao pesquisador normas genéricas destinadas a esclarecer uma ruptura entre objetivos científicos e não científicos. (PRODANOV, 2013). Os métodos definem procedimentos lógicos a serem seguidos para o estudo dos fatos da natureza e da sociedade. No conjunto de métodos de amplo conhecimento pode-se destacar os métodos de procedimento exemplificados pelo método histórico, experimental, observacional, comparativo, estatístico, o clinico e o monográfico.

O método experimental consiste, especialmente, em submeter os objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, para observar os resultados que a variável produz no objeto(GIL, 2008). Supõe-se que a abordagem mais apropriada para a experimentação na área de Engenharia de Software seja o método experimental que considera a proposição e avaliação do modelo com os estudos experimentais (TRAVASSOS G. H.; GUROV; AMARAL, 2002). Com o intuito de se atingir os objetivos de um experimento de forma metodológica algumas abordagem foram elaboradas. Uma dessas abordagens é conhecida como GQM(GOAL/Question/Metric). A definição do GQM pode ser vista na Figura 1.2.

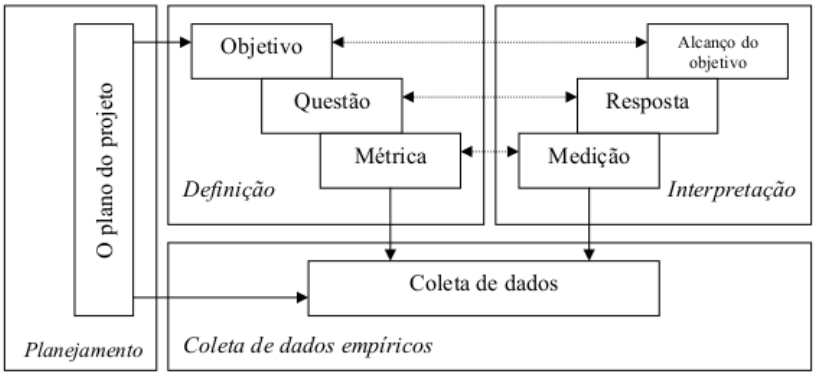

Figura 1 – O processo principal da abordagem GQM

A abordagem oferece o processo da melhoria com o modelo da medição baseado em camadas. A definição e a interpretação do processo da experimentação são divididas em camadas conceitual, operacional, e quantitativa(TRAVASSOS G. H.; GUROV; AMARAL, 2002). O GQM foi portanto a metodologia escolhida para a execução do experimento descrito neste trabalho de conclusão de curso

1.3 Organização do Trabalho

O trabalho será organizado em 6 capítulos seguindo a seguinte estrutura e tópicos:

- ***Capítulo 2:*** Apresenta como o sistema judiciário brasileiros está organizado. Além de apresentar como a taxonomia processual é feita nos tribunais. O capítulo, ainda, descreve técnicas de análise de texto além de algoritmos de classificação.
- ***Capítulo 3:*** Dedicado a apresentar o método utilizado para se adquirir, analisar e separar os processos judiciais e como será executado o experimento no próximo semestre.
- ***Capítulo 4:*** O último capítulo se dedicará a apresentar os resultados alcançados por essa primeira parte do trabalho.

2 Referencial Teórico

Este Capítulo apresenta conceitos importantes a respeito de extração de textos e aprendizado de máquina que subsidiarão o desenvolvimento do experimento planejado.

2.1 O sistema judiciário brasileiro

O Poder Judiciário, assim como o Executivo e o Legislativo são partes da União, independentes e harmônicos entre si, regidos pela Constituição Federal, consolidados na forma de cláusula pétrea, de acordo com o artigo 2º da CF. Cada poder possui funções típicas e atípicas. A violação a essas funções pré-determinadas constitui critério para definir quando há ou não violação ao princípio da separação dos poderes.

No âmbito do poder judiciário, tem-se claramente que função típica é puramente jurisdicional. O poder judiciário tem a função principal de dizer o direito, aplicar a lei ao caso concreto e avaliar a lei. De forma atípica e subsidiária, o judiciário também legisla, na medida em que elabora seus próprios regimentos, assim como também possui função executiva ao se auto organizar e autogerir. Muito embora haja expressa separação entre os Poderes, é importante ressaltar que com a modernização da doutrina, hoje já se fala em colaboração entre os Poderes, baseada no mútuo controle definido pela teoria "Checks and balances". O sistema judiciário brasileiro é dividido de forma funcional, material e territorial, nos termos da Constituição Federal, em seus artigos 92 à 126. Em sua composição estão os seguintes órgãos:

- Supremo Tribunal Federal,
- Conselho Nacional de Justiça,
- Superior Tribunal de Justiça,
- Tribunal Superior do Trabalho,
- Tribunais Regionais Federais e Juízes Federais,
- Tribunais e Juízes do Trabalho,
- Tribunais e Juízes Eleitorais,
- Tribunais e Juízes Militares,
- Tribunais e Juízes dos Estados e do Distrito Federal e Territórios.

Cada órgão possui um âmbito de jurisdição, seja definido pelo local do conflito, pelas partes envolvidas ou pela matéria em discussão, de forma que o sistema funcione, em regra, sem choques de competência. Todavia, uma vez instaurado um conflito de competências entre os órgãos que compõem o judiciário, deve-se adotar os procedimentos previstos nos Regimentos Internos de cada Tribunal, amparados pela Constituição.

As competências de cada um dos ramos do poder judiciário pode ser assim definida:

Supremo Tribunal Federal

Sua principal competência é julgar ações de inconstitucionalidades - declarar se leis infra constitucionais estão de acorda com a carta magna. O órgão ainda é responsável por arbitrar as contendas entre a união e os estado. Além disso, o supremo federal é responsável por julgar nas infrações penais comuns, o Presidente da República, o Vice-Presidente, os membros do Congresso Nacional, seus próprios Ministros e o Procurador-Geral da República.

Superior Tribunal de Justiça

é o guardião da uniformidade da interpretação das leis federais. Desempenha esta tarefa ao julgar as causas, decididas pelos Tribunais Regionais Federais ou pelos Tribunais dos estados, do Distrito Federal e dos territórios, que contrariem lei federal ou deem a lei federal interpretação divergente da que lhe haja atribuído outro Tribunal.

Tribunais Regionais Federais e Juízes Federais

A Justiça Federal julga, dentre outras, as causas em que forem parte a União, autarquia ou empresa pública federal. Dentre outros assuntos de sua competência, os TRFs decidem em grau de recurso as causas apreciadas em primeira instância pelos Juízes Federais

Tribunal Superior do Trabalho e Tribunais e Juízes do Trabalho

A partir de 2004 as competências da justiça do trabalho foram alteradas passando a processar e julgar toda e qualquer causa decorrente das relações de trabalho, o que inclui os litígios envolvendo os sindicatos de trabalhadores, sindicatos de empregadores, análise das penalidades administrativas impostas pelos órgãos do governo incumbidos da fiscalização do trabalho e direito de greve

Tribunais e Juízes Eleitorais

Compete-lhe julgar as causas relativas à legislação eleitoral. O TSE, dentre outras atribuições, zela pela uniformidade das decisões da Justiça Eleitoral. Além disso, a justiça eleitoral cumpre um papel administrativo, de organização e normatização das eleições no Brasil.

Tribunais e Juízes Militares

A justiça militar cabe julgar e processar de acordo com o código penal militar os servidores federais e estaduais que se enquadram nessa categoria de serviços. Além de, diferente dos outros ramos da justiça, ser competente para auditar os serviços militares em sua jurisdição zelando pelo seu devido cumprimento.

Tribunais e Juízes dos Estados e do Distrito Federal e Territórios.

Os Tribunais de Justiça dos estados possuem competências definidas na Constituição Federal, na Constituição Estadual, bem como na Lei de Organização Judiciária do Estado. Basicamente, o TJ tem a competência de, em segundo grau, revisar as decisões dos juízes e, em primeiro grau, julgar determinadas ações em face de determinadas pessoas.

2.2 Taxonomia Processual

A separação de coisas comuns em grupos que compartilham características semelhantes é uma tarefa empreendida pelo ser humano desde de tempos primordiais. Aristóteles separou e classificou diversas plantas gregas gerando um conhecimento sistematizado sobre fauna e flora. A arte - iniciada de forma sistematizada por Aristóteles - de distinguir grupos é chamada de taxonomia. As suas técnicas e métodos, primeiramente aplicados a animais e plantas, foram expandidas para outras áreas das ciências como o direito. A taxonomia processual tem como meta classificar os processos para a criação de estatísticas e dados do sistema judiciário.

No Brasil a taxonomia processual era responsabilidade dos sistemas judiciais em suas diversas instâncias. Assim, cada tribunal poderia ter o seu próprio sistema de classificação e nomenclatura. Em 2004 com a criação do Conselho Nacional de Justiça pela (BRASIL, 2004) foi iniciada uma sistematização das classificações processuais com fins estatísticos.

A resolução 46 de 18 de dezembro de 2007, instituiu tabelas unificadas para a classificação de processos por todos os órgãos do poder judiciário. Nas considerações da resolução é citado: "a ausência de padrão mínimo para cadastro de partes entre os órgãos do Poder Judiciário"como justificativa para a implementação de uma tabela comum a todos os participantes do sistema judicial.

As tabelas unificadas descrevem nove classe genéricas para a classificação dos processos no pais são elas:

- Juizados da infância e da juventude
- Procedimentos Administrativos

- Procedimentos pré-processuais de resolução consensual de conflitos
- Processo Cível e do Trabalho
- Processo Criminal
- Processo Eleitoral
- Processo Militar
- superior Tribunal de Justiça
- Superior Tribunal Federal

Cada classe processual acima possue subclasses e por sua vez essas subclasses. Os procedimentos para o recebimento e classificação desses processos devem seguir orientação dada pelo CNJ.

2.3 Text mining

O processamento e análise de texto adquire o nome de Text mining na computação. Um conjunto de ferramentas foram desenvolvidas para a transformação de informações não estruturadas, o caso textual, para estruturadas podendo assim serem feitas análises utilizando algoritmos de aprendizagem de máquina ou métodos estatísticos para a extração de padrões do texto. As técnicas abaixo foram retiradas da literatura para embasarem a metodologia a ser aplicada ao problema identificado neste trabalho.

As técnicas descritas a seguir podem ser chamadas de técnicas de pré-processamento pois antecedente a etapa de treinamento dos algoritmos de aprendizagem. O seu objetivo é a transformação de textos em dados de entradas para a fase posterior. Algumas técnicas impactam a performance dos algoritmos de aprendizagem por excluir dados irrelevantes para o treinamento. Um exemplo a ser citado é a remoção de palavras muito comuns nas linguagens como artigos e preposições. Os próximos tópicos descreverão com detalhes essas técnicas.

2.3.1 Bag of words

A resolução de cálculos matemáticos foram e são o principal problema a ser solucionado pela ciência da computação. Porém os textos também ganharam visibilidade assim que as condições de hardware melhoraram. (HARRIS, 1954), em seu artigo de 1954, arguiu que os textos teriam uma estrutura interna que poderia ser utilizada para entender o significado do texto desconsiderando as estruturas sintáticas.

(HARRIS, 1954) analisa o comportamento estrutural do texto para as categorias das estruturas de acordo com as regras sintáticas e desconsiderando essas estruturas. As conclusões do autor permitiram assumir que a estrutura sintática não afetaria o resultado estrutural do texto podendo assim ser considerada a ocorrência dos termos como indicador do significado deste. A partir desses achados do autor outras formas de se contabilizar a frequência dos termos em um dado texto foram desenvolvidas com o intuito de melhorar os achados do autor.

2.3.2 Stop Words

Ao analisar a frequência das palavras nos textos (HARRIS, 1954) notou que certas classes de palavras se repetem com frequência em todos os textos. O conjunto dessas palavras foi chamado de stop words.

As stop words são definidas a partir da análise de diversos textos e de sua frequência nos mesmos. A partir disso são construídos dicionários que permitem auxiliar na tarefas de mineração de textos. Os dicionários são feitos para cada linguagem, no inglês palavras como: the, a, an fazem parte do dicionário de stop words enquanto na língua portuguesa serão encontradas palavras como do, das, o, a.

O dicionário de stop words, portanto, permite remover palavras dos textos que serão irrelevantes estruturalmente ou para processos de classificação textual evitando processamentos adicionais e tornando os modelos mais precisos.

2.3.3 Stemming

A análise textual através do algorítimo bag of words levanta alguns pontos de análise na contagem das palavras, ocorrências como levanto e levantamos são conjugações do verbo levantar na primeira e terceira pessoa. As duas palavras podem ser contadas como 2 ocorrências da palavra levantar permitindo uma melhor análise da frequência no texto.

As técnicas para análise de texto não passaram despercebidas as diferentes formas de ocorrências das palavras em um texto, portanto, foram desenvolvidas duas formas de transformar as palavras reduzindo a sua ocorrência para uma uma forma padrão: stemming e lemmatization. Stemming é um algoritmo desenvolvido por (LOVINS, 1968) que consiste na redução da palavra para sua forma raiz.

A forma raiz de uma palavra é alcançada após a retirada dos sufixos adicionado a ela. Na lingua portuguesa para a formação do plural é necessário adicionar, em muitos casos, a letra s ao final dos substantivos tornando gato em gatos, cachorro em cachorros. O algoritmo de stemming leva em consideração essas regras de sintase para reduzir as

palavras as sua raiz. No exemplo dado, uma regra seria a retirada de s de todo o fim de palavras.

A aplicação de diversas regras as palavras do texto pode gerar resultados que não existem propriamente, devido a não possibilidade de generalização na linguagem. Uma eliminação de s ao final de todos os substantivos retiraria o s do final da palavra ônibus produzindo a palavra ônibu que não existe na língua portuguesa.

O algoritmo de stemming foi otimizado por (PORTER, 1980) ao aplicar um conjunto pequeno de regras para retirar sufixos de palavras em inglês. Um conjunto maior de regras é capaz de gerar resultados melhores devido a grande quantidade de regras das linguagens porém haverá uma queda no desempenho do processamento como (PORTER, 1980) aponta em seu artigo.

As línguas europeias como português, espanhol e alemão podem usar os mesmos princípios do algorítimo de (PORTER, 1980) para gerar formas raízes das palavras. (SAVOY, 2006) descreve uma abordagem para a quantidade de regras que devem ser utilizadas para a língua portuguesa dado que essa utiliza o gênero nas conjugações diferentemente do inglês.

2.3.4 Lemmatization

O tempo de processamento nas décadas finais do século passado era um requisito muito importante para os algoritmos devido a disponibilidade de hardware. A melhoria da parte física do computador permitiu a criacão ou implementação de algoritmos mais complexos. A lematizacão é um desses casos, obstante reduzir a palavra a sua raiz como o stemming a lemmatizacão leva em consideração a posição da palavra no texto.

Assim como o stemming, a lemmatização também possui o objetivo de extrair a raiz de palavras que possuem inflexões, entretanto, o stemming realiza um corte indiscriminado, cortando o fim, ou em alguns casos, o começo da palavra, já a lemmatização leva em consideração a análise morfológica da palavra.

2.3.5 Term Frequency(TF)

As técnicas apresentadas anteriormente tem o objetivo de tratar os termos dos documentos que estão sendo analisados. Após a redução das palavras é possível contar a quantidade de vezes que um dado termo aparece em um documento, chamado de termy frequency(TF). A análise de frequência é a forma mais simples de analisar o quanto um termo esta relacionado a um documento, porém, (MANNING; SCHUTZE, 2008) aponta que esse método não traria resultados relevantes para conjuntos de textos de terminada área. A palavra fluído apareceria diversas vezes em contextos de engenharia mais não seria relevante para análise nesses mesmos textos.

O problema levantando é resolvido através de formas de normalização da frequência dos termos. Uma forma de para conseguir esse efeito é atribuido um peso para cada documento df(document frequency) para termo(termy frequency). Há várias formas de se atribuir os pesos relativos aos documentos e os termos. Uma forma descrita por (MANNING; SCHUTZE, 2008) é a frequência inversa do documento, dada pela fórmula:

$$idf_t = \log \frac{N}{df_t} \tag{2.1}$$

Onde:

idf_t = frequência inversa dos documentos
df_t = número de documentos com termo
N = é o número total de documentos analisados

O uso da fórmula acima resulta em frequências altas para palavras pouco encontradas nos documentos sendo em análise e baixos resultados para termos muito frequentes. Essa forma de análise permite encontrar os termos que melhor se adequem aos textos em análise.

2.4 Aprendizado de máquina

O filósofo Platão enunciou bem o valor do aprendizado na frase: a alegria que se tem em pensar e aprender faz-nos pensar e aprender ainda mais. A necessidade de aprender foi transportada para o campo da ciência da computação criando uma nova área de estudo conhecida como aprendizagem de máquina. O objetivo dessa área de estudo é desenvolver técnicas e métodos para que as máquinas aprendam. O pesquisador (SAMUEL, 1959), um pioneiro da área, desenvolveu na IBM um programa capaz de jogar damas melhor do quem escreveu o programa aprendendo com os jogos anteriores.

De acordo com (NILSSON, 1996) uma máquina aprende quando consegue melhorar sua performance em executar uma determinada tarefa a partir da experiência. O primeiro ponto que a definição levanta é como se mensurar a performance da realização de uma tarefa, que é o ponto central da teoria de aprendizado de máquina. O segundo ponto importante dessa definição é a palavra experiência que indica experiências passadas. No aprendizado de máquina experiência pode ser entendido como os dados disponíveis sobre um determinado problema.

Os dados já conhecidos do problema serão utilizados para o treinamento dos algoritmos de aprendizagem. A fase de treinamento pode ser entendida como os ajustes dos parâmetros dos algoritmos para se adequar ao contexto. Após a fase de treinamento dos dados o método escolhido é testando contra um conjunto de dados que também se

conhece as saídas corretas. O resultado da fase de teste, então é utilizado para avaliar o método ou os métodos utilizados no problema.

O aprendizado de máquina pode ser divido em duas ramos de acordo com (NILSSON, 1996): o aprendizado supervisionado e não supervisionado. No primeiro caso os algoritmos usarão em sua fase de testes dados dos quais se conhecem os resultados esperados. No aprendizado não supervisionado os algoritmos não contaram com as saídas esperadas. Os algoritmos descritos nas próximas seções estão na categoria de supervisionados.

2.4.1 Knearest Neighbor Classification (KNN)

KNN é um algoritmo de classificação não paramétrico desenvolvido por (COVER; HART, 1967) na década de 60 do século passado. O algoritmo utiliza um número k de pontos próximos para determinar a categoria a ser atribuída a um novo ponto. O algoritmo pode utilizar qualquer medida de distância a fim de distinguir os k pontos mais próximos. A principal medida utilizada é a distância euclidiana dada pela fórmula:

$$Distanciaeuclidiana = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{2.2}$$

O algoritmo KNN calcula a distância entre o ponto que está sendo avaliado e a sua distância entre todos os pontos do dataset de treinamento. Em seguida, os k pontos com a menor distância são usados na determinação da classe do ponto avaliado.

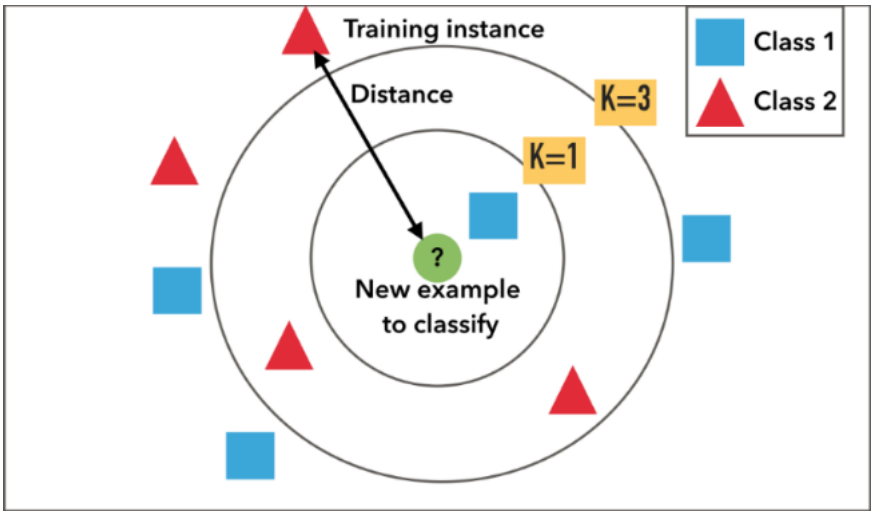

Figura 2 – Exemplo de categorização pelo KNN

A Figura 2.4.1, mostra os pontos mais próximos do ponto verde sendo avaliado. Assumindo k igual a um o ponto verde terá como vizinho mais próximo o quadrado azul e portanto sua classe será a classe 1. Aumentando o valor de k para 3, o ponto verde

passa a ter dois triângulos vermelhos e um quadrado azul como vizinhos mais próximos, passando então a pertencer à classe 2. O exemplo, ilustra o funcionamento do algoritmo e um parâmetro muito importante a ser analisado: o cálculo da distância entre os pontos.

A escolha da melhor medida de distância entre pontos abriu um novo campo de pesquisa chamada de aprendizado de distância. (Yang, Liu) trata do assunto em sua tese, dividindo o problema em dois campos. O aprendizado supervisionada e não supervisionado, os termos não guardam relação com o aprendizado de máquina. No aprendizado supervisionado tenta-se agrupar os pontos de mesma classe e separar os pontos de classes diferentes, enquanto, no aprendizado não supervisionada tenta-se reduzir as dimensões do dataset em análise.

Outro fator importante para a determinação da categoria: a regra adotada para se ponderar os k pontos mais próximos. No exemplo da Figura, para k = 3, foram encontrados 2 triângulos e 1 quadrado como vizinhos mais próximos. A regra de ponderação usada foi a maioria simples e o ponto verde atribuído a classe 2. No caso em que k = 1, não é necessária nenhuma regra pois a categoria será a mesma do ponto mais próximo. Para os casos k= 2, 3, 4 , n é necessário uma fórmula de ponderação entre os pontos.

O problema de ponderação entre os pontos é tratado em vários artigos e tenta melhorar os resultados do KNN. (SAMWORTH, 2012) descreve em seu artigo os efeitos de pesos no classificador e como derivar pesos que minimizem os erros. Outro método utilizado é chamado de Bootstrap aggregating desenvolvido por Breiman, que seleciona um subconjunto da amostra de teste original criando novos datasets de teste e aplicando um algoritmo de classificação nesse novo conjunto. O algoritmo de Breiman não melhorou a performance do KNN. As desvantagem do algoritmo são pontuadas por (YONG, 2009) para a classificação textual em três pontos:

- Grande complexidade de cálculos, devido ao grande número de palavras e documentos o cálculo da distância se torna complexo.
- Grande dependência do dataset de treinamento.
- Não diferenciação entres as amostras do dataset de treinamento.

Algumas soluções para os problemas apontados pelos autores são propostas de melhorias no algoritmo KNN. Os próprios autores desenvolvem técnicas de atribuir pesos diferentes para cada amostra do dataset de treinamento melhorando os resultados do KNN. O algoritmo tem uma alta potencialidade para a solução do problema devido a sua facilidade de uso, porém, provavelmente enfrentará problemas de grandes dimensionalidades devido ao grande número de palavras que podem ser encontradas. Para o escopo do trabalho o uso do KNN classico sem nenhuma modificação de métrica de distância ou de atribuição de pesos para o dataset de treino é a opção mais viável devido a complexidade.

2.4.2 Naive Bayes

O algoritmo naive bayses é baseado no teorema de Bayses. O matemático inglês desenvolveu os princípios do seu teorema em 1700, mas o método foi estendido pelo matemático francês Laplace anos após a morte de Lord Bayses. O teorema de Bayses dado pela fórmula:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.3}$$

O teorema permite uma simplificação dos cálculos ao assumir que todos os elementos são independentes em relação aos outros tornando Naive Bayes um classificador bastante eficiente computacionalmente e que alcança resultados compatíveis com métodos mais robustos. Os autores em discutem porque mesmo assumindo a independência entre as features do dataset ainda assim Naive Bayses consegue ser um método eficiente de predição.

Um classificador Naive Bayses irá utilizar os seguintes passos para indicar a qual classe o objeto sendo classificado pertence. Dado um objeto c com as características c = x1,x2 o objeto irá pertencer a classe como maior probabilidade S. Tomando como possíveis valores de S = S1 e S2 a probabilidade de c dara-se por:

$$\text{Probabilidade de s1 nos dados de teste} = \frac{\text{total de casos de s1}}{\text{total de casos na amostra amostra}} \tag{2.4}$$

$$\text{Probabilidade de s2 nos dados de teste} = \frac{\text{total de casos de s2}}{\text{total de casos na amostra amostra}} \tag{2.5}$$

$$\text{Probabilidade de x1 na classe s1} = \frac{\text{total objetos da amostra com x1}}{\text{total de objetos classe s1}} \tag{2.6}$$

$$\text{Probabilidade de x2 na classe s1} = \frac{\text{total objetos da amostra com x1}}{\text{total de objetos classe s1}} \tag{2.7}$$

$$\text{Probabilidade de x1 na classe s2} = \frac{\text{total objetos da amostra com x1}}{\text{total de objetos classe s2}} \tag{2.8}$$

$$\text{Probabilidade de x2 na classe s2} = \frac{\text{total objetos da amostra com x1}}{\text{total de objetos classe s2}} \tag{2.9}$$

$$p(c|s1) = \frac{2.6 \cdot 2.7}{2.4} \tag{2.10}$$

$$p(c|s1) = \frac{2.7 \cdot 2.7}{2.5} \tag{2.11}$$

$$\text{Resultado} = max(2.10, 2.11) \tag{2.12}$$

Os autores (MCCALLUM; NIGAM et al., 1998) descrevem as diferenças existentes nos dois tipos mais comuns de implementação do classificador Naive Bayses. No primeiro caso é usado uma distribuição de Bertolli para executar os cálculos de probabilidade. No segundo caso, mais comum, é utilizado um modelo multinomial. Os autores ainda descrevem como os dois modelos são utilizados para a classificação textual, na distribuição de Bertolli são analisados os textos de forma binária podendo conter ou não uma dada palavra. Por outro lado, no modelo multinomial o texto é descrito como a quantidade de uma dada palavra. Os dois modelos podem ser usados para a classificação textual e será utilizado no desenvolvimento do experimento deste trabalho.

2.4.3 Avaliação

Os métodos de classificação apresentados devem ser avaliados para distinguir o que apresenta melhor performance para a tarefa de classificar os processos judiciais. As matrizes de confusão e a ROC curve são ferramentas para se avaliar métodos de classificação por levarem em consideração a quantidade de valores classificados corretamente e o número de resultados classificados de forma errada.

Uma matriz de classificação é criada classificando-se todos os casos do modelo em categorias, determinando se o valor previsto correspondeu ao valor real. Os dados de teste em cada categoria são contabilizados e os totais são exibidos na matriz. A matriz de classificação é uma ferramenta padrão para avaliação de modelos estatísticos e é muitas vezes chamada de matriz de confusão.

Uma matriz de classificação é uma ferramenta importante para avaliar os resultados de previsão porque facilita o entendimento e reage aos efeitos de previsões erradas. Ao exibir a quantidade e os percentuais em cada célula desta matriz, é possível consultar rapidamente com que frequência o modelo é previsto com precisão.

Outro método bastante utilizado para avaliar classificadores é chamado de curva ROC(Receiver Operator Characteristic), usada inicialmente por James P para análise de processamento de sinais, o método foi expandido para outras áreas da ciência como medicina e psicologia.

		Condição Positiva	
	Populção total	Condição positiva	Condição negativa
Valores preditos	Valores positivos	Verdadeiros Postivos	Falso positivo
	Valores negativos	Falsos negativos	Verdadeiros negativos

Tabela 1: Matriz de confusão

A curva ROC pode ser explicada através de um exemplo de classificação binária onde apenas dois valores são possíveis para a classificação. A partir dos resultados gerados pelos modelos e os reais valores da amostra é possível derivar quatro métricas dos dados: os verdadeiros positivos - resultados verdadeiros que o modelo predisse como verdadeiros -, falsos positivos - amostras verdadeiras preditas como falsa -, verdadeiros negativos - amostras de valores negativos preditas como negativos -, e falsos negativos - amostras falsas preditas positivas.

As quatro métricas apresentadas acima podem ser visualizadas na Tabela 1 juntamente como outras métricas derivadas que podem ser utilizadas para mensurar modelos de classificação.

O gráfico da curva ROC é plotado em um espaço bi dimensional em que o verdadeiros positivos são marcados no eixo Y e os verdadeiros negativos são sinalizados no eixo X. A figura xx mostra um conjunto de cinco classificadores marcados de A a E. Um classificador gera apenas um ponto na espaço bi-dimensional o que torna pouco informativo a curva ROC. os algoritmos então podem ser avaliados levando-se em consideração aspectos internos de cada algoritmo para construir um curva que mensura o valor de tp/fp para as medidas de referência dos algoritmos.

3 Metodologia

A metodologia escolhida para o desenvolvimento deste trabalho de curso foi um experimento balizado pelo GQM. Os objetivos, questões e métricas estão descritos nas próximas seções assim como o processo que será utilizado para desenvolver o trabalho. O cronograma das tarefas termina a seção metodológica.

3.1 GQM

O GQM tem função de nortear o desenvolvimento do experimento deixando claro os objetivos que se quer chegar, as questões que são levantadas a partir desses objetivos e por último as métricas que irão responder essas perguntas. Tendo escolhido o GQM como modelo balizador para o desenvolvimento deste trabalho: os objetivos, questões e métricas derivados são apresentados a seguir:

Objetivo	Porpose	Melhorar
	Issue	Processo de busca
	Object	Processos judiciais
	ViewPoint	Pesquisadores do ipea
Questão	Tempo de busca atual de processos para pesquisa?	
Métrica	Dias utilizados para encontrar processos judiciais.	
Questão	Algoritmos de aprendizagem de máquina melhoram esse tempo?	
Métrica	Porcentagem de melhorara temporal = tempo novo / tempo antigo * 100 Menor Taxa de Erro	

Tabela 2: Primeiro objetivo

Objetivo	Porpose	Avaliar
	Issue	Classificação textual
	Object	Métodos de classificação
	ViewPoint	Pesquisadores do ipea
Questão	Qual a taxa de erro dos classificadores?	
Métrica	Taxa de erro.	
Questão	Qual o melhor classificador?	
Métrica	Menor taxa de erro.	

Tabela 3: Segundo objetivo

O GQM descrito nas Tabelas 2 e 3 será subsidio para a elaboração e realização do experimento proposto nesse trabalho de conclusão de curso.

3.2 Processo

O desenvolvimento do projeto seguirá um processo definido como na Figura 3.2.

As ações e tarefas se basearam no processo definido em (AGGARWAL; ZHAI, 2012) , adaptado para o contexto do IPEA e dos processos judiciais brasileiros. O processo poderá ser repetido diversas vezes caso parâmetros das técnicas utilizadas possam ser ajustados para melhorar os resultados. Cada procedimento é detalhado adiante:

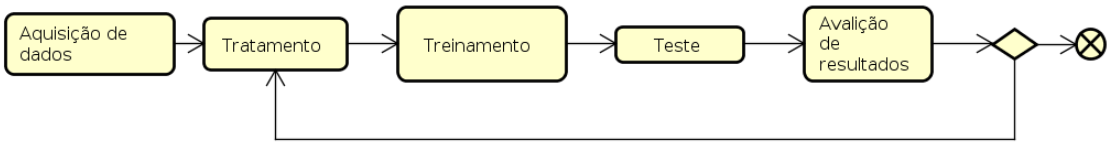

Figura 3 – Processo do experimento

1. Aquisição de dados

   A aquisição dos processo judiciais se dará através de webscraping em sites eletrônicos. No Brasil a publicação desse material é facilitada pela lei 14.419 de 2006 que em seu Art. 4o dispõe que "Os tribunais poderão criar Diário da Justiça eletrônico, disponibilizado em sítio da rede mundial de computadores, para publicação de atos judiciais e administrativos próprios e dos órgãos a eles subordinados, bem como comunicações em geral.". Um grande número de tribunais cumpre o descrito no artigo 4º e publica os seus atos em meios eletrônicos sendo essa a fonte primária de informação do projeto.

2. Tratamento

   A tarefa de tratamento consiste em tornar os dados consistentes e usáveis para análise. O tratamento utilizará as técnicas descritas na literatura seguindo os seguintes passos:

   a) Remoção de stop words

   b) Stemming

   c) Criação de tabela de frequência

As técnicas descritas acima estão implementadas em Python e fazem parte da biblioteca NLTK(Natural Language Toolkit) criada por (LOPER; BIRD, 2002). Devido a essas vantagens a NLTK será usada no projeto.

3. Treinamento

   Os dados devidamente tratados serão utilizados para o treinamento dos algoritmos de aprendizagem escolhidos na revisão bibliográfica. Nessa tarefa serão utilizadas bibliotecas já implementadas, preferencialmente na linguagem python. A biblioteca Scikit-learn, (BUITINCK et al., 2013), contém ferramentas implementadas para o treinamento do algoritmo KNN e Naive Bayses e será utilizada para executar essa fase do processo.

4. Teste

   O processo de aprendizagem de máquina descrito no item 2.4 evidencia a necessidade de se testar o algoritmo que foi treinado para verificar a acurácia do método. A fase de teste que possibilita a comparação entres os diversos métodos escolhidos. Os dados de entrada para os algoritmos serão iguais para se ter uma comparabilidade maior entre os algoritmos.

5. Avaliação de resultados

   Os testes de todos os métodos possibilitará a análise dos resultados encontrados e a definição dos melhores métodos para a classificação dos processos judiciais, além de possíveis melhorias futuras nas fases do processo.

3.3 Cronograma

As etapas definidas no processos será executadas durante o primeiro semestre de 2018 seguindo portanto o cronograma de aulas da Universidade de Brasília. A tabela abaixo mostra como as fases do processo estarão distribuídas durante o semestre. O cronograma ainda detalha as tarefas e as datas em que serão realizadas. Algumas tarefas serão executas em paralelo para a otimização do tempo e para aumentar a possibilidade de se gerar mais resultados.

O cronograma foi previsto para terminar no meio do mês de junho para que possíveis imprevistos não afetem o término do trabalho.

Fase	Tarefa	Duração	Data
Aquisição de Dados	Web Scraping sentenças TJ SP	2 semanas	05/03/2018 a 16/03/2018
	Web Scraping diários brasil	2 semanas	19/03/2018 a 30/03/2018
	Acompanhamento Processual DJE	2 semanas	02/04/2018 a 13/04/2018
Tratamento	Remover stop words	1 semana	16/04/2018 a 20/04/2018
	Remover pontuação e caracters	1 semana	23/04/2018 a 27/04/2018
	Criar tabela de frequência	1 semana	30/04/2018 a 04/05/2018
Treinamento e teste	Testar KNN	2 semanas	07/05/2018 a 18/05/2018
	Testar Naive Bayses	2 semanas	07/04/2018 a 18/04/2018
Avaliação de resultados	Descrever resultados	3 semanas	21/05/2018 a 08/06/2018

Tabela 4: Cronograma do desenvolvimento do trabalho

4 Resultados

4.1 Coleta de dados

Os dados foram recolhidos do Tribunal de Justiça de São Paulo que disponibiliza em seu domínio eletrônico os processos que tiveram sentença proferida e que não estão sobre segredo de justiça, porém, os processos não estão disponíveis como um conjunto de dados para a pronta utilização. O desenvolvimento de um scraper foi necessário pra baixar os processos para posteriores análises. Um crawler é uma software responsável por simular uma interação com uma página da internet e encontrar dados específicos para serem utilizados em análises posteriores. O software desenvolvido para executar essa funcionalidade utilizou-se de um software em python e conseguiu realizar o download de mais de um milhão de processos. O software conseguiu recuperar os seguintes dados dos processos:

1. Classe
2. Magistrado: juiz que proferiu a sentença sobre o caso
3. Comarca
4. Foro
5. Vara
6. Data de disponibilização
7. Despachos: ato do juiz responsável

Os despacho são o principal atributo para as análise posteriores. A classe e o assunto foram utilizados para definir a categoria a qual o processo pertence levando-se em consideração que serão utilizados algoritmos de aprendizagem supervisionada. Todo o processo de download dos dados levou em torno de 1 mês devido a necessidade de não comprometer os serviços dos tribunais.

4.2 Resultados

Os dados coletados do Tribunal de Justiça de São Paulo após transformação dos dados e treinamento dos algoritmos gerou os resultados que serão apresentados nas próximas seções. Os resultados foram gerados com números diferentes de amostras para cada

classe. Na primeira análise foram consideras 2000 processos por categoria e na segunda 3000.

Número	Classe
0	Abertura
1	Alvará Judicial - Lei 6858/80
2	Arrolamento Comum
3	Arrolamento Sumário
4	Auto de Prisão em Flagrante
5	Ação Civil Pública
6	Ação Penal - Procedimento Ordinário
7	Ação Penal - Procedimento Sumaríssimo
8	Ação Penal - Procedimento Sumário
9	Ação Penal de Competência do Júri
10	Ação de Exigir Contas
11	Busca e Apreensão em Alienação Fiduciária
12	Consignação em Pagamento
13	Cumprimento Provisório de Sentença
14	Cumprimento de Sentença contra a Fazenda Pública
15	Cumprimento de sentença
16	Despejo
17	Despejo por Falta de Pagamento
18	Despejo por Falta de Pagamento Cumulado Com Cobrança
19	Embargos de Terceiro
20	Embargos à Execução
21	Embargos à Execução Fiscal
22	Execução Fiscal
23	Execução da Pena
24	Execução de Título Extrajudicial
25	Falência de Empresários
26	Habilitação de Crédito
27	Homologação de Transação Extrajudicial
28	Inquérito Policial
29	Interdição
30	Inventário
31	Mandado de Segurança
32	Monitória
33	Outros procedimentos de jurisdição voluntária
34	Procedimento Comum
35	Procedimento Especial da Lei Antitóxicos
36	Procedimento do Juizado Especial Cível
37	Processo Administrativo
38	Reclamação Pré-processual
39	Reintegração / Manutenção de Posse
40	Requisição de Pequeno Valor
41	Termo Circunstanciado
42	Usucapião

Tabela 5: Classe de processo encontradas

As categorias encontradas nos processos adquiridos na fase de coleta são apresentadas na Tabela 5. As classes serão referenciadas pelos seus números nos gráficos para facilitar a sua construção.

4.2.1 Resultados para 2000 amostras

No primeiro resultado foram utilizadas 2000 amostras de cada classe de processos. O gráfico abaixo mostra o número de vizinhos utilizado na avaliação do método KNN. A Figura 4.2.1 mostra que o melhor K foi encontrada com 8 vizinhos.

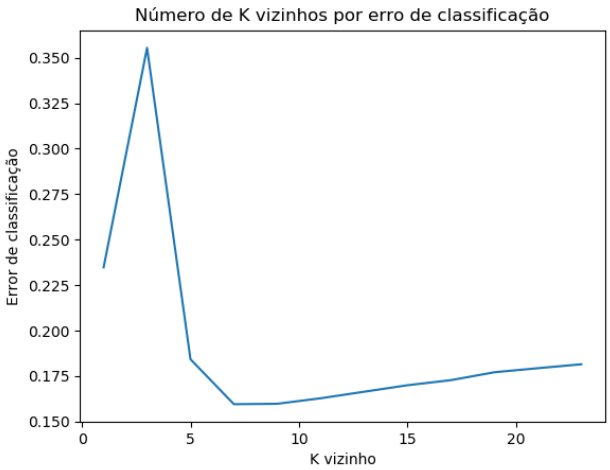

Figura 4 – Erro KNN para K para 3000 amostras

A matriz de confusão da Figura 4.2.1 para cada o algoritmo KNN pode ser vista abaixo mostrando bons resultados para a maioria das classes, porém tendo os resultados mais baixos para as classes Despejo e Despejo por falta de pagamento. Esse resultado pode estar relacionado ao contexto das duas classes que tem expressões semelhantes. O escopo do trabalho não englobou um analista jurídico que poderia ajudar na junção de classes semelhantes como as duas citadas para uma melhor performance dos algoritmos.

A matriz de confusão para o algoritmo naive bayses por sua vez gerou a matriz de confusão da Figura 4.2.1. O algoritmo também teve dificuldades em distinguir a classe Despejo , contudo o algoritmo ainda teve mais dificuldades de separar a classe Ação Penal - Procedimento Sumário.

A curva ROC da Figura 4.2.1 mostra que os dois algoritmos desempenharam suas tarefas de classificar os processos judiciais de forma bem semelhante demostrando uma área semelhante de curvas. Nesse caso qualquer um dos dois algoritmos seria satisfatório para a tarefa.

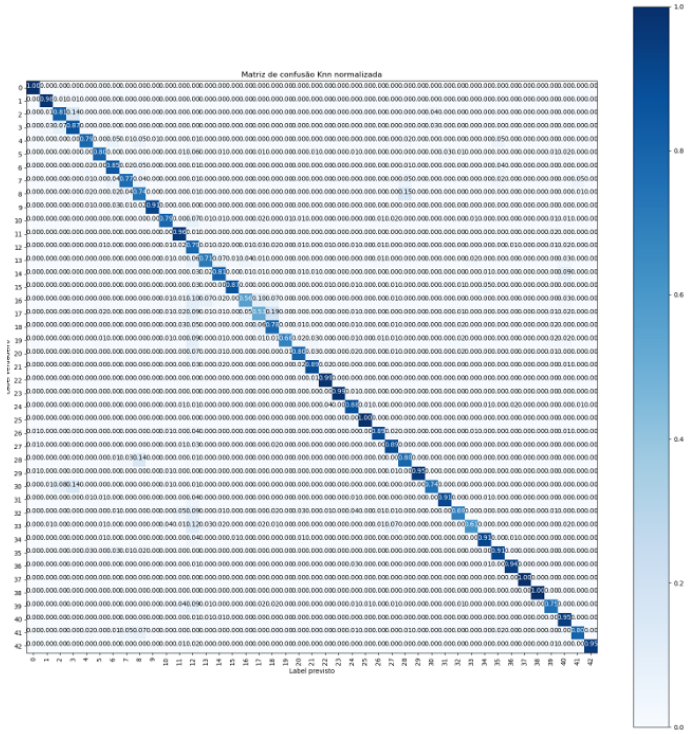

Figura 5 – Erro KNN para K com 3000 amostras

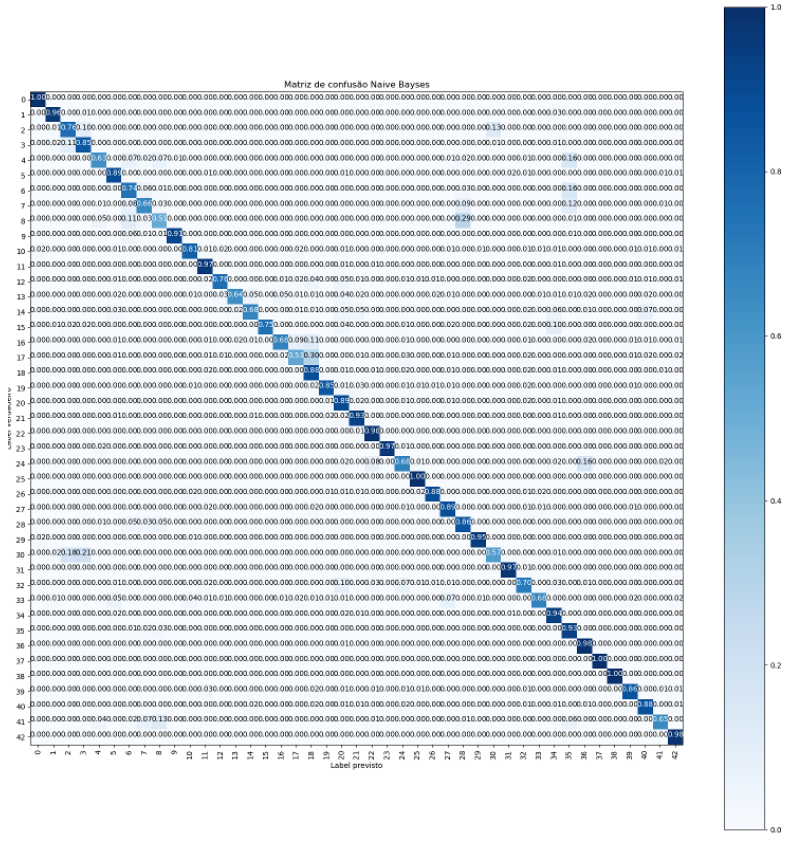

Figura 6 – Matriz de confusão Naive Bayses para 2000 amostras

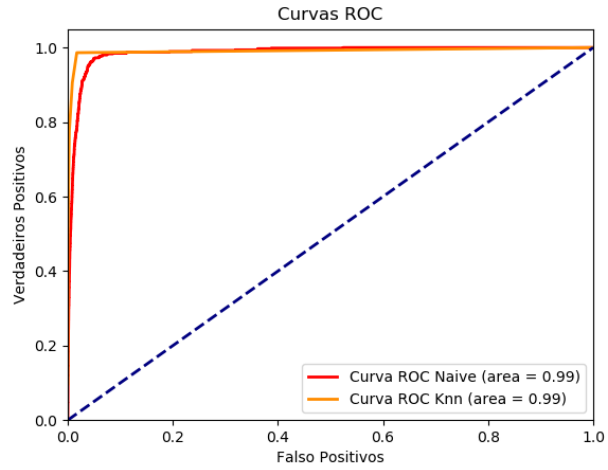

Figura 7 – Curva ROC para 2000 amostras

4.2.2 Resultados para 3000 amostras

Os resultados apresentados nessa seção foram gerados com uma quantidade maior de amostras por classe para se observar a performance de cada algoritmo. O primeiro passo como nos resultados anteriores foi a descoberta do k ótimo para os dados. A Figura 4.2.2 permite visualizar que o menor erro de classificação é encontrado para k = 10 diferentemente do primeiro resultado onde o melhor valor pode ser 8 ou 9. O gráfico também mostra que o erro de classificação se mantém estável para k > 10.

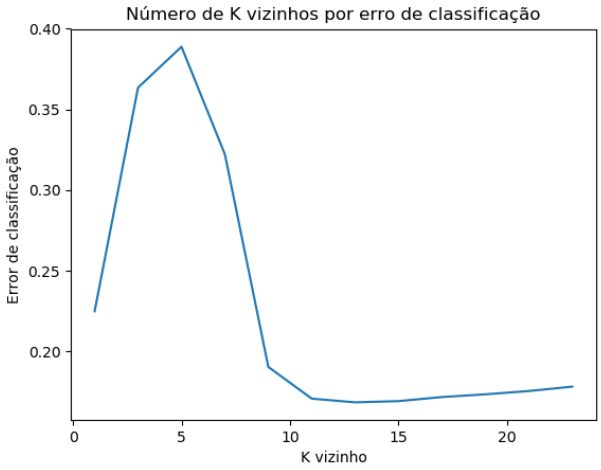

Figura 8 – Erro KNN para K com 3000 amostras

A matriz de confusão do KNN para o segundo grupo de treinamento acenta a percepção da dificuldade do algoritmo de separar as classes 16 e 17 que de novo são as classes como o menor índice de classificação da matriz.

A matriz de confusão da Figura 4.2.2 apresenta resultados semelhantes as matrizes apresentadas anteriormente. As classes de com o menor indice de classificação continuam

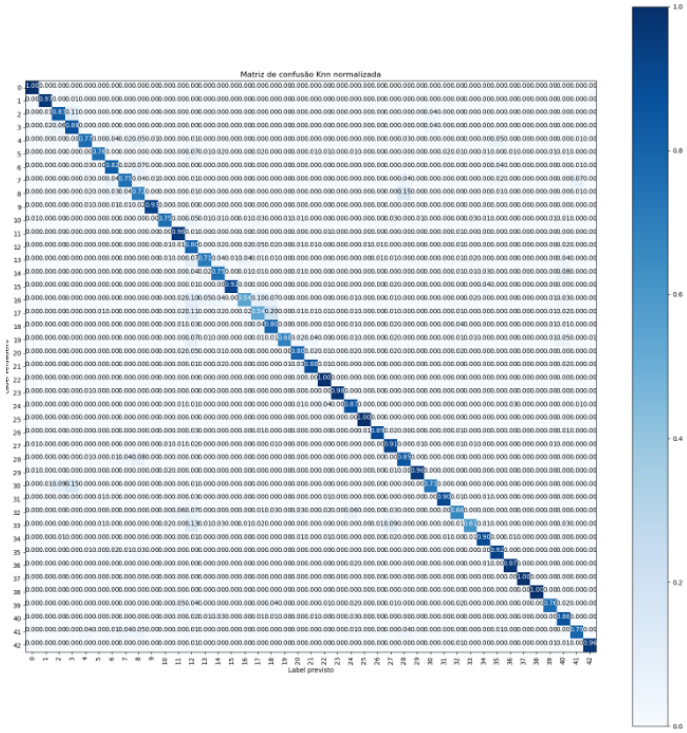

Figura 9 – Matriz de confusão KNN para 3000 amostras

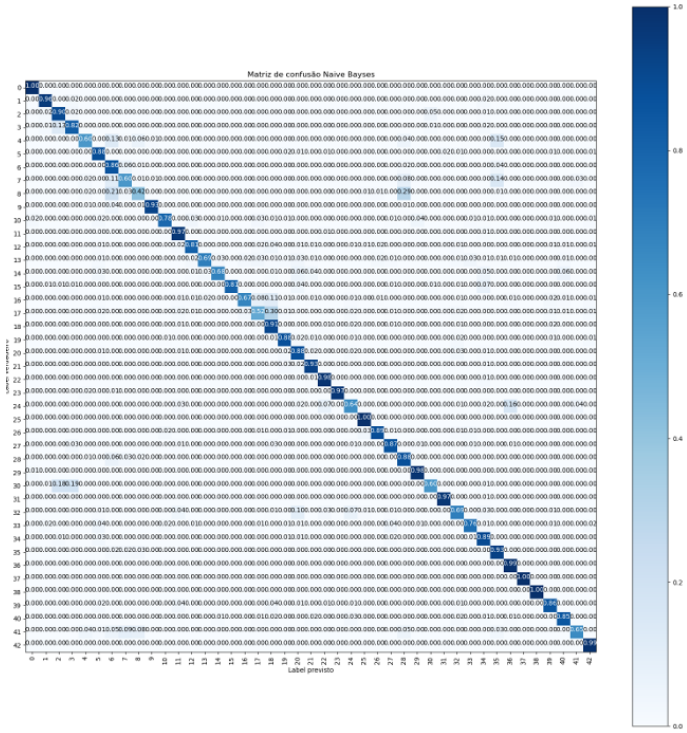

Figura 10 – Matriz de confusão Naive para 3000 amostras

sendo as mesmas.

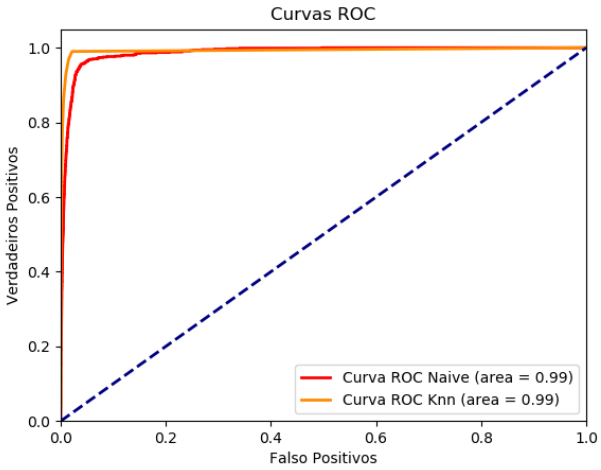

Figura 11 – Erro KNN para K de 0 a 20

A curva ROC não sofre alterações da Figura 4.2.1 para a Figura 4.2.2 mostrando que para os dois conjuntos de dados os algoritmos conseguem bons resultados na separação de processos judiciais.

5 Conclusão

Os métodos de classificação manuais utilizados para as pesquisas no IPEA são trabalhosos e consomem muito tempo dos pesquisadores, os métodos computacionais de classificação tem um grande potencial de utilização para melhorar a tarefa inicial das pesquisas otimizando o tempo dos pesquisadores.

O universo de processos judiciais são uma fonte de pesquisa inexplorados e de grande possibilidades para pesquisadores da área de text mining. A quantidade de litigância no pais pode ser analisada e melhorada para gerar ideias para a administração pública se adequar as mudanças que o pais vem enfrentando.

Os métodos de classificação naive bayse e knn são mecanismo simples e eficientes para a classificação de processos dada a baixa complexidades dos seus algoritmos. Os dois métodos vem sendo aplicados de forma eficiente em outros contextos e mostram grande eficácia para os casos dos processos judiciais. Os métodos de avaliação não foram capazes de gerar resultados conclusivos para apontar qual dos dois teria uma performance melhor porém mostrou que os dois podem ser usados nesse contexto. As pesquisas podem ser ampliadas utilizando-se de processos de outros tribunais além do TJ-SP para verificar se o comportamento dos algoritmos permanecem inalterados.

Uma outra melhoria para auxiliar em pesquisas futuras seria a implantação de webservices nos tribunais para facilitar a aquisição dos dados e a maior participação social no poder judiciário.

Referências

AGGARWAL, C. C.; ZHAI, C. **Mining text data**. [S.l.]: Springer Science & Business Media, 2012. Citado na página 30.

BRASIL. Emenda constitucional nº 45, de 30 de dezembro de 2004. 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/emendas/emc/emc45.htm>. Citado na página 19.

BUITINCK, L. et al. API design for machine learning software: experiences from the scikit-learn project. In: **ECML PKDD Workshop: Languages for Data Mining and Machine Learning**. [S.l.: s.n.], 2013. p. 108–122. Citado na página 31.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, v. 13, n. 1, p. 21–27, January 1967. ISSN 0018-9448. Citado na página 24.

HAN JIAWEI; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. [S.l.]: Elsevier, 2012. Citado na página 13.

HARRIS, Z. S. Distributional structure. 1954. Citado 2 vezes nas páginas 20 e 21.

IJSMI, E. Natural language processing concepts and methods revisited. **International Journal of Statistics and Medical Informatics**, v. 4, n. 1, 2017. Disponível em: <http://www.ijsmi.com/Journal/index.php/IJSMI/article/view/8>. Citado na página 14.

LOPER, E.; BIRD, S. Nltk: The natural language toolkit. In: **Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ETMTNLP '02), p. 63–70. Disponível em: <https://doi.org/10.3115/1118108.1118117>. Citado na página 31.

LOVINS, J. B. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, v. 18, n. 1, 1968. Citado na página 21.

MANNING, P. R. C. D.; SCHUTZE, H. **Introduction to Information Retrieval**. [S.l.: s.n.], 2008. Citado 2 vezes nas páginas 22 e 23.

MCCALLUM, A.; NIGAM, K. et al. A comparison of event models for naive bayes text classification. In: MADISON, WI. **AAAI-98 workshop on learning for text categorization**. [S.l.], 1998. v. 752, p. 41–48. Citado na página 27.

NILSSON, N. J. Introduction to machine learning. an early draft of a proposed textbook. Citeseer, 1996. Citado 2 vezes nas páginas 23 e 24.

PORTER, M. An algorithm for suffix stripping. **Program**, v. 14, n. 3, p. 130–137, 1980. Disponível em: <https://doi.org/10.1108/eb046814>. Citado na página 22.

PRODANOV, C. C. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. [S.l.]: Feevale, 2013. Citado na página 15.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, July 1959. ISSN 0018-8646. Citado na página 23.

SAMWORTH, R. J. Optimal weighted nearest neighbour classifiers. **Ann. Statist.**, The Institute of Mathematical Statistics, v. 40, n. 5, p. 2733–2763, 10 2012. Disponível em: <https://doi.org/10.1214/12-AOS1049>. Citado na página 25.

SAVOY, J. Light stemming approaches for the french, portuguese, german and hungarian languages. ACM, New York, NY, USA, p. 1031–1035, 2006. Disponível em: <http://doi.acm.org/10.1145/1141277.1141523>. Citado na página 22.

TRAVASSOS G. H.; GUROV, D.; AMARAL, E. A. G. d. Introdução à engenharia de software experimental. COPPE / UFRJ, 2002. Citado na página 15.

YONG, Z. An improved knn text classification algorithm based on clustering. **Journal of Computers**, v. 4, n. 3, March 2009. Citado na página 25.