FARO

Ferramenta de Análise de Risco de Ouvidoria

Coordenação-Geral de Inteligência de Dados
Diretoria de Informações Estratégicas
Secretaria de Combate à Corrupção

CGU

Controladoria-Geral da União

LISTA DE ILUSTRAÇÕES

Figura 1 – Quantidade de denúncias por rótulo 18
Figura 2 – Quantidade de anexos por tipo de arquivo 19
Figura 3 – Exemplo de extração e expansão de dados estruturados 21
Figura 4 – Distribuição das denúncias por rótulo utilizando 2 classes 24
Figura 5 – Métricas de acordo com o número de *features* 26
Figura 6 – Distribuição dos scores por métrica e modelo 27
Figura 7 – Curva Característica de Operação do Receptor (ROC) 28
Figura 8 – Distribuição dos scores por métrica e modelo 30
Figura 9 – Curva Característica de Operação do Receptor (ROC) 31

LISTA DE TABELAS

Tabela 1 – Comparativo das soluções avaliadas 12
Tabela 2 – Visão geral do dataset após o processamento das denúncias 22
Tabela 3 – Métricas de acordo com o número de *features* 26
Tabela 4 – Matriz de confusão do modelo, avaliando-se pelos dados de teste 28
Tabela 5 – Relatório de classificação, avaliando-se pelos dados de teste 28
Tabela 6 – Exemplo de *features* geradas pelo TF-IDF 29
Tabela 7 – Matriz de confusão do modelo, avaliando-se pelos dados de teste 31
Tabela 8 – Relatório de classificação, avaliando-se pelos dados de teste 31
Tabela 9 – Matriz de confusão do modelo, avaliando-se pelos dados de teste 32
Tabela 10 – Relatório de classificação, avaliando-se pelos dados de teste 32
Tabela 11 – Matriz de confusão do modelo, avaliando-se pelos dados de teste 33
Tabela 12 – Relatório de classificação, avaliando-se pelos dados de teste 33
Tabela 13 – Matriz de confusão do modelo, avaliando-se pelos dados de teste 34
Tabela 14 – Relatório de classificação, avaliando-se pelos dados de teste 34
Tabela 15 – Tabela comparativa dos resultados obtidos para cada modelo 34

LISTA DE ABREVIATURAS E SIGLAS

BERT	*Bidirectional Encoder Representations from Transformers*
BOW	*Bag of Words*
CADIN	Cadastro Informativo de Créditos não Quitados do Setor Público Federal
CEPIM	Cadastro de Entidades Privadas sem Fins Lucrativos
CGU	Controladoria-Geral da União
CNPJ	Cadastro Nacional de Pessoa Jurídica
CPF	Cadastro Nacional de Pessoa Física
IA	Inteligência Artificial
NER	*Named Entity Recognition*
NLP	Processamento de Linguagem Natural
OCR	*Optical Character Recognition*
OGU	Ouvidoria-Geral da União
POST	*Part-of-Speech Tagging*
TF-IDF	*Term Frequency–Inverse Document Frequency*
TIC	Tecnologias da Informação e Comunicação
VSM	Vector Space Model

SUMÁRIO

1 INTRODUÇÃO 9
1.1 Motivação 9
1.2 Estrutura do Documento 10

2 REVISÃO BIBLIOGRÁFICA 11
2.1 Considerações Iniciais 11
2.2 Trabalhos Relacionados 11
2.3 Processamento de Linguagem Natural 13
2.3.1 Bag of Words *13*
2.3.2 Part-of-Speech Tagging *14*
2.3.3 Word Embedding *14*
2.3.4 Named Entity Recognition *15*
2.4 Discussão 15

3 MODELO PARA CLASSIFICAR AS DENÚNCIAS 17
3.1 Considerações Iniciais 17
3.2 Detalhamento do Problema 17
3.3 Coleta de Dados 18
3.3.1 Extração de Textos dos Anexos *19*
3.3.2 Extração de Dados Estruturados *20*
3.3.3 Expansão das Informações *20*
3.3.4 Qualificação dos Dados Estruturados Encontrados *20*
3.3.5 Preparação do Dataset *21*

4 AVALIAÇÃO 23
4.1 Considerações Iniciais 23
4.2 Recursos Utilizados 23
4.3 Experimentos Propostos 24
4.3.1 Definições Metodológicas *25*
4.3.2 Modelo considerando apenas os dados estruturados *25*
4.3.3 Modelo considerando apenas os textos *29*
4.4 Combinação dos Modelos 32
4.4.1 Combinação dos modelos pela menor probabilidade *32*
4.4.2 Combinação dos modelos pela maior probabilidade *33*

4.4.3 Combinação dos modelos pela média das probabilidades 33
4.5 Discussão 34

5 CONCLUSÃO 37

REFERÊNCIAS 39

GLOSSÁRIO 41

CAPÍTULO

1

INTRODUÇÃO

1.1 Motivação

No ambiente de negócios, o gerenciamento de reclamações de clientes tem se tornado um fator crítico de sucesso para manter um bom relacionamento e a fidelidade de clientes, conforme afirmam Coussement e Poel (2008). No ambiente da administração pública, onde o cidadão é o cliente, a prestação de serviços à população deve ser guiada por princípios como qualidade, eficiência e economicidade. Assim, diversos canais são disponibilizados aos cidadãos para entender melhor suas necessidades ou mesmo receber informações importantes sobre anomalias ou ilicitudes identificadas.

Centenas de denúncias são enviadas diariamente através da Plataforma Integrada de Ouvidoria e Acesso à Informação (Fala.BR). Para que possam ser apuradas, primeiramente elas precisam ser triadas por uma equipe da Ouvidoria-Geral da União (OGU), área integrante da Controladoria-Geral da União (CGU), responsável por exercer as competências de órgão central do sistema de ouvidoria do poder executivo federal. O processo de triagem consiste em avaliar se há o mínimo de informações necessárias presente na denúncia e, caso haja, encaminhar a denúncia para que uma área competente realize a apuração dos fatos denunciados.

Em média, apenas 30% das denúncias que chegam são consideradas aptas pelos especialistas da Ouvidoria. Com isso, uma quantidade alta de esforço é gasta pela equipe em denúncias que não possuem o mínimo de informação para qualquer tipo de apuração. O volume de denúncias é um obstáculo que exige uma grande equipe trabalhando no processo de triagem.

Assim, a criação de um modelo de classificação que habilite denúncias permite direcionar melhor os esforços das equipes responsáveis pelo processo de triagem. O objetivo deste trabalho é propor uma metodologia que permita a implementação deste modelo no processo de triagem de denúncias da OGU.

O dados de denúncias, recebidos através do sistema Fala.BR, possuem poucos campos

estruturados. A essência da denúncia está contida no texto escrito pelo cidadão e nos diversos anexos que, por vezes, são enviados em conjunto. Para uma pessoa, a análise de uma denúncia, na maioria das vezes é uma tarefa simples. Basicamente, o analista procura por elementos no texto que possam ser confirmados por consultas em sistemas oficiais como nomes de pessoas, números de CPF , números de CNPJ, endereços, órgãos, contratos, convênios, valores, além de outros elementos que possam remeter a uma certa gravidade sobre o problema exposto pelo cidadão.

Realizar este tipo de análise computacionalmente não é uma tarefa trivial. O Processamento de Linguagem Natural (NLP) é uma subárea da Inteligência Artificial (IA) cujo o foco é a extração de informação. Este campo é vasto e é possível, por exemplo, utilizar abordagens estatísticas como em Manning e Schutze (1999) ou, ainda, abordagens simbólicas como em Justeson e Katz (1995).

Assim, para que se possa formular uma metodologia eficaz, será necessário compreender as diferentes formas de extração de informação a partir de textos e, ainda, analisar as principais técnicas e/ou algoritmos que podem ser utilizados para criar um modelo que atenda às necessidades propostas.

1.2 Estrutura do Documento

O Capítulo 1 apresenta uma breve introdução, objetivos e organização geral do trabalho. O Capítulo 2 apresenta a revisão bibliográfica acerca de assuntos pertinentes ao trabalho e trabalhos similares. O Capítulo 3 descreve a abordagem proposta para a extração de características de interesse a partir das denúncias. O Capítulo 4 discute sobre os experimentos realizados e a avaliação do modelos propostos. O Capítulo 5 reflete sobre os resultados, conclusões e evoluções futuras.

CAPÍTULO

2

REVISÃO BIBLIOGRÁFICA

2.1 Considerações Iniciais

Neste capítulo são apresentados alguns trabalhos relacionados ao tema proposto e, também, o embasamento teórico necessário para compreender os conceitos e técnicas que serão utilizados ou referenciados ao longo do estudo.

2.2 Trabalhos Relacionados

A utilização de reclamações e denúncias como subsídio para melhorar a qualidade dos serviços oferecidos pelo Estado é uma prática comum. Com a crescente evolução de técnicas e capacidade computacional nos últimos anos, é natural observar um aumento do seu emprego na automatização de processos relacionados à classificação e análise desse conjunto imenso de informações não estruturadas que chegam aos órgãos públicos constantemente. Esse entendimento de que associar tecnologia, serviços governamentais e participação social é uma tendência, é afirmado em Mehr, Ash e Fellow (2017) e enfatizado em Chun *et al.* (2010, p.1):

> A revolução nas tecnologias da informação e comunicação (TIC) vem mudando não apenas o cotidiano das pessoas, mas também as interações entre governos e cidadãos. O governo digital ou o governo eletrônico começou como uma nova forma de organização pública que suporta e redefine as informações novas e existentes, as comunicações e as interações relacionadas às transações com as partes interessadas (por exemplo, cidadãos e empresas) por meio das TIC, especialmente por meio de serviços de internet, com o objetivo de melhorar o desempenho e os processos do governo.

Na Alemanha explorou-se a ideia de que seria possível complementar modelos frequentemente utilizados em investigação forense eleitoral com informações de denúncias de cidadãos

a respeito das eleições. Tais modelos utilizam métodos estatísticos, geralmente baseados em contagem de votos e na quantidade de eleitores elegíveis, para avaliar se os resultados de eleições refletem as intenções dos eleitores ou, ainda, avaliar se há indícios de fraudes no processo eleitoral. O uso dessas denúncias é uma forma de incorporar informação contextual e isso pode ser útil para avaliar e refinar os modelos existentes (MEBANE; KLAVER; MILLER, 2016).

Em Liyanage *et al.* (2018), propõe-se uma forma automatizada de priorização de problemas urbanos reportados por cidadãos. O processo sugerido utiliza informações textuais, imagens e votos dados pelos cidadãos em cada problema reportado. Com estas informações o modelo proposto atribui uma probabilidade de o problema ser de alta importância permitindo assim, uma ordenação por prioridade.

Outra forma interessante de automatizar serviços oferecidos aos cidadãos pode ser vista em Tjandra, Warsito e Sugiono (2015). Neste trabalho, apresenta-se a forma como a administração da cidade de Surabaia na Indonésia, segunda mais importante do país, resolve o direcionamento de denúncias e outras comunicações. A combinação de algumas técnicas de preprocessamento de textos e o uso de um algoritmo de classificação de documentos permitem apontar para qual departamento da cidade a denúncia ou comunicação deverá ser direcionada.

Talvez o trabalho mais próximo do que se pretende avaliar, seja o proposto em Coussement e Poel (2008). Nele, os autores incluem, como *features* do modelo, informações sobre o estilo de escrita tais como quantidade de verbos, pronomes, palavras únicas, palavras de negação, palavras afirmativas, artigos, preposições entre outras informações derivadas do texto. A utilização destes elementos de estilo linguístico juntamente com vetores de palavras como TF-IDF, tradicionalmente utilizados em atividades de classificação de documentos, aumenta significativamente, conforme afirmam os autores, a performance de um classificador de reclamações ou denúncias.

A Tabela 1 apresenta uma visão comparativa para as soluções avaliadas. Nela, é possível avaliar as principais similaridades e diferenças entre as abordagens.

Tabela 1 – Comparativo das soluções avaliadas

Solução Avaliada	Pre-processamento de texto	Vetores de frequência de palavras	Informações derivadas do texto
Coussement e Poel (2008)	Sim	Sim	Sim (estilo linguístico)
Tjandra, Warsito e Sugiono (2015)	Sim	Sim	Não
Mebane, Klaver e Miller (2016)	Sim	Sim	Não
Liyanage *et al.* (2018)	Sim	Sim	Não
Presente estudo (2020)	Sim	Sim	Sim (entidades nomeadas)

Fonte: Autor.

2.3 Processamento de Linguagem Natural

A área de processamento de linguagem natural (NLP), une diferentes disciplinas como ciência da computação, estatística e linguística computacional, com o objetivo de permitir que máquinas possam, de certa forma, reconhecer informações descritas em linguagem humana. Mais especificamente, ter a capacidade de processar léxica e semanticamente uma linguagem.

2.3.1 Bag of Words

Há técnicas que permitem que se utilize apenas o conteúdo léxico de uma linguagem para que se possa extrair informação relevante. Por exemplo, ao transformar-se um texto em um vetor de palavras e suas frequências, é possível comparar e agrupar diferentes textos apenas pelas coocorrências de palavras similares. Esta forma de representação de texto é geralmente descrita na literatura como *Bag of Words* (BOW). Uma evolução desta representação é proposta por Jones (1972). Nesta abordagem, a autora propõe que a especificidade de um termo está diretamente associada a frequência deste termo em um dado documento e inversamente associada ao quão comum este termo é em uma coleção de documentos. Termos que ocorrem em todos os documentos são pouco importantes. Entretanto, termos que ocorrem com maior frequência em um determinado grupo de documentos são mais interessantes para discriminar um grupo de outro. Esta nova abordagem é o *Term Frequency–Inverse Document Frequency* (TF-IDF), ou a frequência do termo pelo inverso da frequência nos documentos e foi uma das técnicas mais utilizadas por buscadores na internet para a recuperação de informações através do uso de palavras-chave. Outros empregos comuns para esta técnica, ainda nos dias de hoje, devido a sua simplicidade e baixa necessidade computacional são nas áreas de classificação e clusterização de documentos.

Quando usa-se as abordagens descritas acima, é frequente o uso de determinados tratamentos no texto, com o objetivo de eliminar possíveis problemas de padronização além de remover ruídos que podem existir. Em qualquer linguagem, há palavras que não carregam muita informação independentemente do contexto em que estão inseridas. Este conjunto de palavras é comumente chamado de *stopwords*. Algumas *stopwords* da língua portuguesa: de; a; o; que; e; do; da; em; um. Remover as *stopwords* antes de determinadas aplicações de NLP é algo comum. Ainda na mesma linha, pode-se querer achar os radicais das palavras eliminando-se as suas flexões. Para esse propósito é comum o uso de algoritmos de *stemming* e *lemmatization*. Enquanto *stemming* apenas corta sufixos e prefixos das palavras mantendo um radical, *lemmatization* tenta chegar a uma palavra válida no vocabulário e assim é um processo mais lento e que exige desambiguação em alguns casos (MANNING; SCHUTZE, 1999).

O que foi discutido até aqui, apenas relaciona textos utilizando pesos atribuídos a palavras no documento sem que sua ordem seja considerada. Dessa forma, apesar de haver uma certa relação que permanece no todo, utilizando estas abordagens, perdemos muita informação

relacionada ao contexto em que as palavras foram utilizadas, tanto em relação ao conteúdo quanto em relação ao papel da palavra na estrutura da linguagem. Assim, as próximas técnicas discutidas tentam preservar alguns tipos de informações capturadas de cada palavra dentro de sua sentença. Ou seja, à partir de agora há uma certa captura de aspectos sintáticos e semânticos.

2.3.2 Part-of-Speech Tagging

Ao analisar-se uma palavra dentro de uma sentença, sua posição e relação com as outras palavras lhe conferem uma classe gramatical. Conhecer as diferentes funções que uma classe gramatical pode exercer quando está disposta em uma sentença pode ser uma vantagem para resolver ambiguidades da linguagem. A marcação gramatical ou *Part-of-Speech Tagging* (POST) é essencial em algumas tarefas de NLP como o reconhecimento de entidades nomeadas ou *Named Entity Recognition* (NER). Saber a classe gramatical das palavras em uma sentença pode ser crucial para saber se estamos falando de um nome de pessoa ou de um nome de uma rua no texto, por exemplo (MANNING; SCHUTZE, 1999).

2.3.3 Word Embedding

Enquanto as técnicas relacionadas a BOW mapeiam as palavras em vetores esparsos e com valores discretos, as técnicas baseadas em *word embedding* mapeiam sintaxe e semântica em um espaço de representação denominado Vector Space Model (VSM). Este vetor é, normalmente, um vetor de baixa dimensionalidade e com valores reais. Palavras similares tendem a ter vetores similares e acabam sendo representadas muito próximas no VSM. A principal ideia por trás dessas técnicas é que se você pode prever o contexto em que uma palavra pode aparecer então você pode entender o significado da palavra (GOLDBERG, 2017).

O algoritmo word2vec proposto por Mikolov *et al.* (2013) utiliza esta abordagem e consegue capturar de forma muito eficiente diversas relações semânticas entre as palavras apenas observando suas coocorrências nos textos. O processo de aprendizado é realizado de forma que se deslize sobre o texto uma espécie de janela de tamanho "n"na qual as "n"palavras situadas nela são utilizadas para prever a próxima palavra.

Já o GloVe, utiliza um mecanismo similar ao word2vec, entretanto utiliza-se também de informações de contexto global de coocorrência de palavras produzindo vetores densos de palavras mais expressivos que em comparação aos gerados pelo word2vec (PENNINGTON; SOCHER; MANNING, 2014).

O ELMo é um modelo que gera *word embeddings* de uma forma um pouco diferente do word2vec e do GloVe. Nele, significados diferentes de palavras iguais são preservados, assim ele possibilita uma melhor resolução de ambiguidades para uma dada palavra (PETERS *et al.*, 2018).

Por fim, o *Bidirectional Encoder Representations from Transformers* (BERT) criado pelo

google em 2018 é considerado o estado da arte para tarefas de NLP. Nele, além de palavras iguais poderem ter representações vetoriais diferentes para contextos diferentes, o algoritmo ainda utiliza todos os tokens próximos a palavra que está sendo avaliada para capturar seu contexto. Ou seja, se outros modelos antes permitiam avaliar os tokens a esquerda ou a direita, à partir do BERT esta captura de contexto passou a ter uma orientação bidirecional. Esse avanço permitiu que o modelo proposto se sobressaísse em 11 diferentes atividades de NLP.

2.3.4 Named Entity Recognition

Named Entity Recognition (NER) é uma das áreas de pesquisa de NLP na qual tenta-se identificar entidades específicas como nomes de pessoas, locais, nomes de empresas e números em textos (NADEAU; SEKINE, 2007). Por vezes, a extração de algumas informações chave do texto pode ser suficiente para cumprir um propósito específico. Assim, o treinamento de modelos com o propósito de reconhecer determinadas entidades em textos é muito comum em diversas áreas. Existem abordagens não supervisionadas que podem ser utilizadas, entretanto, o mais comum é que se forneçam textos com marcações informando onde está e qual o tipo de entidade está presente no texto. Assim os modelos tentam encontrar informações específicas de contexto que estejam presentes sempre que há a referência a uma entidade. Isso é muito importante quando há ambiguidades entre as entidades. Por exemplo, o nome de uma pessoa e o nome de uma rua podem ser o mesmo, mas como é possível identificar que trata-se de uma entidade e não outra. Por essa e outras razões esta é uma tarefa não trivial dentro da NLP e técnicas de aprendizado de máquina e aprendizado profundo vem sendo empregadas nos últimos anos para a evolução dos resultados.

2.4 Discussão

O que todos os exemplos mencionados na Seção 2.2 têm em comum é o uso de informações não estruturadas, reportadas por cidadãos, acerca de assuntos diversos e em uma escala difícil de atender apenas com a análise manual realizada por pessoas. Ao mesmo tempo em que a necessidade de processar e compreender corretamente as demandas torna-se cada vez mais essencial para prestar um serviço melhor ao cidadão, o grande desafio em questão é justamente como fazer isso.

Todos os trabalhos analisados se utilizam das técnicas de pré-processamento de texto e extração de vetores de palavras por documento com a finalidade de classificar documentos, *e-mails* ou textos. Entretanto, apenas Coussement e Poel (2008) utiliza informações derivadas do texto (além das palavras) para melhorar o modelo de classificação, sendo que, essas informações derivadas são obtidas apenas pela utilização de estatísticas de estilo de escrita.

Assim, uma das contribuições desse trabalho é a utilização de uma nova forma para a extração de *features* dos textos. O trabalho propõe a utilização das entidades nomeadas extraídas

dos textos como *features* para os modelos textuais.

Além disso, também se utiliza informações derivadas dessas entidades, obtidas em outras bases de dados, a fim de utilizar-se um conjunto de *features* mais rico no modelo proposto.

Outra contribuição desse estudo é a proposta de uma metodologia para a criação de um modelo de classificação que permita habilitar uma denúncia no seu processo de triagem.

CAPÍTULO

3

MODELO PARA CLASSIFICAR AS DENÚNCIAS

3.1 Considerações Iniciais

Neste capítulo pretende-se detalhar melhor o problema a ser resolvido. Posteriormente, será descrito todo o *pipeline* de coleta e tratamentos dos dados para a montagem do conjunto de dados que será utilizado para treinamento e avaliação do modelo proposto.

3.2 Detalhamento do Problema

Ao chegar uma nova denúncia através do sistema Fala.BR, da OGU, uma equipe de pessoas atua para avaliar as informações presentes na denúncia e decidir se a denúncia está apta a ser apurada ou não. Ou seja, a avaliação da equipe diz respeito a elementos presentes na denúncia que permitem uma apuração dos fatos relatados.

Em geral, estes elementos estão centrados em informações como CPF, CNPJ, Contratos e Convênios entre órgãos públicos e empresas privadas, materialidade (valores monetários descritos na denúncia ou identificados nos contratos e convênios). Com tais informações, a equipe de triagem de denúncias decide se deverá haver apuração (denúncia apta) ou não (denúncia não apta).

Uma grande parte deste trabalho pode ser automatizada. Assim, as consultas realizadas pela equipe, em diversos sistemas, podem ser substituídas por um processo automatizado, desde que seja possível identificar tais entidades no conteúdo das denúncias. Além disso, durante o processo de triagem, a equipe atribui uma nota de 0 a 100 (de 10 em 10) para definir o quão apta é uma denúncia. Por definições internas, notas entre 0 e 50 são consideradas não aptas e notas entre 60 e 100 são consideradas aptas. Tais notas podem ser utilizadas como rótulo para um aprendizado supervisionado.

Assim, sendo possível montar um conjunto de variáveis adequadas é possível propor a criação de um modelo para inferir notas às novas denúncias automaticamente.

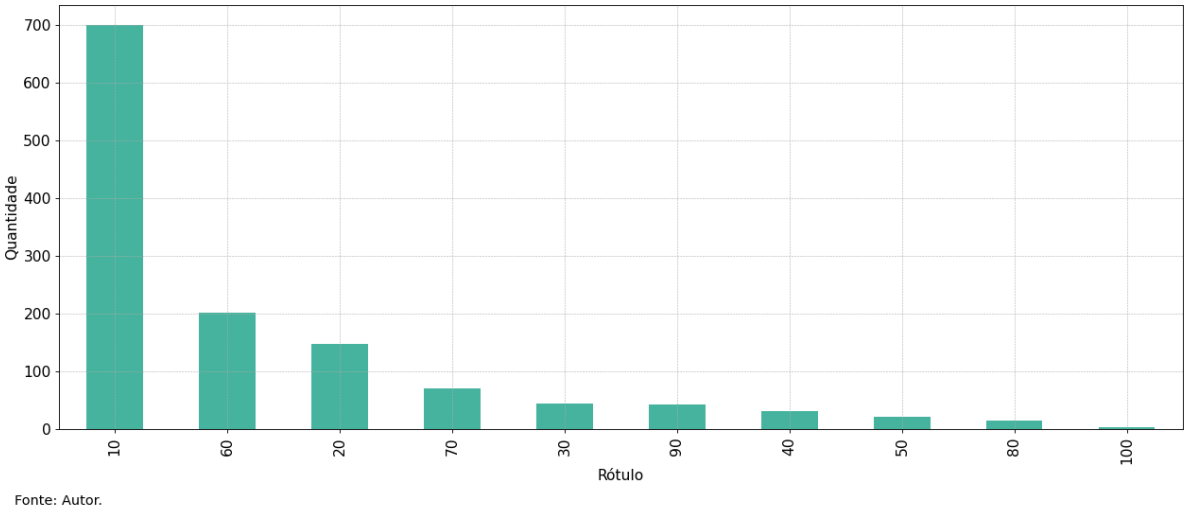
O primeiro problema a ser resolvido, então, é a coleta e preparação dos dados. A próxima seção detalha os desafios enfrentados e como foram resolvidos.

3.3 Coleta de Dados

O processo de coleta e análise de dados para utilizar em treinamentos de aprendizado de máquina, geralmente, é composto por uma fase de análise exploratória. Em muitos casos, os dados estão estruturados e o maior trabalho é uma análise em que se verificam duplicidades, dados faltantes, *outliers* e outras peculiaridades comuns em dados provenientes de sistemas informatizados. Já, no caso deste trabalho, não há, diretamente, dados estruturados. As únicas informações que podem ser obtidas são o texto da denúncia e os anexos, quando informados. Assim, a análise das denúncias e o seu uso para um aprendizado supervisionado devem abranger técnicas de NLP.

Como já foi mencionado, há rótulos sendo atribuídos às denúncias. Assim, o *dataset* que será utilizado contém 1489 registros contendo informações da denúncia, anexos e a informação de grau de aptidão. Estes registros compreendem todas as denúncias cadastradas e que possuem rótulo atribuído entre 04/12/2019 e 23/06/2020.

Figura 1 – Quantidade de denúncias por rótulo



Fonte: CGDATA/DIE.

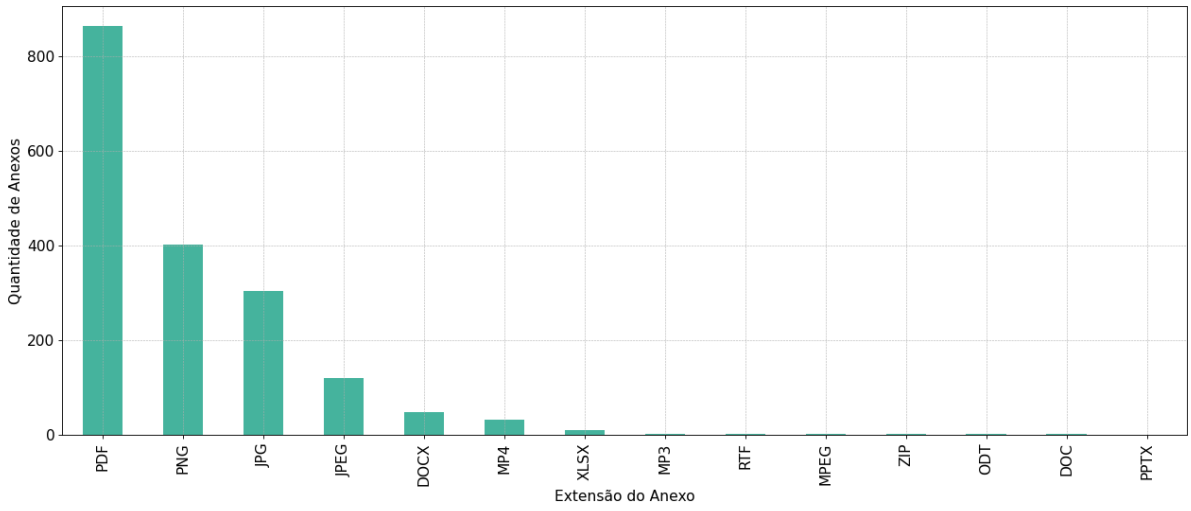
Conforme pode ser visto na Figura 1, a maior parte das denúncias recebeu um grau de aptidão 10, ou seja, são denúncias as quais os especialistas da OGU não possuem dúvidas de que as situações reportadas não devem ser apuradas.

3.3.1 Extração de Textos dos Anexos

Ao avaliar os anexos presentes em todas as denúncias cadastradas, identificou-se uma variedade muito grande de tipos de arquivos. Por essa razão, implementar um código, mesmo que utilizando bibliotecas prontas em Python (ROSSUM; DRAKE, 2009), não se provou uma alternativa viável. Não foi possível identificar nenhuma biblioteca que conseguisse, sozinha, lidar com qualquer tipo de arquivo. Por outro lado, escrever um código para identificar o tipo de arquivo e realizar a chamada para uma biblioteca especializada no tipo de arquivo em questão não seria uma tarefa trivial e fugiria demais do escopo do trabalho.

Existe, no conjunto de dados avaliado por este trabalho, um total de 1801 anexos. A Figura 2, apresenta a quantidade de anexos de acordo com o tipo de arquivo.

Figura 2 – Quantidade de anexos por tipo de arquivo



Fonte: Autor.

Assim, para possibilitar a extração dos textos dos anexos, optou-se por utilizar uma ferramenta chamada Apache Tika (disponível em <http://tika.apache.org>) implementada em java. Esta ferramenta identifica automaticamente o tipo de arquivo e extrai o conteúdo textual e os metadados. A ferramenta, ainda, é capaz de realizar *Optical Character Recognition* (OCR) quando há imagens dentro de arquivos dos tipo pdf, xls, doc, entre outros ou quando o arquivo é uma imagem.

Desta forma, configurou-se um servidor com a ferramenta instalada na forma de um serviço e utilizou-se uma biblioteca em Python para realizar as chamadas ao serviço passando os arquivos que deveriam ter o conteúdo extraído.

3.3.2 Extração de Dados Estruturados

A extração de dados estruturados é a tentativa de identificar certas entidades de interesse no conteúdo do texto da denúncia e seus anexos. As entidades de interesse, neste momento são CPF, CNPJ, convênios, contratos, NIS, nomes de pessoas, palavras fortes (palavras que indicam cenários específicos como superfaturamento, fraude, roubo, desvio, etc. ...) e valores.

Apesar de haver diversas formas de reconhecimento de entidades nomeadas, por questões de escopo e tempo, decidiu-se realizar a extração destas informações utilizando expressões regulares. A implementação de regras com expressões regulares para as entidades citadas acima é simples, na maioria dos casos, e não exige um treinamento como no caso dos algoritmos mencionados no Capítulo 2.

3.3.3 Expansão das Informações

Especialistas da OGU, ao analisar uma denúncia, pesquisam informações a partir do conteúdo das denúncias. Da mesma forma, a etapa de expansão compreende todos os dados estruturados, derivados daqueles encontrados na etapa anterior.

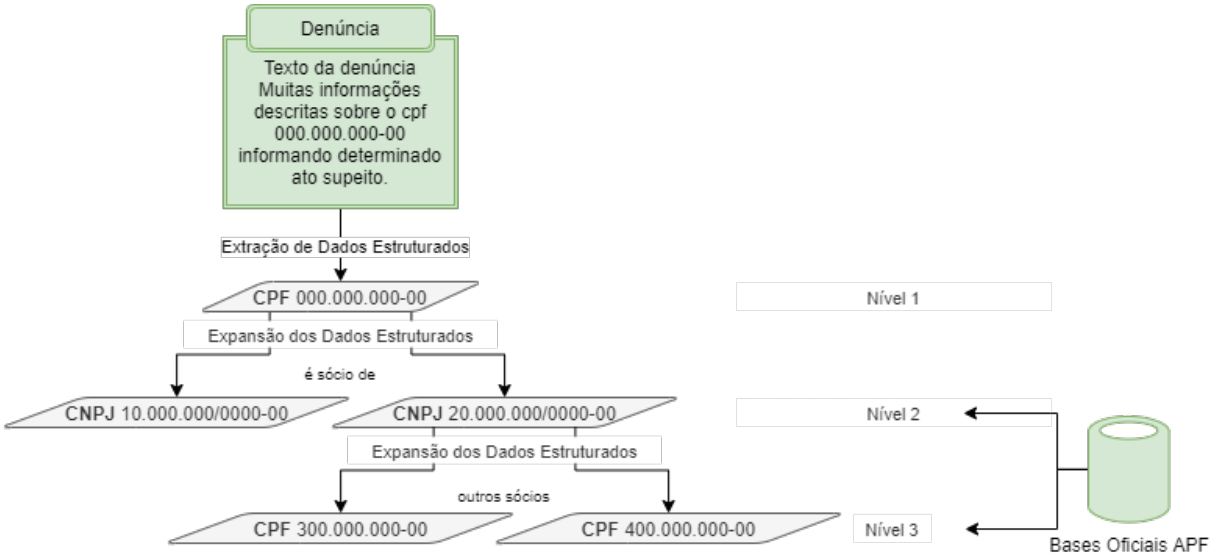
Por exemplo, para um CPF citado, pode-se avaliar em quais CNPJs este CPF consta como sócio. Em um CNPJ pode-se avaliar os sócios. Em contratos ou convênios pode-se querer avaliar os participantes (órgãos e CNPJs), além do valor envolvido no ato administrativo. Para um nome de pessoa citado, pode-se tentar buscar o CPF caso não haja homônimos na base de CPFs da Receita Federal. Estas informações são extraídas e armazenadas mantendo-se a hierarquia, ou seja, um CNPJ extraído por que um CPF aparece no quadro societário possui um apontamento para o CPF que o originou e a relação “é sócio de” é preservada. Além disso, todos os dados estruturados preservam o identificador da denúncia a qual foram extraídos.

A Figura 3 apresenta um exemplo fictício no qual o processo de extração encontra um CPF. A partir deste CPF dois CNPJs, nos quais o CPF pertence ao quadro societário, são encontrados. Para cada CNPJ encontrado busca-se os CPFs dos demais sócios. Assim, os dados estruturados provenientes da extração são definidos como nível 1, seus dados estruturados derivados nível 2 e assim por diante. Avaliou-se que, acima do nível 3, a quantidade de dados estruturados aumentaria demasiadamente e os mesmos teriam pouca relevância. Assim, na base de dados utilizada para este trabalho serão expandidos e trabalhados apenas informações até o terceiro nível.

3.3.4 Qualificação dos Dados Estruturados Encontrados

O processo de qualificação tem por objetivo extrair outras características para cada dado estruturado encontrado, independentemente do nível. Assim, para um CPF encontrado, por exemplo, pode-se extrair informações como: é ou não servidor público federal, foi demitido

Figura 3 – Exemplo de extração e expansão de dados estruturados



Fonte: CGDATA/DIE.

da administração pública federal, possui processo administrativo disciplinar, é pessoa exposta politicamente, fez doações a partidos políticos (valor total das doações), entre outras.

No caso de um CNPJ, como exemplo, podemos verificar se a empresa consta no Cadastro Informativo de Créditos não Quitados do Setor Público Federal (CADIN), Cadastro de Entidades Privadas sem Fins Lucrativos (CEPIM), se já teve outras sanções feitas pela administração pública federal, se a empresa fez doações a partidos políticos, entre outras.

As qualificações encontradas podem ser utilizadas como *features* das denúncias.

3.3.5 Preparação do Dataset

Esta etapa consiste na consolidação, em *features* ou características, de todas as informações encontradas nas etapas anteriores. Assim, realiza-se um agrupamento por origem de informações. Algumas características como materialidade são somadas outras como CPF de servidor público são contabilizadas. Dessa forma, para cada denúncia, há um registro com uma quantidade grande de colunas nas quais consolidam-se todas as informações encontradas nas etapas anteriores. Por fim, inclui-se o rótulo. A Tabela 2 apresenta uma visão em alto nível de como o *dataset* é montado bem como uma explicação da origem de algumas *features* e sua forma de consolidação.

Tabela 2 – Visão geral do dataset após o processamento das denúncias

Feature	Descrição	Forma de Consolidação
grau_aptidao	Rótulo definido pelos especialistas	Único por denúncia
txt_denuncia	Conteúdo da denúncia	Único por denúncia
txt_anexos	Conteúdo textual extraído dos anexos	Único por denúncia
nome_em_denuncia	Nomes de pessoas encontrados no texto ou anexos da denúncia. São contabilizados apenas nomes existentes na tabela de pessoa física da receita federal	Contagem de ocorrências
cpf_em_texto_denuncia	CPFs encontrados no texto ou anexos da denúncia	Contagem de ocorrências
materialidade	Referências a valores monetários encontrados diretamente no texto da denúncia e seus anexos ou em contratos e convênios extraídos das mesmas	Soma dos valores
denuncia_anonima	Indicador se a denúncia é anônima ou não	Binário (1 ou 0)
...	...	...
palavras_fortes	Palavras fortes que possam levar ao entendimento de que há ato suspeito sendo denunciado (fraude, ilegal, desvio, superfaturamento, etc.)	Contagem de ocorrências

Fonte: Autor.

Além das 7 *features* e da variável alvo (rótulo) detalhadas na Tabela 2, são extraídas outras 71 *features*. Com o dataset finalizado pode-se iniciar a avaliação dos dados para a proposição de um modelo de classificação.

CAPÍTULO

4

AVALIAÇÃO

4.1 Considerações Iniciais

Neste capítulo, serão descritos os recursos de hardware e software utilizados, o detalhamento da metodologia utilizada, os experimentos realizados, as métricas utilizadas e o modelo escolhido. Por fim, serão exibidos alguns resultados obtidos a partir da utilização do modelo selecionado aplicado em dados reais.

4.2 Recursos Utilizados

Para a realização dos experimentos, utilizou-se um servidor com 4 com processadores Intel(R) Xeon(R) CPU E5-4628L v4 @ 1.80GHz de 28 núcleos lógicos (14 núcleos físicos com *hyperthreading*), totalizando 112 núcleos, com 1 TB de memória RAM. O sistema operacional utilizado foi o CentOS Linux 7 (Core). Utilizou-se a distribuição Anaconda (Python 3.7, 64 bits). As principais bibliotecas Python utilizadas são listadas as seguir:

- imbalanced-learn==0.7.0;
- joblib==0.17.0;
- lightgbm==3.0.0;
- numpy==1.19.2;
- pandas==1.1.3;
- scikit-learn==0.23.2;
- scikit-optimize==0.8.1;

- scipy==1.5.3;
- tika==1.24;
- xgboost==1.2.1.

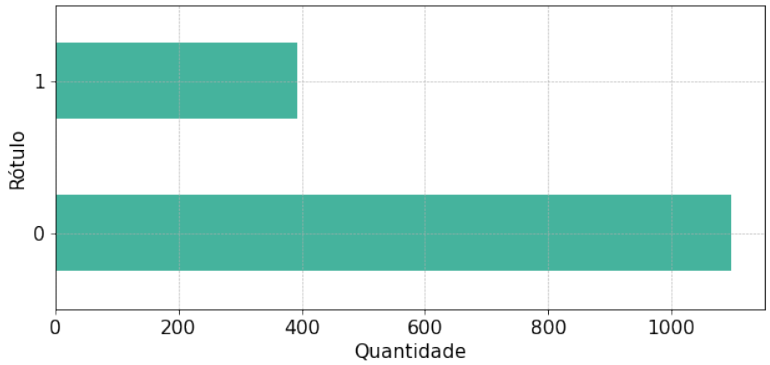
4.3 Experimentos Propostos

Nesta seção, serão exploradas algumas possibilidades de uso das informações extraídas para construir classificadores. Desta forma, identificam-se algumas alternativas óbvias que devem ser avaliadas em um primeiro momento.

A primeira trata-se de tentar construir um modelo apenas utilizando os dados estruturados extraídos dos textos. A segunda, tentar construir um modelo apenas com as informações textuais contidas nas denúncias e seus anexos. Por fim, uma combinação de ambas as alternativas.

Como mencionado na Figura 1, a distribuição de denúncias de acordo com os rótulos não é balanceada. Além disso, algumas classes possuem muito poucas observações. Por essa razão, decidiu-se trabalhar com apenas duas classes: apta (1) e não apta (0). Rótulos com valor entre 10 e 50 foram considerados como 0 e rótulos com valores entre 60 e 100 foram considerados como 1. A figura Figura 4 apresenta a distribuição das denúncias considerando apenas duas classes. Apesar de o *dataset* permanecer desbalanceado, a quantidade de denúncias por classe agora é representativa.

Figura 4 – Distribuição das denúncias por rótulo utilizando 2 classes



Fonte: CGDATA/DIE.

Neste estudo, não há a necessidade de análise e tratamento de dados faltantes. A razão para isso é que todas as informações são obtidas a partir dos textos e anexos das denúncias. Assim, no caso de textos curtos com poucas informações e nenhum anexo a denúncia tende a ser rejeitada, por que poucos ou nenhum dado estruturado é extraído e o próprio texto não traz

elementos suficientes para uma apuração. Assim, a maior parte ou todas as *features* de dados estruturados acabam tendo o seu valor zerado. Além disso, o TF-IDF gerado para o modelo textual tende a trazer a maior parte de suas *features* com valores baixos ou zerados.

4.3.1 Definições Metodológicas

Conforme já informado no Capítulo 3, o *dataset* utilizado possui 1489 registros de denúncias rotuladas como apuráveis (aptas) ou não apuráveis (não aptas). Assim, separou-se 80% (1191) dos registros em um *dataset* de treinamento e 20% dos registros (298) em um *dataset* de testes. Além disso, a separação em conjuntos de treinamento e testes foi estratificada, isto é, manteve-se a mesma proporção de registros do *dataset* original, para as classes positiva (apurável) e negativa (não apurável), em ambos os conjuntos.

Todos os procedimentos relacionados a seleção de *features*, seleção de modelos e *tunning* de hiper-parâmetros foram realizados utilizando-se o *dataset* de treinamento e as avaliações de desempenho foram realizadas utilizando-se validação cruzada no próprio *dataset* de treinamento. O *dataset* de testes foi utilizado apenas para aferir os scores finais dos modelos.

4.3.2 Modelo considerando apenas os dados estruturados

A primeira tarefa realizada nesta etapa foi a avaliação da representatividade das *features* obtidas na tentativa de explicar a variável alvo. O objetivo é diminuir a complexidade do modelo. Por questões de objetividade e tempo, optou-se por um processo automatizado.

O processo escolhido utiliza a classe *SelectFromModel* da biblioteca *scikit-learn* do Python (PEDREGOSA *et al.*, 2011). Esta classe foi utilizada para selecionar as 'k' melhores *features* após o treinamento de um *RandomForestClassifier* (BREIMAN, 2001). Estas informações ficam armazenadas no atributo *feature_importances_* do modelo. Foram realizados 39 testes para $for_{\,k} = 2, ..., 78 \; \forall \; k = 2n, \; n \in \mathbb{N}$.

Em cada iteração, é criada uma instância da classe *SelectFromModel* com o parâmetro $max\_features = k$, equivalente a iteração. A classe então é utilizada para selecionar as k melhores *features* e transformar o *dataset* de treino para descartar aquelas não selecionadas. Com o *dataset* de treino transformado, treina-se um novo *RandomForestClassifier* e realiza-se a avaliação do modelo de acordo com as métricas de *Average Precision*, *Balanced Accuracy* e *ROC AUC*. A tabela Tabela 3 e a Figura 5 apresentam um resumo dos resultados obtidos.

Com base nos resultados avaliou-se que, a partir de 20 *features*, o ganho dos scores não é tão relevante e a utilização do conjunto completo aumentaria o tempo de treinamento e a complexidade do modelo. Assim, escolheu-se utilizar as 20 mais relevantes na etapa de modelagem.

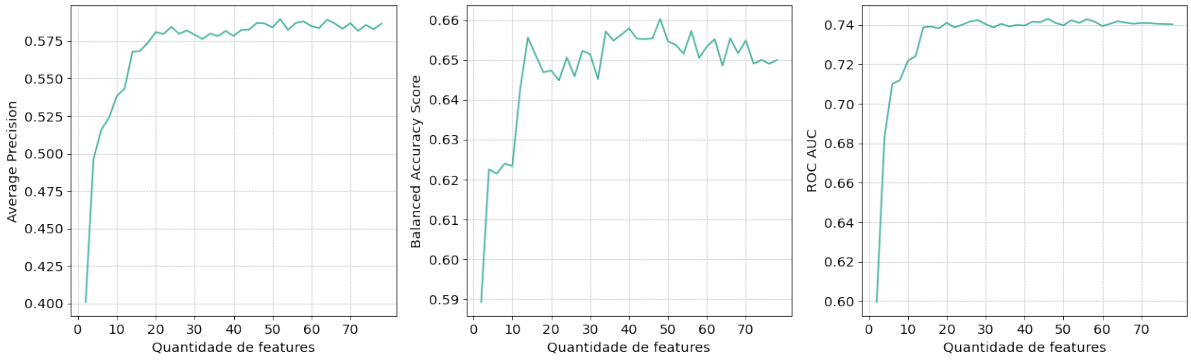
A decisão sobre qual algoritmo utilizar para a geração do modelo foi tomada com base na comparação do score de área sob a curva ROC (*roc_auc*) obtido em diversos modelos testados.

Tabela 3 – Métricas de acordo com o número de *features*

Número de *Features*	Average Precision	Balanced Accuracy	ROC AUC
10	0.538429	0.623408	0.721747
20	**0.581019**	**0.647302**	**0.741056**
30	0.579207	0.651423	0.740430
40	0.578372	0.657877	0.739688
50	0.584122	0.654597	0.739786
60	0.584938	0.653307	0.739417
70	0.587008	0.654857	0.740949

Fonte: CGDATA/DIE.

Figura 5 – Métricas de acordo com o número de *features*



Fonte: CGDATA/DIE.

Esta métrica foi escolhida por ser uma métrica interessante para avaliação do desempenho do modelo com foco na classe positiva, neste caso a classe 1.

O processo consistiu basicamente em realizar as seguintes etapas para cada algoritmo a ser avaliado:

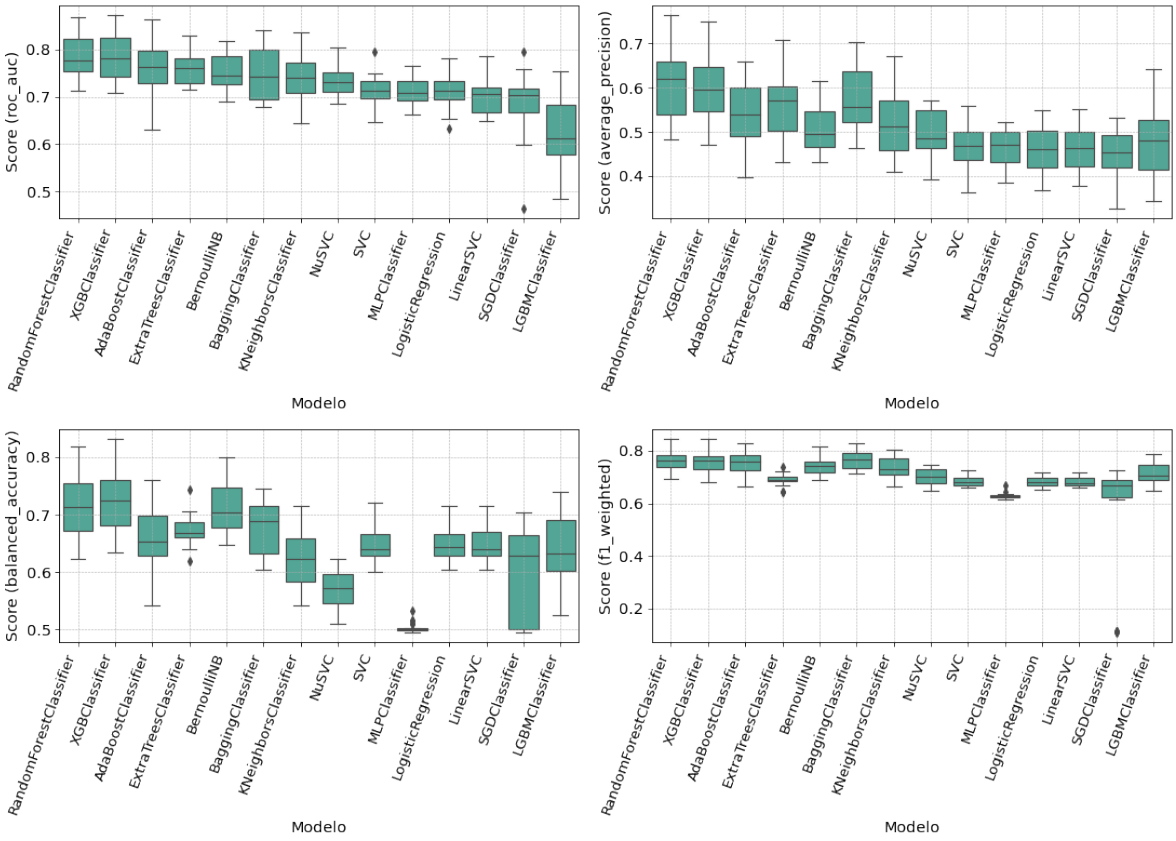
- utilizar a classe *GridSearchCV*, do *scikit-learn* com algumas combinações de hiper-parâmetros e validação cruzada com 10 partições estratificadas para escolher um boa versão de modelo e tornar mais justa a comparação;

- utilizar a melhor combinação de hiper-parâmetros encontrada para realizar uma validação cruzada com 10 partições e obter as seguintes métricas: *roc_auc*, *balanced_accuracy*, *average_precision* e *f1_weighted*;

- utilizar a distribuição de scores obtidos para cada métrica e modelo para realizar a escolha do algoritmo mais adequado ao problema.

A Figura 6 apresenta um gráfico de *boxplot* para cada algoritmo avaliado. A ordem em que os algoritmos aparecem, em todos os gráficos, foi definida através do score obtido pela métrica *roc_auc*, em ordem decrescente, da esquerda para a direita.

Figura 6 – Distribuição dos scores por métrica e modelo



Fonte: CGDATA/DIE.

O algoritmo escolhido foi o *RandomForestClassifier*, por apresentar o melhor desempenho pela métrica escolhida. Além disso, o algoritmo também teve uma boa performance nas outras três métricas avaliadas.

O passo seguinte foi a otimização dos hiper-parâmetros. Utilizou-se a biblioteca *skopt* para realizar uma busca automática. Por fim, um novo modelo foi treinado utilizando-se os parâmetros selecionados e todos os dados de treinamento. A Tabela 4 apresenta a matriz de confusão e a Tabela 5 apresenta o relatório de classificação, provenientes das previsões do modelo para os dados de teste.

O conjunto de testes utilizado possui 298 registros. Assim, como é possível observar na Tabela 4, tem-se 188 verdadeiros negativos e 50 verdadeiros positivos que somam um total de 238 classificações corretas. Ao avaliar os resultados por classe (Tabela 5), observa-se que o modelo possui uma precisão para a classe 0 (0,87) bem superior à da classe 1 (0,61). O modelo

Tabela 4 – Matriz de confusão do modelo, avaliando-se pelos dados de teste

	Predito como 0	Predito como 1
Classe 0	188	32
Classe 1	28	50

Fonte: CGDATA/DIE.

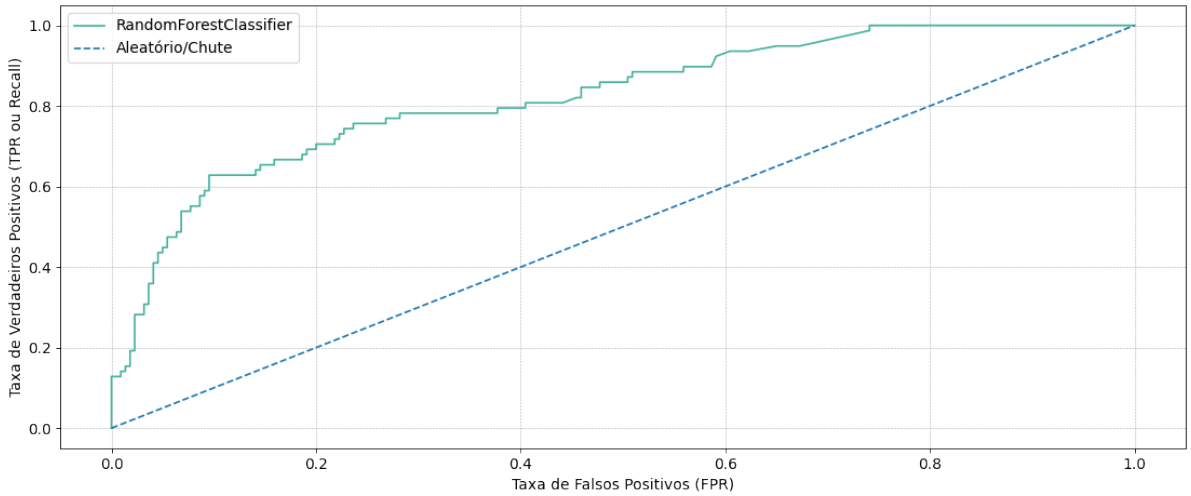
Tabela 5 – Relatório de classificação, avaliando-se pelos dados de teste

	Precisão	Revocação	F1-Score	Suporte
Classe 0	0,87	0,85	0,86	220
Classe 1	0,61	0,64	0,62	78
Acurácia			0,80	298
Média	0,74	0,75	0,74	298
Média Ponderada	0,80	0,80	0,80	298

Fonte: CGDATA/DIE.

apresenta uma acurácia de 0,80, porém, a acurácia balanceada foi de 0,75. O score *roc_auc* foi de 0.82 e a curva ROC é apresentada na Figura 7.

Figura 7 – Curva Característica de Operação do Receptor (ROC)



Fonte: CGDATA/DIE.

Pela análise da curva ROC, entende-se que seria possível elevar o *recall* do modelo diminuindo-se o limiar utilizado para computar as classes positivas e negativas. Com o limiar em 0,5 (padrão), tem-se uma taxa de falsos positivos de 0,14 e um *recall* de 0,64. Entretanto, ao mover-se o limiar para 0.37, tem-se uma taxa de falsos positivos de 0,28 e um *recall* de 0,76.

Talvez seja um *tradeoff* interessante pois o falso positivo de uma denúncia é menos problemático do que um falso negativo.

Em geral, avalia-se que a capacidade de distinção entre as classes, aprendida pelo modelo, é bastante razoável.

4.3.3 Modelo considerando apenas os textos

Nesta seção será avaliada a possibilidade de criação de um modelo utilizando apenas os textos das denúncias e de seu anexos. Como foi visto no Capítulo 2, há diferentes formas de extrair informações de textos. Elas variam bastante de acordo com o propósito pretendido.

Primeiramente realizou-se um pré-processamento em todos os documentos com o objetivo de padronizar os textos. Todas as letras foram convertidas para minúsculas, caracteres acentuados foram substituídos por seus equivalentes sem acentuação, *stopwords* foram removidas e espaços excessivos foram eliminados.

Para o propósito deste estudo, será avaliado o TF-IDF como técnica para extração de características de documentos. Como explicado anteriormente, esta técnica combina a contagem de palavras presentes no texto com a ocorrência das mesmas em outros documentos. Com isso, palavras que ocorrem em muitos documentos terão seu peso atenuado enquanto que palavras que ocorrem um conjuntos pequenos de documentos receberão um peso maior. Como exemplo, a Tabela 6 exibe algumas *features* geradas pelo TF-IDF para as primeiras 10 denúncias do *dataset* de treinamento.

Tabela 6 – Exemplo de *features* geradas pelo TF-IDF

...	ativo	ato	atos	atraves	atribuicoes	atuacao	atual	auditoria	augusto	ausencia	...
...	0,0000	0,0988	0,0658	0,0036	0,0115	0,0069	0,0156	0,0000	0,0000	0,0266	...
...	0,0084	0,0449	0,0184	0,0090	0,0107	0,0045	0,0000	0,0000	0,0027	0,0000	...
...	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	...
...	0,0000	0,0000	0,0000	0,1386	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	...
...	0,0000	0,0320	0,0145	0,0158	0,0029	0,0087	0,0074	0,0000	0,0007	0,0000	...
...	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	...
...	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	...
...	0,0000	0,0001	0,0002	0,0001	0,0000	0,0003	0,0000	0,0000	0,0006	0,0000	...
...	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,1226	0,0000	0,0000	0,0000	...
...	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	...

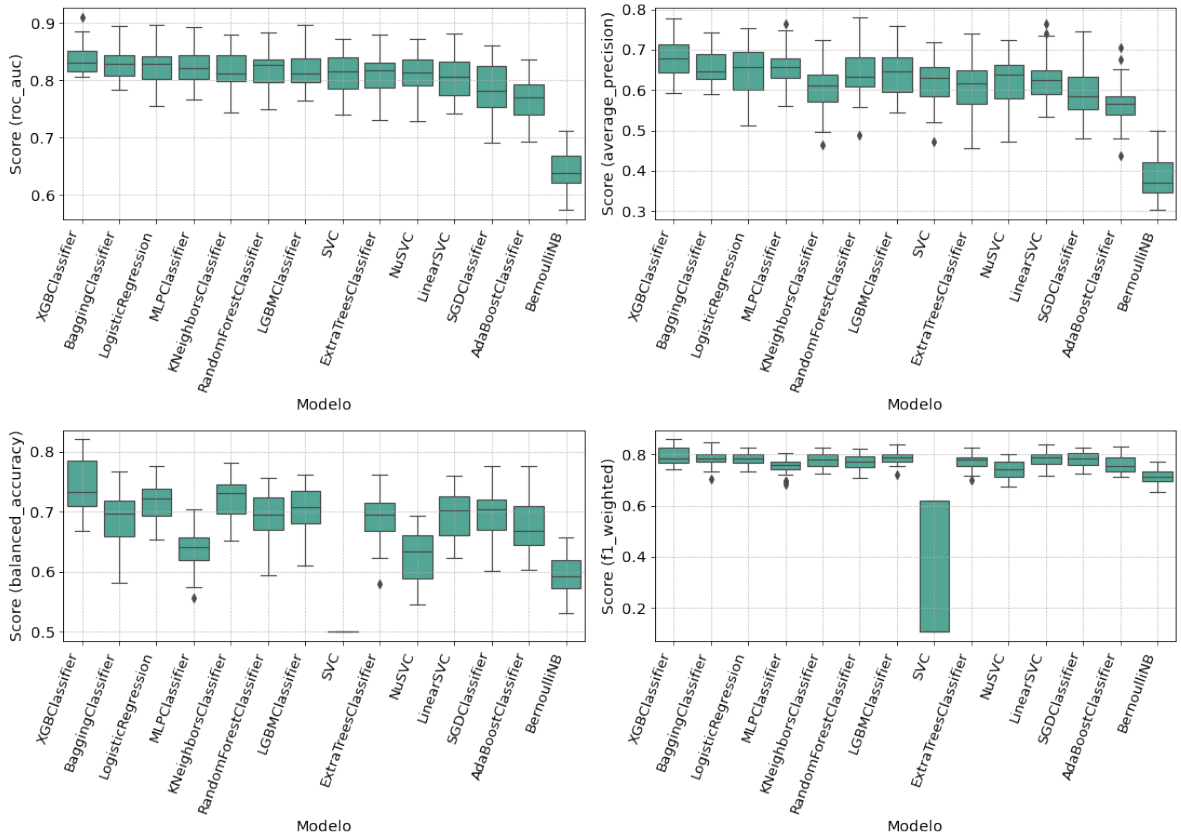
Fonte: CGDATA/DIE.

Na Tabela 6, cada linha representa uma denúncia e cada coluna representa uma palavra. Palavras que não são encontradas na denúncia recebem um peso 0 e palavras que estão presentes na denúncia recebem um peso que é ponderado pela quantidade de vezes que ela está presente no texto e, também, pela sua presença ou não nas demais denúncias analisadas. Esta matriz de palavras por documento será utilizada para o treinamento e comparação dos modelos. Para a comparação dos algoritmos, o TF-IDF foi parametrizado de forma a permitir no máximo 5.000 palavras e só serão aceitas palavras que ocorram, em pelo menos 5 documentos ou no máximo

em 50% dos documentos. Assim, elimina-se palavras que não discriminam muito bem grupos de documentos por serem específicas demais ou genéricas demais.

A mesma metodologia utilizada na escolha do algoritmo na Subseção 4.3.2 foi utilizada para a escolha do algoritmo para este modelo. A Figura 8 apresenta um gráfico de *boxplot* com o desempenho de cada modelo. Os modelos foram ordenados na figura pela métrica *roc_auc*, em ordem decrescente, da esquerda para a direita. O algoritmo que teve o melhor desempenho foi o *XGBClassifier* com um *score* de 0,830. Além disso, teve ainda, o melhor desempenho nas outras métricas avaliadas. Por essa razão, escolheu-se o *XGBClassifier* para a otimização de hiper-parâmetros e avaliação final.

Figura 8 – Distribuição dos scores por métrica e modelo



Fonte: CGDATA/DIE.

O mesmo processo de otimização utilizado para o modelo da Subseção 4.3.2 foi adotado para este modelo. Assim, após a execução da biblioteca *skopt* para encontrar bons hiper-parâmetros para o modelo, foi gerado um modelo treinado com todos os dados de treinamento. Os resultados são exibidos nas tabelas 7 e 8.

É possível observar na matriz de confusão (Tabela 7) 187 verdadeiros negativos e 47 verdadeiros positivos que somam um total de 234 classificações corretas. Ao avaliar os resultados

Tabela 7 – Matriz de confusão do modelo, avaliando-se pelos dados de teste

	Predito como 0	Predito como 1
Classe 0	187	33
Classe 1	31	47

Fonte: CGDATA/DIE.

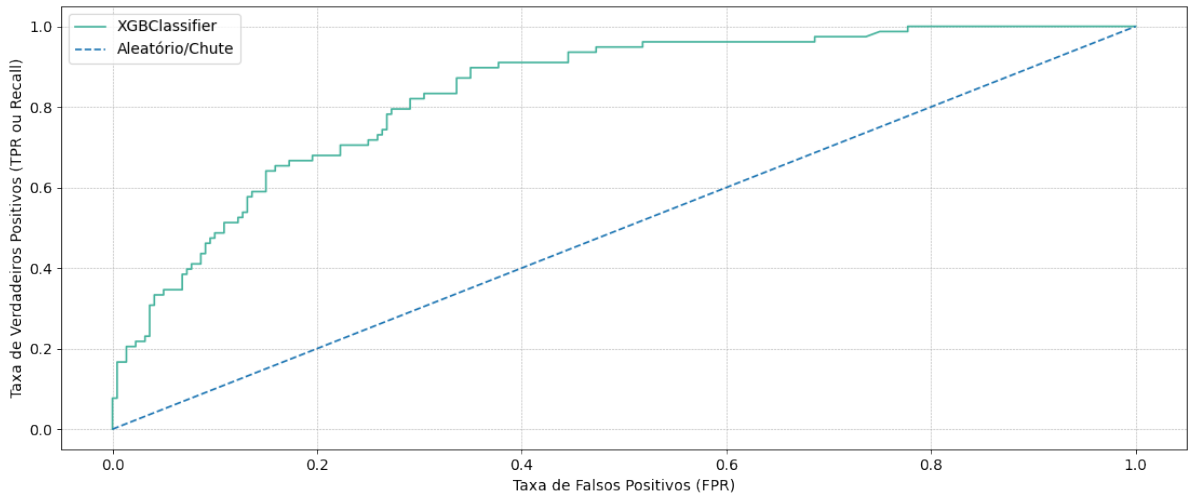
Tabela 8 – Relatório de classificação, avaliando-se pelos dados de teste

	Precisão	Revocação	F1-Score	Suporte
Classe 0	0,86	0,85	0,85	220
Classe 1	0,59	0,60	0,59	78
Acurácia			0,79	298
Média	0,72	0,73	0,72	298
Média Ponderada	0,79	0,79	0,79	298

Fonte: CGDATA/DIE.

por classe (Tabela 8), observa-se que o modelo possui uma precisão para a classe 0 (0,86) bem superior à da classe 1 (0,59). O modelo apresenta uma acurácia de 0,79, porém, a acurácia balanceada foi de 0,73. O score *roc_auc* foi de 0,83 e a curva ROC é apresentada na Figura 9.

Figura 9 – Curva Característica de Operação do Receptor (ROC)



Fonte: CGDATA/DIE.

De forma geral, pode-se afirmar que este modelo teve um desempenho bem parecido com o modelo proposto na Subseção 4.3.2. Talvez levemente inferior. Mas, ainda assim, com um poder de discriminação entre as classes razoável, conforme observa-se na Figura 9.

4.4 Combinação dos Modelos

Nesta seção, avalia-se a possibilidade de combinar os modelos obtidos nas seções 4.3.2 e 4.3.3. O objetivo aqui é identificar se algum tipo de combinação pode melhorar o desempenho da classificação em relação aos desempenhos isolados de cada modelo.

Após aplicar-se as correlações de Pearson e Spearman nas probabilidades (para a classe positiva) previstas pelos dois modelos, para os dados de teste, obtém-se, respectivamente, os coeficientes 0,6879 e 0,6419. Por esta razão, entende-se que há uma certa quantidade de registros nos quais os modelos possivelmente classificam de forma diferente. Assim, foram testadas 3 combinações possíveis. A menor, a maior e a média das probabilidades, para a classe positiva, previstas pelos dois modelos, para cada registro do conjunto de testes.

4.4.1 Combinação dos modelos pela menor probabilidade

Neste cenário, foram aplicados os dois modelos nos dados de teste. Para cada registro utilizou-se a menor probabilidade obtida entre os modelos e a outra foi descartada. Entende-se que, neste caso, tem-se uma classificação mais pessimista ou conservadora pois se, ao menos, um dos modelos indicar uma probabilidade abaixo de 0,5, o registro será classificado como pertencente à classe 0. O resultado desta combinação pode ser avaliado nas tabelas 9 e 10.

Tabela 9 – Matriz de confusão do modelo, avaliando-se pelos dados de teste

	Predito como 0	Predito como 1
Classe 0	204	16
Classe 1	39	39

Fonte: CGDATA/DIE.

Tabela 10 – Relatório de classificação, avaliando-se pelos dados de teste

	Precisão	Revocação	F1-Score	Suporte
Classe 0	0,84	0,93	0,88	220
Classe 1	0,71	0,50	0,59	78
Acurácia			0,79	298
Média	0,77	0,71	0,73	298
Média Ponderada	0,81	0,82	0,80	298

Fonte: CGDATA/DIE.

Esta combinação leva o modelo, de uma forma geral, a rejeitar mais denúncias. Isso ocasiona um aumento de precisão para ambas as classes, entretanto, a revocação ou taxa de verdadeiros positivos cai bastante. Neste caso tem-se um aumento considerável na quantidade de falsos negativos. Esse comportamento não é interessante para o problema em questão. O *ROC*

AUC score foi de 0,83 o *balanced accuracy score* foi de 0,71. Houve uma melhora na métrica ROC AUC e uma piora na acurácia balanceada quando comparado com os modelos das seções 4.3.2 e 4.3.3.

4.4.2 Combinação dos modelos pela maior probabilidade

Neste cenário, utilizou-se, para cada registro a maior probabilidade obtida entre os modelos e a outra foi descartada. Entende-se que, neste caso, tem-se uma classificação mais otimista ou arrojada pois se, ao menos, um dos modelos indicar uma probabilidade acima de 0,5, o registro será classificado como pertencente à classe 1. O resultado desta combinação pode ser avaliado nas tabelas 11 e 12.

Tabela 11 – Matriz de confusão do modelo, avaliando-se pelos dados de teste

	Predito como 0	Predito como 1
Classe 0	171	49
Classe 1	20	58

Fonte: CGDATA/DIE.

Tabela 12 – Relatório de classificação, avaliando-se pelos dados de teste

	Precisão	Revocação	F1-Score	Suporte
Classe 0	0,90	0,78	0,83	220
Classe 1	0,54	0,74	0,63	78
Acurácia			0,77	298
Média	0,72	0,76	0,73	298
Média Ponderada	0,80	0,77	0,78	298

Fonte: CGDATA/DIE.

Esta combinação leva o modelo, de uma forma geral, a aceitar mais denúncias. Isso ocasiona uma diminuição de precisão para a classe positiva, entretanto, a revocação ou taxa de verdadeiros positivos aumenta consideravelmente. Neste caso, tem-se um aumento considerável na quantidade de falsos positivos. Esse comportamento pode ser aceitável para o problema em questão. O *ROC AUC score* foi de 0,85 o *balanced accuracy score* foi de 0,76. Houve uma melhora em ambas as métricas quando comparado com os modelos das seções 4.3.2 e 4.3.3.

4.4.3 Combinação dos modelos pela média das probabilidades

Neste cenário, utilizou-se, para cada registro a média das probabilidades obtidas entre os modelos. Entende-se que, neste caso, tem-se uma classificação mais equilibrada. O resultado desta combinação pode ser avaliado nas tabelas 13 e 14.

Tabela 13 – Matriz de confusão do modelo, avaliando-se pelos dados de teste

	Predito como 0	Predito como 1
Classe 0	192	28
Classe 1	34	44

Fonte: CGDATA/DIE.

Tabela 14 – Relatório de classificação, avaliando-se pelos dados de teste

	Precisão	Revocação	F1-Score	Suporte
Classe 0	0,85	0,87	0,86	220
Classe 1	0,61	0,56	0,59	78
Acurácia			0,79	298
Média	0,73	0,72	0,72	298
Média Ponderada	0,79	0,79	0,79	298

Fonte: CGDATA/DIE.

Observa-se um equilíbrio maior entre precisão e revocação para ambas as classes. Entretanto há uma diminuição no número de acertos da classe positiva e, consequentemente, um aumento no número de falsos negativos. O *ROC AUC score* foi de 0,86 o *balanced accuracy score* foi de 0,72. Houve uma melhora para ROC AUC e uma piora para a acurácia balanceada quando comparado com os modelos das seções 4.3.2 e 4.3.3.

4.5 Discussão

A escolha de qual modelo seria mais interessante para o problema tem como baliza um ponto muito importante: enviar uma denúncia fraca ou sem elementos para a apuração é um problema muito menor do que não apurar uma denúncia com elementos suficientes e verdadeira.

A Tabela 15 apresenta as principais métricas obtidas para cada experimento apresentado nas seções 4.3 e 4.4.

Tabela 15 – Tabela comparativa dos resultados obtidos para cada modelo

Modelo	Precisão média	Acurácia Balanceada	F1-Score (Classe Positiva)	ROC AUC
Dados Estruturados	0,48	0,75	0,62	0,82
Texto	0,46	0,74	0,59	0,84
Combinação pela menor probabilidade	0,49	0,71	0,59	0,84
Combinação pela maior probabilidade	0,47	0,76	0,61	0,85
Combinação pela média das probabilidades	0,46	0,72	0,59	0,86

Fonte: CGDATA/DIE.

Entende-se que a metodologia proposta atinge o objetivo de melhorar a classificação de denúncias aptas. O modelo baseado apenas nos dados estruturados evidencia esta afirmação.

Ou seja, a derivação de informações a partir de elementos chave encontrados nas denúncias efetivamente colabora com o processo de classificação. Ainda assim, ao avaliar-se as previsões obtidas pelos dois modelos, constata-se que, em muitos casos, ambos se complementam. A consequência disso foi um melhor desempenho obtido na combinação otimista destes modelos em relação aos seus desempenhos isolados.

CAPÍTULO

5

CONCLUSÃO

Neste trabalho foi abordado como tema, o processamento automático de denúncias, enviadas através da plataforma Fala.BR. A principal contribuição apresentada, foi uma abordagem baseada na extração de entidades nomeadas. A identificação de entidades nomeadas permitiu que se utilizasse bases de dados oficiais do governo federal para o enriquecimento das informações extraídas diretamente do conteúdo da denúncia e de seus anexos. Tais informações puderam, então, ser utilizadas como variáveis em um modelo de classificação.

Outra contribuição relevante, apresentada neste trabalho, foi o desenvolvimento de uma metodologia para a extração completa de dados textuais, a partir de diversos formatos de arquivos, utilizando a linguagem de programação Python e o software Apache Tika. O processo utilizado pode ser aproveitado separadamente em qualquer trabalho de processamento de linguagem natural no qual os dados precisem ser adquiridos de arquivos.

A avaliação da metodologia, em dados reais, demonstrou que ela melhora a capacidade de classificação do modelo, tanto isoladamente quanto em combinação com técnicas tradicionais como vetores de palavras por documento, como o TF-IDF. Considera-se que os resultados obtidos pelo modelo final são adequados para o seu emprego no fluxo de trabalho da CGU. Sua adoção permitirá que parte da equipe responsável pela triagem de denúncias possa ser realocada em outras atividades, preservando, ainda assim, a qualidade na triagem das denúncias dos cidadãos.

Em trabalhos futuros, pretende-se avaliar técnicas mais avançadas para NER, assim pode-se expandir a quantidade de entidades nomeadas identificáveis, mantendo-se ou até aprimorando-se a qualidade da detecção sem haver a necessidade de um aumento excessivo na capacidade de processamento. Isso permitiria a obtenção de melhores *features* para o modelo baseado em dados estruturados. Assim, o estudo de representações numéricas mais elaboradas para textos, como ELMo, GloVe e BERT serão essenciais, tanto para as novas técnicas de NER quanto para a utilização de arquiteturas de aprendizado profundo, que também serão foco de estudo. Por fim, mas não esgotando todas as possibilidades de melhorias, pretende-se utilizar *autoencoders*

com o objetivo de aproveitar o imenso passivo de denúncias não rotuladas para transformar as *features* baseadas em dados estruturados em um conjunto mais relevante de *features*.

REFERÊNCIAS

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 25.

CHUN, S.; SHULMAN, S.; SANDOVAL, R.; HOVY, E. Government 2.0: Making connections between citizens, data and government. **Information Polity**, IOS Press, v. 15, n. 1, 2, p. 1–9, 2010. Citado na página 11.

COUSSEMENT, K.; POEL, D. Van den. Improving customer complaint management by automatic email classification using linguistic style features as predictors. **Decision Support Systems**, Elsevier, v. 44, n. 4, p. 870–882, 2008. Citado nas páginas 9, 12 e 15.

GOLDBERG, Y. Neural network methods for natural language processing. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017. Citado na página 14.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of documentation**, MCB UP Ltd, 1972. Citado na página 13.

JUSTESON, J. S.; KATZ, S. M. Technical terminology: some linguistic properties and an algorithm for identification in text. **Natural Language Engineering**, Cambridge University Press, v. 1, n. 1, p. 9–27, 1995. Citado na página 10.

LIYANAGE, Y.; YAO, M.; YONG, C.; ZOIS, D.-S.; CHELMIS, C. What matters the most? optimal quick classification of urban issue reports by importance. In: IEEE. **2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. [S.l.], 2018. p. 106–110. Citado na página 12.

MANNING, C.; SCHUTZE, H. **Foundations of statistical natural language processing**. [S.l.]: MIT press, 1999. Citado nas páginas 10, 13 e 14.

MEBANE, W. R.; KLAVER, J.; MILLER, B. A. P. Frauds, strategies and complaints in germany. University of Michigan, 2016. Citado na página 12.

MEHR, H.; ASH, H.; FELLOW, D. Artificial intelligence for citizen services and government. **Ash Cent. Democr. Gov. Innov. Harvard Kennedy Sch., no. August**, p. 1–12, 2017. Citado na página 11.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citado na página 14.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, John Benjamins, v. 30, n. 1, p. 3–26, 2007. Citado na página 15.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.;

PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado na página 25.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543. Citado na página 14.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. **arXiv preprint arXiv:1802.05365**, 2018. Citado na página 14.

ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697. Citado na página 19.

Tjandra, S.; Warsito, A. A. P.; Sugiono, J. P. Determining citizen complaints to the appropriate government departments using knn algorithm. In: **2015 13th International Conference on ICT and Knowledge Engineering (ICT Knowledge Engineering 2015)**. [S.l.: s.n.], 2015. p. 1–4. ISSN 2157-099X. Citado na página 12.

GLOSSÁRIO

Feature: Uma variável ou característica que faz parte de uma observação ou registro.

Outlier: Em estatística, outlier, valor aberrante ou valor atípico, é uma observação que apresenta um grande afastamento das demais da série, ou que é inconsistente.

Pipeline: Na ciência de dados *pipeline* refere-se a uma série de etapas necessárias para o completo tratamento de dados.

CGU

Controladoria-Geral da União