Universidade Estadual de Londrina

JANAÍNA IGNÁCIO DE MORAIS

CLASSIFICAÇÃO MULTIRRÓTULO DE NOTÍCIAS CONSIDERANDO SUA LEGITIMIDADE E OBJETIVIDADE

Londrina
2020

JANAÍNA IGNÁCIO DE MORAIS

CLASSIFICAÇÃO MULTIRRÓTULO DE NOTÍCIAS CONSIDERANDO SUA LEGITIMIDADE E OBJETIVIDADE

Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Sylvio Barbon Jr.

Londrina
2020

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Morais, Janaina.
Classificação Multirrótulo de Notícias Considerando sua Legitimidade e Objetividade / Janaina Morais. - Londrina, 2020.
79 f. : il.

Orientador: Sylvio Barbon Júnior.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2020.
Inclui bibliografia.

1. Fake News - Tese. 2. Multirrótulo - Tese. 3. Mineração de Texto - Tese. 4. Notícias Falsas - Tese. I. Barbon Júnior, Sylvio . II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519

JANAÍNA IGNÁCIO DE MORAIS

CLASSIFICAÇÃO MULTIRRÓTULO DE NOTÍCIAS CONSIDERANDO SUA LEGITIMIDADE E OBJETIVIDADE

Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

_____
Orientador: Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina – UEL

_____
Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estatual de Londrina – UEL

_____
Prof. Dr. André Azevedo da Fonseca
Universidade Estatual de Londrina - UEL

Londrina, 14 de agosto de 2020.

Dedico este trabalho a minha avó Maria, que sempre acreditou e me incentivou em todas as minhas escolhas e principalmente por ser o meu maior exemplo de força e determinação.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, pelo suporte e por se mostrar sempre presente em minha vida auxiliando em minhas escolhas e objetivos, e a minha família pelo suporte emocional que sempre me proporcionou em todas as etapas da minha vida, em especial minha mãe Maria Cristina e minha avó Maria.

Em segundo lugar, gostaria de agradecer ao meu orientador Prof. Dr. Sylvio Barbon Jr. pela oportunidade concedida e por todo o auxilio durante este mestrado que me proporcionou uma mudança significativa de visão de mundo, principalmente sobre a área da pesquisa, e que farão total diferença em meus próximos desafios. Também quero agradecer aos meus colegas do Remid, principalmente ao Hugo Queiroz Abonizio pela parceria nos artigos durante o mestrado e as dicas concedidas que foram extremamente relevantes durante esta jornada. Ao Everton José Santana, por tornar a vivência do laboratório leve e divertida com boas conversas e dicas ao longo desse período.

Gostaria de agradecer a CAPES pelo apoio financeiro durante o decorrer da minha pesquisa e a alguns professores da UEL que convivi durante este período, que me inspiram com sua disciplina e amor a profissão. Agradeço também aos amigos que fiz fora do laboratório, Mariani Margarida Bento e Camila Sonoda Gomes, que demonstraram o valor de uma boa amizade e me apresentaram outros nichos dentro da universidade, o que tornou esta experiência mais rica e abrangente. Por fim, ao Lucas Busatta Galhardi, que me ensinou que na vida é preciso ter equilibro em todos os aspectos e principalmente ter autoconfiança, proporcionando momentos de leveza emocional.

"Besides the world isn't split into good people and Death Eaters. We've all got both light and dark inside of us. What matters is the part we choose to act on." – J.K. Rowling, Order of the Phoenix.

MORAIS, J. I.. **Classificação Multirrótulo de Notícias Considerando sua Legitimidade e Objetividade**. 2020. 81f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2020.

RESUMO

Atualmente, a disseminação de notícias falsas tem aumentado significativamente em relação à classe política e aos membros da sociedade em geral, aumentando as preocupações sobre a potencial propagação de desinformação, de forma que vem aparecendo no centro dos debates sobre os resultados das eleições em todo o mundo. Por outro lado, as notícias falsas e satíricas tem um propósito divertido, mas geralmente são colocadas por engano no mesmo contexto das notícias falsas e objetivas. Nesta dissertação, abordamos as diferenças entre objetividade e legitimidade dos documentos noticiosos, tratando cada artigo como tendo duas classes conceituais: objetiva/satírica e legítima/falsa. Assim, propomos um Sistema de Apoio à Decisão (*Decision Support System* — DSS) baseado em um *pipeline* de Mineração de Texto e um conjunto de novas características (*features*) textuais que usam métodos multirrótulo para classificar artigos de notícias nesses dois domínios. Para validar a abordagem, um conjunto de métodos multirrótulo foi avaliado com uma combinação de diferentes classificadores de base e, em seguida, comparado com uma abordagem multi-classe. Os resultados apresentaram o nosso DSS como adequado (*F1-score* de 0,81) ao abordar o cenário de notícias enganosas da perspectiva desafiadora da modelagem multirrótulo, superando os métodos multi-classe (*F1-score* de 0,68) em um conjunto de dados de notícias com dados reais coletado de vários portais de notícias. Além disso, foi analisado como os grupos de *features* estilométricas podem influenciar nos resultados, com o objetivo de descobrir se um determinado grupo possui mais relevância que outros. Como resultado, observou-se que o grupo de *features* de complexidade obteve maior destaque dentre os grupos analisados, o que demonstrou que a diferença de complexidade textual entre as classes conceituais se destacou como sendo um dos maiores parâmetros auxiliares na decisão do modelo.

Palavras-chave: Notícias Falsas. Sistema de Apoio à Decisão. Mineração de Texto. Multirrótulo.

MORAIS, J. I.. **A Multi-label News Classification Considering your Legitimacy and Objectivity**. 2020. 81p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina, 2020.

ABSTRACT

Currently, the widespread of fake news has raised on the political class and society members in general, increasing concerns about the potential of misinformation that can be propagated, appearing at the center of the debate about election results around the world. On the other hand, satirical news has an entertaining purpose and are mistakenly put on the same context of objective fake news. In this work, we address the differences between objectivity and legitimacy of news documents, handing each article as having two conceptual classes: objective/satirical and legitimate/fake. Thus, we propose a Decision Support System (DSS) based on a Text Mining (TM) pipeline and a set of novel textual features that uses multi-label methods for classifying news articles on those two domains. To validate the approach, a set of multi-label methods was evaluated with a combination of different base classifiers and then compared to a multi-class approach. The results reported our DSS as proper (0.81 f1-score) in addressing the scenario of misleading news from challenging the perspective of multi-label modeling, outperforming the multi-class methods (0.68 f1-score) over a real-life news dataset collected from several portals of news. Moreover, it was analyzed how stylometric features groups influenced the result, which showed that complexity features have more relevance than others. Also, we analyzed how the group of stylometric features can influence the results in order to find out if a specific group is more relevant than others. As a result, it was observed that complexity features got more influence among the analyzed groups, which demonstrated that the difference in textual complexity between the conceptual classes stood out as one of the best auxiliary parameters in the model decision.

Keywords: Fake News. Decision Support System. Text Mining. Multi-label.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	*Aprendizado de Máquina*
BR	*Binary Relevance*
CNN	*Convolutional Neural Networks*
CD	*Critical Distance*
DM	*Data Mining*
DT	*Decision Tree*
DSS	*Decision Support System*
FN	*False Negative*
FP	*False Positive*
GRU	*Gated Recurrent Units*
HAN	*Hierarchical Attention Network*
IA	Inteligência Artificial
*k*NN	*k-Nearest Neighbor*
LP	*Label Powerset*
LR	*Logistic Regression*
MC	*Multi-class*
ML	*Multi-label*
ML-*k*NN	*Multi-label k-Nearest Neighbor*
MT	Mineração de Texto
NB	*Naive Bayes*
NLTK	*Natural Language Toolkit*
OOV	*Out-of-vocabulary*
PCA	*Principal Component Analisys*
PLN	Processamento de Linguagem Natural

POS	*Part-of-speech*
RF	*Random Forest*
RNN	*Recurrent Neural Network*
SVM	*Support Vector Machine*
TN	*True Negative*
TP	*True Positive*
TTR	*Type-token ratio*

SUMÁRIO

1 INTRODUÇÃO

As redes sociais provocaram transformações substantivas na maneira pela qual os cidadãos consomem, interpretam e processam as notícias. Durante o século XX, as informações que circulavam nas mídias de massa eram previamente selecionadas pelos editores dos veículos de imprensa que, em geral, atendiam aos critérios clássicos do jornalismo – tal como atualidade, relevância social e veracidade [2].

Contudo, a credibilidade da imprensa se construiu precisamente a partir do compromisso de um conjunto de empresas de mídia com as práticas do jornalismo profissional. Deste modo, leitores e espectadores se acostumaram a delegar à imprensa a tarefa de processar o volume monumental de informações sobre a realidade e apresentá-las em um subconjunto assimilável de notícias periódicas [3].

No entanto, no contexto do acesso à informação através das redes sociais, a mediação das notícias, sobretudo na etapa final do consumo, deixou de ser realizada por jornalistas e editores profissionais. Portanto, em vez de jornalistas profissionais, os algoritmos se tornaram os principais agentes encarregados de selecionar e distribuir as informações que chegam de forma individualizada aos consumidores [4].

Essa dinâmica se aliou à proliferação de sites amadores que obtêm alta lucratividade com o tráfego online, graças ao acesso facilitado aos programas de anúncios digitais, como o *Google Adsense*. Impulsionados pelas redes sociais e potencializados pelas análises de dados que indicam os gostos, os preconceitos e as predisposições dos usuários, uma infinidade de sites se dedica a produzir conteúdos de fácil e rápida viralização, sem quaisquer compromissos com questões alheias à própria lucratividade. Como consequência, os usuários passam a receber e consumir um volume maciço de informação de procedência duvidosa. Segundo a pesquisa divulgada em setembro de 2018 pela Ipsos[1], cerca de 62% da população brasileira acredita nas notícias falsas, o que mostra o quão grave é o problema e o quão expressivo é em nossa sociedade.

Estudos no campo da *media litteracy*[2] [5] buscam há anos educar o público para as particularidades das mensagens nos meios de comunicação. A complexidade crescente do ecossistema midiático exige a formação de leitores e espectadores capazes de compreender a diversidade de fatores que condicionam a produção de informação. Mas para firmar a crítica dos conteúdos, é imprescindível formular a reflexão sobre as linguagens das mídias. Sem essa pedagogia, usuários dispõem de menos recursos para discernir entre a linguagem enganosa de um site de notícias falsas (*fake news*) e a linguagem irônica de um site de

[1] https://www.ipsos.com/pt-br/global-advisor-fake-news

[2] Uma abordagem da educação do século XXI, que fornece uma estrutura para acesso, análise, avaliação, consumo e criação de mídia.

humor, por exemplo.

O Sensacionalista[3] é um dos sites de humor mais populares do Brasil. Através de uma paródia explícita com a linguagem jornalística, os redatores inventam notícias cômicas envolvendo personalidades reais e inspiram críticas sociais através da ironia. Por isso, o Sensacionalista – que ironiza até no nome – não se caracteriza propriamente como um site de *fake news*, pois o objetivo declarado é o humor.

Contudo, leitores desatentos – e sem formação em *media litteracy* – frequentemente se confundem ao interpretar os textos humorísticos como reportagens verdadeiras. Como a ironia é um recurso sofisticado de linguagem, nem sempre a piada é evidente. Além disso, a habilidade dos redatores na construção da paródia em forma de notícia busca precisamente extrair o humor através da alegoria. As diferenças são propositadamente sutis.

Fake news, ao contrário, é melhor definida como "*news articles that are intentionally and verifiably false, and could mislead readers*" [6]. O debate internacional mais intenso sobre o tema foi travado sobretudo após a eleição de Donald Trump. Ainda, de acordo com uma pesquisa divulgada em outubro de 2018 pelo Datafolha[4], a maioria das notícias propagadas nas últimas eleições brasileiras vieram de redes sociais como o *Facebook*.

A ironia presente nos textos possui características muito particulares, nas quais podemos visualizar sentimentos negativos ou opostos em certas afirmações, o que geralmente apresentam um significado implícito dentro do contexto geral do texto. No que diz respeito à política, além da ironia, podemos associar a análise de sentimentos à decepção do público contra um partido em particular, o que influencia os resultados das pesquisas [7].

Com isso, uma alternativa para tentar mitigar este problema é o uso do conceito de Mineração de Texto (MT), que é uma subárea da Inteligência Artificial (IA), sendo capaz de ler documentos textuais e analisá-los, onde torna-se possível a automatização e determinação se uma notícia é ou não falsa.

Embora existam vários trabalhos de MT no estado da arte que abordam a possibilidade de uma notícia ser falsa ou não com base em seu conteúdo textual, acreditamos que uma única notícia possa pertencer a várias classes conceituais. Portanto, um desafio adicional é colocado ao MT tradicional no que diz respeito a evolução para uma classificação textual multirrótulo (*multi-label* - ML).

A classificação comum *single-label* (único rótulo) aborda a indução de um modelo a partir de um conjunto de exemplos associado a um único rótulo l, a partir de um conjunto

[3] https://www.sensacionalista.com.br/

[4] http://datafolha.folha.uol.com.br/opiniaopublica/2018/10/1983765-24-dos-e eitores-usam-whatsapp-para-compartilhar-conteudo-eleitoral.shtml

de rótulos separados L, $|L| > 1$. Se $|L| = 2$, temos um problema de classificação binária. Como alternativa, se $|L| > 2$ é um cenário de classificação multi-classe. Finalmente, na classificação multirrótulo, os exemplos estão relacionados a um conjunto de rótulos L, onde o conjunto de classes $Y \subseteq L$.

Nesse contexto de notícias enganosas como um problema de várias classes, temos $|Y| = 2$ com classes conceituais de y_1 = "falso/legítimo" e y_2 = "satírico/objetivo". Os rótulos (*labels*) são L = {objetivo-legítimo, objetivo-falso, satírico-legítimo, satírico-falso}. As definições formais das classes conceituais usadas neste trabalho são extraídas de Shu et al. [2] :

Definição 1 (Notícias falsas)**.** Uma notícia é considerada falsa quando seu conteúdo é comprovadamente falso e sua intenção é voltada para enganar o leitor.

Definição 2 (Notícias legítimas)**.** Uma notícia é considerada legítima quando seu conteúdo é comprovadamente verdadeiro e sua intenção consiste em transmitir informações autênticas ao leitor.

Definição 3 (Notícias satíricas)**.** Uma notícia é considerada satírica quando sua intenção é voltada para o entretenimento do consumidor onde a informação é abordada de forma irônica de modo a criticar ou zombar do assunto em questão.

Definição 4 (Notícias objetivas)**.** Uma notícia é considerada objetiva quando sua intenção é expressar a informação de forma clara e direta.

Portanto, a proposta desta dissertação tem como objetivo propor e validar um *pipeline* de mineração de texto para a detecção de notícias falsas baseadas em seu conteúdo, utilizando classificação multirrótulo de notícias incorporadas em um sistema de apoio a decisão de legitimidade de notícias, de modo a não depender de formatos específicos provenientes de portais de notícias. As classes conceituais estão relacionadas à sua legitimidade (falso/legítimo) ou sua objetividade (objetivo/satírico) de modo a demonstrar que uma notícia não é necessariamente totalmente verdadeira ou totalmente falsa, onde podem ser notadas características de sarcasmo em notícias verdadeiras e objetividade em notícias falsas, por exemplo. Para isso, escolhemos trabalhar com algoritmos clássicos do estado da arte multi-classe comparando-os com uma abordagem multirrótulo de modo a demonstrar a contribuição de nosso Sistema de Apoio à Decisão (*Decision Support System* — DSS) para o estado da arte.

As contribuições secundárias deste trabalho são:

1. Apresentar nosso conjunto de dados (*dataset*) de notícias com várias etiquetas (*multi-labelled*) extraídas de sites reais;

2. Identificar o melhor algoritmo de Aprendizado de Máquina (AM) e características (*features*) textuais em nosso cenário de notícias multirrótulo;

3. Propor novas *features* textuais e avaliar o impacto obtido na classificação dos resultados.

O restante dessa dissertação está organizada da seguinte forma. O Capítulo 2 apresenta os conceitos e definições básicas em que o trabalho se baseia. O Capítulo 3 mostra pesquisas relacionadas no estado da arte, demonstrando uma visão geral do problema. O Capítulo 4 apresenta a abordagem proposta, mostrando o *pipeline* e a extração de *features* utilizadas nesta pesquisa. O Capítulo 5 apresenta uma descrição dos métodos e a avaliação do modelo. O Capítulo 6 apresenta os resultados e discussões do nosso problema. Por último, o Capítulo 7 apresenta as considerações finais do trabalho e os próximos passos a serem realizados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os conceitos teóricos necessários para o entendimento do restante deste trabalho, onde a Seção 2.1 introduz o conceito de Aprendizado de Máquina, a Seção 2.2 introduz o conceito de classificação multi-classe e alguns dos métodos usados neste trabalho. A Seção 2.3 explica o conceito de problemas multirrótulo e sua diferença em relação aos problemas multi-classe. Por fim, a Seção 2.4 introduz o Processamento de Linguagem Natural, explicando sua importância e como a máquina entende e lida com a linguagem natural.

2.1 Aprendizado de Máquina

Tradicionalmente um programa de computador é constituído pelo conceito de programação estruturada, onde um conjunto de instruções definem exatamente o que o computador deve fazer para resolver determinado problema, sendo suscetível a erros com responsabilidade inteiramente humana [8, 1]. No entanto, a programação estruturada não tem capacidade de aprender ou reconhecer padrões a partir de exemplos, o que a torna limitada para a maioria dos problemas da realidade.

Com isso, a partir da década de 1970, tornou-se popular o uso de técnicas baseadas em Inteligência Artificial (IA) para a resolução de problemas complexos ou específicos através de sistemas especialistas, que são programas baseados em tomadas de decisões com base no conhecimento de especialistas humanos sobre determinado problema. No entanto, esse processo possui inúmeras limitações, além da dependência do conhecimento humano. Além disso, os problemas da atualidade lidam com um volume muito maior de dados, e a necessidade da resolução de problemas complexos do dia-a-dia sem a necessidade de interferência humana tornou-se cada vez mais necessária, e com isso, surgiu o conceito de Aprendizado de Máquina (AM) [1].

O Aprendizado de Máquina é um campo de pesquisa em IA que tem como objetivo descobrir *insights* e padrões a partir de experiências passadas, analisando e predizendo situações futuras sem precisar de instruções detalhadas, dando ao computador a habilidade de aprender e se adequar ao problema sem a necessidade de intervenção humana [8, 9].

No entanto, para que isso seja possível, é necessário o uso de algoritmos que sejam treinados. Para o processo de treinamento é utilizado um conjunto de dados onde estão disponíveis amostras de exemplos que mostram à máquina como resolver um determinado problema. A quantidade de amostras para o treinamento é extremamente importante, pois com um intervalo maior de amostras, a máquina aprende uma quantidade maior de possíveis casos e pode fazer mais associações para classificar o problema, pois reconhece

uma gama maior de possibilidades.

Além disso, as amostras podem apresentar diferentes formas e atributos, e para que possam ser utilizadas é necessário o uso de um modelo lógico e consistente de características (*features*) que possa descrever melhor sua essência para os computadores. Com isso, após identificar as *features*, o algoritmo buscará encontrar relações entre as *features* dos dados da amostra e possíveis rótulos (*labels*).

Ainda, um algoritmo de AM precisa saber lidar com dados imperfeitos, pois o desempenho do algoritmo pode ser afetado pelo estado do conjunto de dados, que pode apresentar problemas como: presença de ruídos, dados inconsistentes ou duplicados, número baixo de amostras, número muito baixo ou muito alto de atributos, etc. E para isso, são usadas técnicas de pré-processamento destes dados de modo a minimizar possíveis problemas de desempenho desta natureza.

Os algoritmos de AM podem ser agrupados em quatro categorias que englobam diferentes gamas de problemas existentes no estado da arte, são elas: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço [9, 10, 11].

- **Aprendizado supervisionado:** é quando o algoritmo trabalha com rótulos de saída conhecida, ou seja, é quando o algoritmo lida com um conjunto de dados que possui amostras rotuladas com as respostas corretas para auxiliar no treinamento do algoritmo, onde é possível fazer a associação de conjuntos de padrões com a reposta esperada. No entanto, esta abordagem não decora simplesmente as amostras para comparação, e sim, usa o subconjunto para generalizar o conhecimento sobre o problema para a aprendizagem.

- **Aprendizado não supervisionado:** consiste em quando não há um valor específico determinado como saída. Os dados que descrevem o problema são conhecidos, porém, cabe à técnica de AM agrupar dados similares, inferir possíveis comportamentos, extrair padrões, etc.

- **Aprendizado semi-supervisionado:** é quando existem aspectos da aprendizagem supervisionada e não supervisionada. Neste caso, parte da saída esperada é conhecida e parte não. Geralmente a informação conhecida auxilia na caracterização de comportamentos similares provenientes de conjuntos conhecidos, e também permite uma melhora significativa na acurácia, pois há a possibilidade de mescla de parte do conjunto de dados com e sem rótulos.

- **Aprendizado por reforço:** de todas as categorias descritas, esta é a única que não possui conjunto de treinamento com ou sem valor determinado como saída. O aprendizado por reforço atua baseado no conceito de remuneração, onde durante o

processo de treinamento, dependendo das ações executadas, há uma "recompensa" como um incentivo ou um "castigo" como advertência até atingir o seu objetivo.

Além disso, o AM também pode ser dividido em três categorias diferentes, que são usadas de acordo com a finalidade do problema e a saída desejada [8, 12], são elas:

- **Classificação:** é considerada uma subcategoria do aprendizado supervisionado. Geralmente é usada quando as previsões são de natureza distinta, como, por exemplo "verdadeiro" ou "falso", onde o algoritmo atribui cada nova amostra como pertencente a uma das classes disponíveis. A Figura 1 ilustra um problema de classificação, onde duas classes representam os possíveis resultados de um exame médico.

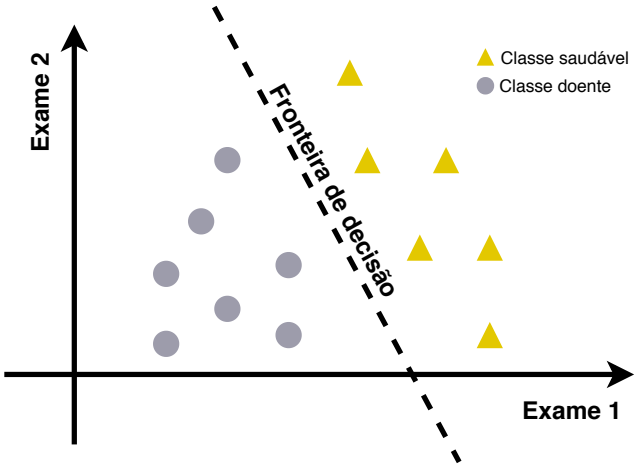

Figura 1 – Gráfico ilustrativo de uma tarefa de classificação [1].

- **Regressão:** também é considerada uma subcategoria do aprendizado supervisionado. É usada quando o valor que está sendo previsto difere do conceito binário de "verdadeiro" ou "falso", ou seja, quando o resultado desejado é uma variável contínua. Este tipo de sistema é usado para responder perguntas, como, por exemplo, a quantidade de um determinado produto. A Figura 2 ilustra um problema de regressão, onde se espera o retorno de uma função aproximada que relacione a vazão de água de um dado rio há um determinado ano.

- **Clusterização:** já a clusterização está relacionada ao aprendizado não supervisionado, onde não há rótulos específicos como saída no treinamento. Neste caso, as amostras são agrupadas, onde cada nova amostra pode ser categorizada como pertencente a um novo grupo, do qual não tenha sido descoberto ainda. A Figura 3 ilustra um exemplo de separação por *clusters*.

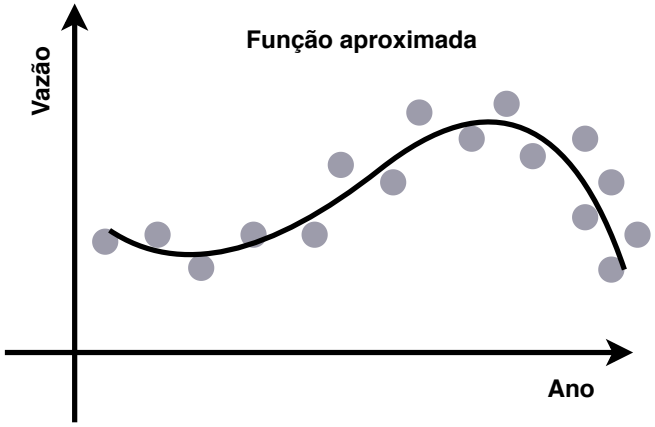

Figura 2 – Gráfico ilustrativo de uma tarefa de regressão [1].

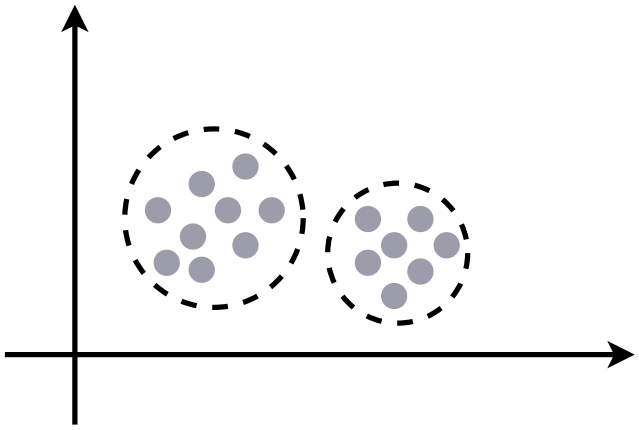

Figura 3 – Gráfico ilustrativo de um conjunto de dados separados em *clusters*.

2.2 Classificação Multi-classe

Os problemas de classificação, em sua forma básica, trabalham com problemas binários, onde uma determinada amostra pode fazer parte de uma classe ou de outra. Os problemas de classificação multi-classe (*multi-class* — MC) diferem desse conceito base, pois trabalham com mais de duas classes, assim, é possível trabalhar com problemas mais complexos, como, por exemplo, ao invés de descobrir se uma pessoa gosta ou não de livros, é possível descobrir qual o gênero literário de determinado livro, o que engloba uma gama maior de possibilidades [13].

Para a resolução desses problemas geralmente são usados algoritmos que propõem reduzir a complexidade de um problema multi-classe a um problema binário. Alguns algoritmos de natureza binária podem ser utilizados naturalmente em problemas multi-classe, enquanto outros necessitam de ajustes, onde o problema é reduzido tornando-se binário ou decomposto em subproblemas binários para então ser possível sua aplicação em algoritmos de classificação binária. Nas próximas seções serão apresentados os algoritmos

multi-classe utilizados neste trabalho.

2.2.1 k-Nearest Neighbor

O k-Vizinho Mais Próximo (k-Nearest Neighbor — kNN) é um método baseado em distâncias e é uma das técnicas mais simples e conhecidas de aprendizado supervisionado. Esta técnica pode ser aplicada em problemas de classificação e regressão e é considerada uma técnica preguiçosa (*lazy*) e bastante efetiva na área de Aprendizado de Máquina [8]. A ideia principal do kNN é determinar qual rótulo de classificação de uma amostra pertence baseando-se em suas amostras vizinhas, e para isso, o algoritmo relaciona cada um dos exemplos da etapa de treinamento a um ponto em um espaço n-dimensional, de modo que n seja o número de atributos de entrada que descrevem o conjunto de dados [1].

Quando faz predições, o kNN acessa o conjunto de dados de treinamento inteiro, isso significa que não é necessário que haja aprendizado, pois todos os dados são armazenados em memória permanecendo acessíveis para consulta, o que justifica o uso do termo *lazy*. Com isso, a cada nova amostra adicionada, o kNN pesquisa por todos os dados treinados para as k entradas mais próximas (vizinhos próximos) que são ordenados da menor para a maior distância em relação à nova amostra como mostram a Figura 4 e a Tabela 1.

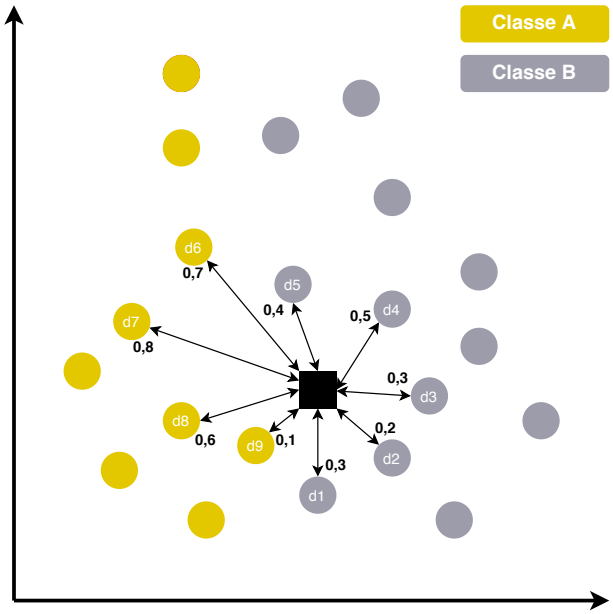

Figura 4 – Exemplo do processo de execução do algoritmo kNN, onde a distância de cada vizinho mais próximo é calculada e ordenada.

A partir desta ordenação, serão considerados apenas os vizinhos referentes à quantidade determinada por k, por exemplo, quando $k = 1$ apenas o valor do vizinho mais próximo (primeiro da lista) será considerado, para $k = 2$ o primeiro e o segundo valor da

Tabela 1 – Tabela representativa da ordenação dos vizinhos mais próximos referente a Figura 4

Classes Conceituais		Distância	Classe
1	d9	0,1	A
2	d2	0,2	B
3	d1	0,3	B
4	d3	0,3	B
5	d5	0,4	B
6	d4	0,5	B
7	d8	0,6	A
8	d6	0,7	A
9	d7	0,8	A

lista serão considerados, para $k = \infty$ o conjunto de dados completo será considerado, etc. [14].

A Figura 5 exemplifica uma classificação com duas classes e com $k = 5$. Neste exemplo é possível perceber que há uma nova amostra (representada por um quadrado preto), e amostras de treinamento da classe A e classe B, respectivamente, representadas pelas cores amarelo e cinza. Como o k deste exemplo é 5, então apenas as 5 amostras mais próximas (vizinhos mais próximos) serão consideradas para a classificação desta nova amostra. Com isso, das cinco amostras de treinamento consideradas, três delas pertencem à classe A e duas à classe B, e assim, a nova amostra será classificada como pertencente à classe A, pois possui uma quantidade maior de amostras vizinhas pertencentes a esta classe.

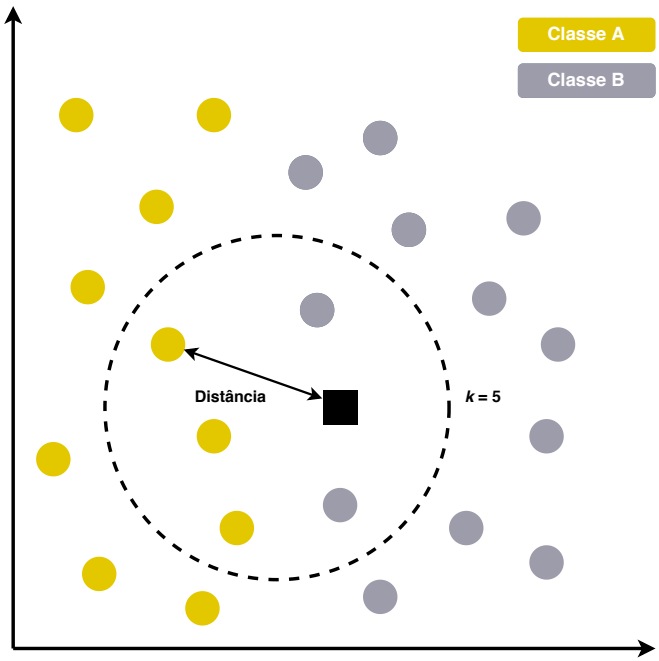

Figura 5 – Gráfico ilustrativo do algoritmo kNN com $k = 5$.

Porém, é importante salientar que o valor k escolhido influencia na acurácia da classificação, pois uma amostra pode ser classificada como pertencente à outra classe baseada na escolha do valor k a ser utilizado, portanto, a escolha do valor de k é um dos passos mais importantes para o uso do kNN [1]. Além disso, é extremamente importante escolher a métrica de distância que melhor se encaixa ao problema trabalhado. Normalmente, a maioria dos problemas utiliza como métrica a distância *Euclidiana* que trabalha com valores reais e é representada por:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \tag{2.1}$$

Além da distância Euclidiana também são bastante utilizadas a distância *Manhattan* 2.2, que calcula a distância entre vetores reais através da soma de sua diferença absoluta e é bastante utilizada quando as variáveis de entrada do problema não possuem um padrão, como, por exemplo: nome, idade, sexo, etc. E a distância *Minkowski* 2.3, que é a generalização das distâncias Euclidiana e Manhattan.

$$d(x, y) = |x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n| \tag{2.2}$$

$$d(x, y) = (|x_1 - y_1|^q + |x_2 - y_2|^q + ... + |x_n - y_n|^q)^{\frac{1}{q}} \tag{2.3}$$

Além da escolha do k e da métrica de distância, o tamanho do conjunto de dados também influencia diretamente no desempenho do algoritmo, pois como o kNN é um algoritmo do tipo *lazy*, todo o conjunto de dados é usado na fase de predição para o cálculo da distância escolhida, e com isso, a predição tende a ser custosa e demorada se o conjunto de dados for muito grande [1].

Para problemas de regressão, o kNN conta com pequenas modificações em seu algoritmo, onde ao invés de calcular a distância tomando como parâmetro a classe que aparece com mais frequência entre os k vizinhos mais próximos, é utilizada a média dos valores dessas instâncias [8].

2.2.2 Random Forest

O algoritmo de Floresta Aleatória (*Random Forest* — RF) é um algoritmo de aprendizagem supervisionada que cria várias árvores de decisão (*Decision Tree* — DT) e as combina formando uma espécie de floresta, como o próprio nome sugere [15]. Este método é relativamente novo comparado a outros, porém, é bastante vantajoso por ser simples de implementar e funcionar bem tanto em problemas de classificação quanto de regressão mesmo sem ajuste de hiperparâmetros.

O funcionamento da árvore de decisão é similar ao de um fluxograma, no qual todo o conjunto de treinamento é disposto em uma árvore onde cada nó ocupa determinada posição baseada em condicionamento como mostra a Figura 6. Este método não é aconselhável para uso em conjuntos de dados muito grandes, pois tende a perder desempenho preditivo, o que pode gerar *overfitting*.

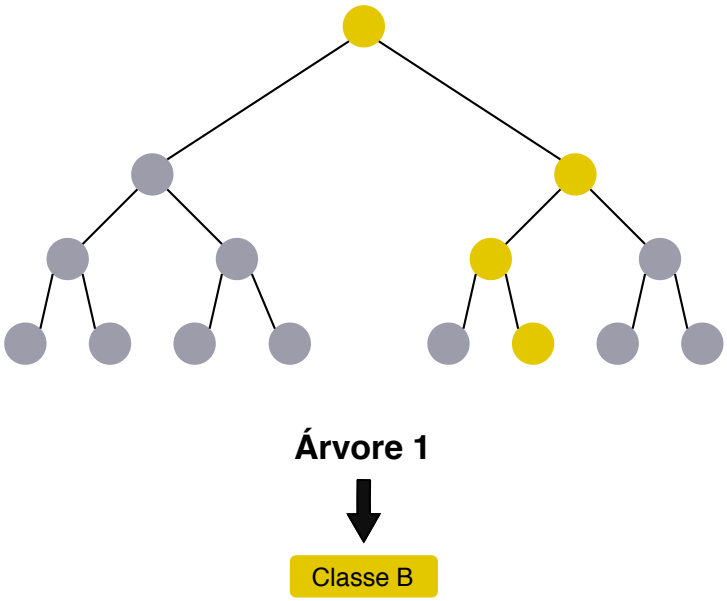

Figura 6 – Gráfico ilustrativo do algoritmo de Árvore de Decisão.

Já a RF não possui este problema, pois cada árvore da floresta é construída através de uma combinação (*ensemble*) de árvores de decisão, que em sua maioria são treinadas com o método de *bagging*, onde os classificadores são treinados de forma independente por diferentes conjuntos de treinamento [15]. Além disso, é importante salientar que as árvores contidas na RF não possuem as mesmas características entre si, pois no processo de construção de cada uma delas, a seleção de *features* de cada nó é feita de maneira aleatória o que proporciona maior diversidade entre as árvores do modelo como mostra o exemplo da Figura 7, e com isso, há uma melhora significativa em seu desempenho.

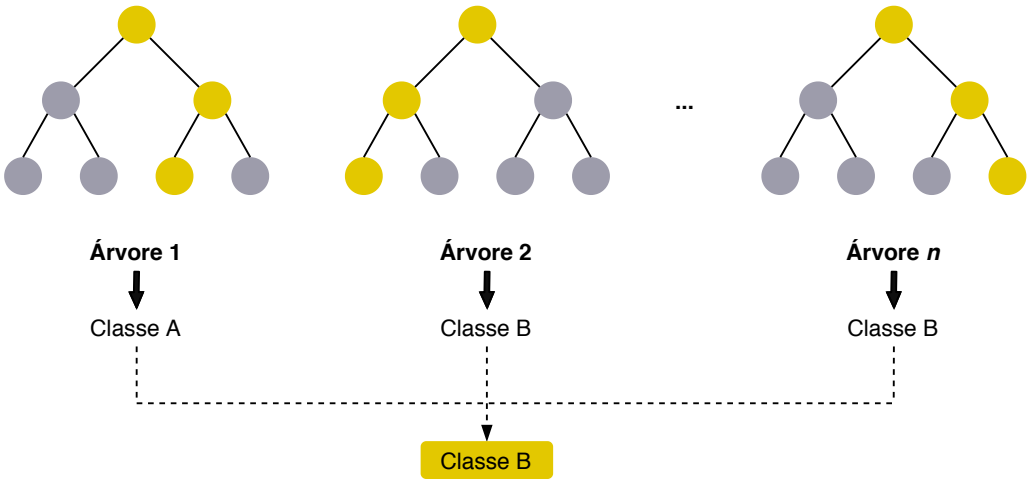

Figura 7 – Gráfico ilustrativo do algoritmo de Floresta Aleatória.

Em sua fase de predição, cada nova amostra é executada por todas as árvores da floresta, onde cada árvore retornará o seu voto referente a qual classe a amostra pertence. Com isso, a predição final da RF no processo de classificação será a classe que obtiver o maior número de votos dentre todas as árvores da floresta. Já em regressão, a predição final será a média dos valores retornados previstos por todas as árvores da floresta.

2.2.3 Support Vector Machine

A Máquina de Vetor de Suporte (*Support Vector Machine* — SVM) é um algoritmo de aprendizado supervisionado proposto em 1992 pelo russo Vladimir Vapnik [16], e consiste em encontrar um hiperplano (plano n-dimensional) ótimo a fim de separar diferentes classes de dados com a maior margem possível, sendo capaz de lidar com conjuntos de dados de alta dimensionalidade, além de ser eficiente em uso de memória devido ao emprego de vetores de suporte para a previsão [17].

Para encontrar o hiperplano ótimo, a SVM considera a distância entre os pontos mais próximos de cada classe em relação ao hiperplano. Estes pontos são chamados de vetores de suporte, como mostra a Figura 8. Então, é calculada a distância entre estes vetores de suporte e o hiperplano, conhecida como margem.

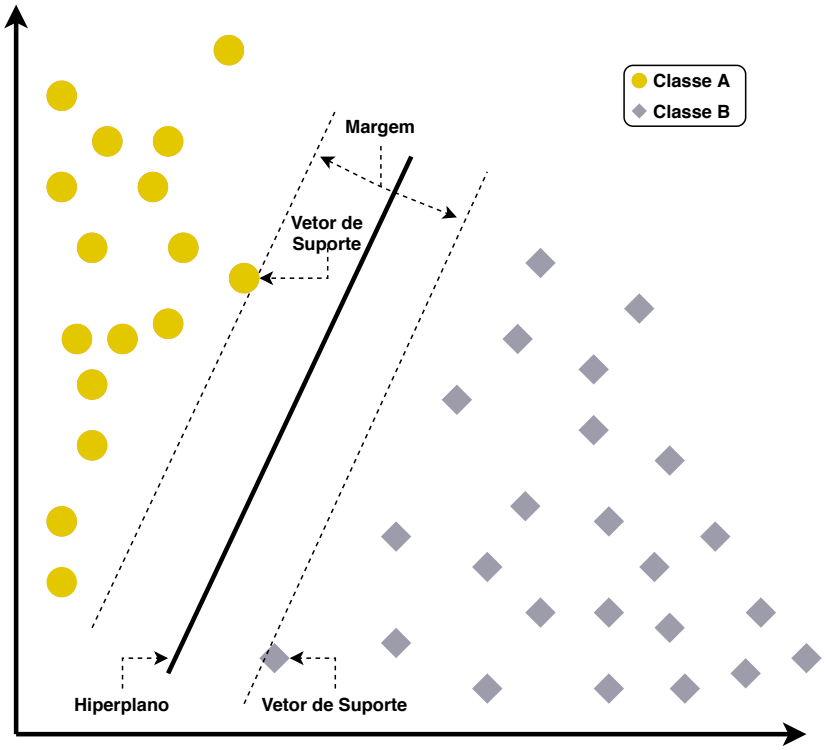

Figura 8 – Exemplo de hiperplano ótimo linear entre duas classes.

Com isso, a SVM espera tomar um limite de decisão onde a separação entre as classes seja a mais ampla possível para ambas as classes, o que garante uma separação generalizada. No entanto, dependendo do problema, não é possível determinar um hiperplano de separação exato entre as classes, sendo possível em alguns casos a conversão do espaço de transformação destes dados tornando possível a separação entre as classes, onde seus dados de entradas podem ser mapeados implicitamente em um espaço de maior dimensão [1, 18], como mostram as Figuras 9 e 10.

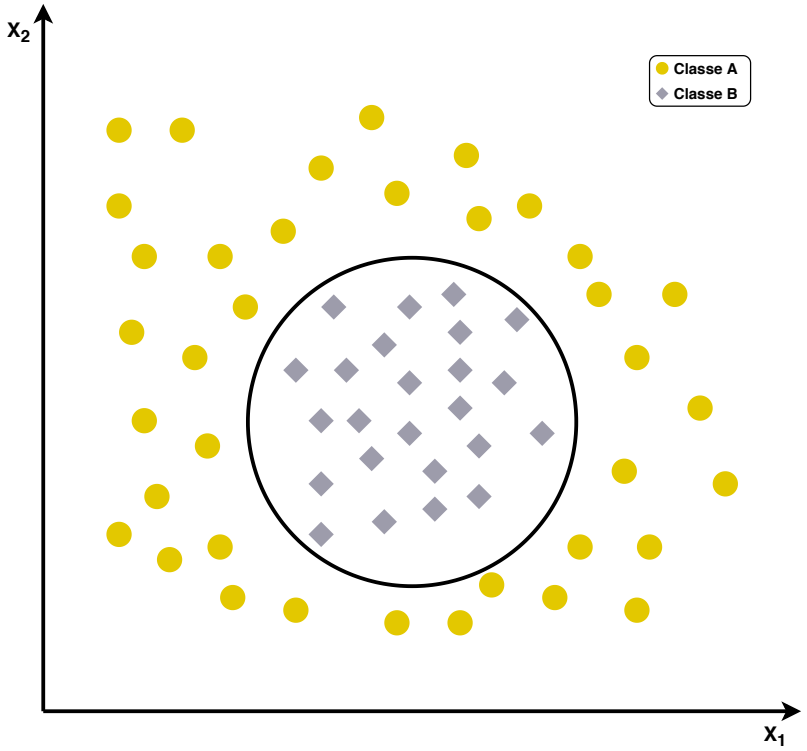

Figura 9 – Ilustração de um problema não linearmente separável em duas dimensões.

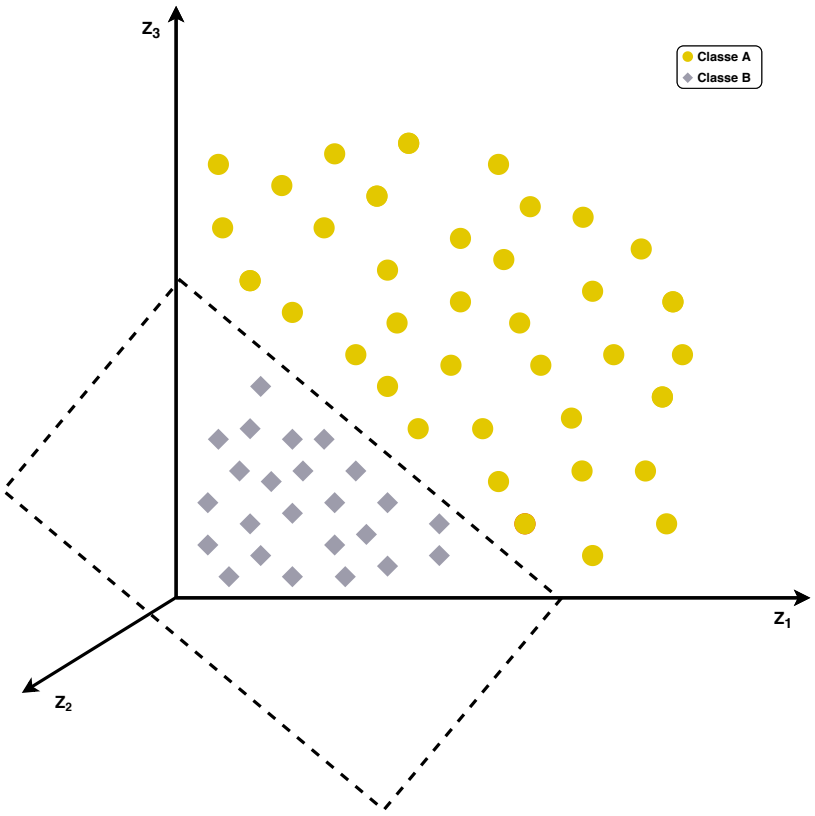

Figura 10 – Representação da resolução do problema da Figura 9 através da adição da terceira dimensão z [1].

No caso da Figura 10, uma nova dimensão representada pelo eixo z foi adicionada, e com isso, os dados se tornaram linearmente separáveis, sendo possível o uso da SVM. Por isso, as SVMs são consideradas métodos baseados em *kernel* [18, 19].

Além da classificação linear, a SVM também pode ser usada para classificação e regressão não linear através do uso do truque de *kernel* (núcleo), que permite que a SVM forme limites não lineares em problemas de qualquer dimensão [20]. Para isso, existem diversos kernels que podem ser usados nas SVMs, onde o kernel linear é usado para problemas lineares simples e os kernels polinomial, base radial e sigmoide para problemas não lineares mais complexos.

No entanto, é importante ressaltar que no caso de problemas de regressão, como o resultado é um número real, torna-se complicado prever as informações disponíveis, com isso é considerada uma margem de tolerância ϵ que é definida em aproximação à SVM. Além disso, a implementação deste algoritmo modificado conhecido como SVR é mais complicada que sua versão clássica, o que não será abordado neste trabalho.

2.3 Classificação Multirrótulo

Como visto anteriormente, a classificação multi-classe engloba um conjunto maior de possibilidades de escolha comparado à classificação binária, onde é possível classificar uma amostra em uma gama maior de classes disponíveis [21]. Os problemas de classificação multirrótulo (*multi-label* — ML) vão além desse conceito, pois uma amostra pode pertencer a mais de uma classe ao mesmo tempo. Por exemplo, uma amostra de um livro pode pertencer a uma determinada biblioteca e a um determinado gênero literário, dentro de n possibilidades de bibliotecas e gêneros possíveis. A Figura 11 exemplifica a diferença entre um problema de classificação multi-classe e um problema de classificação multirrótulo.

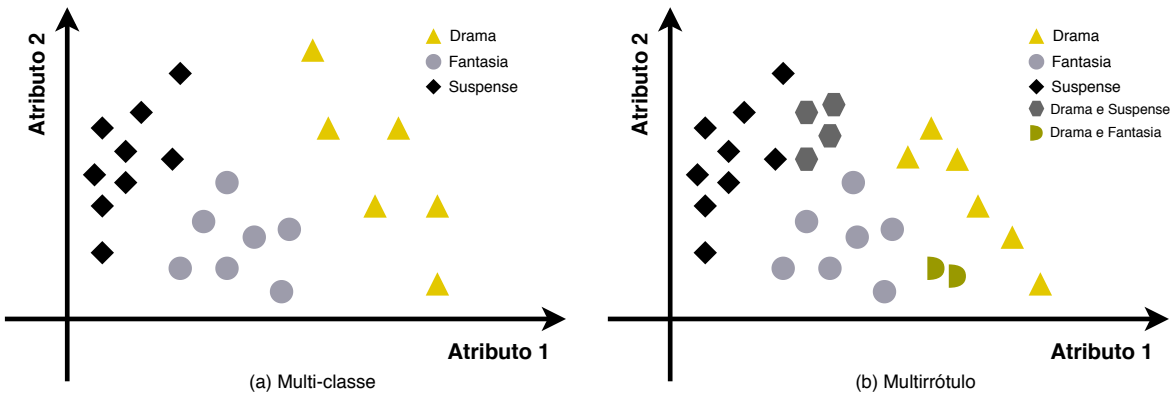

Figura 11 – Exemplificação de possíveis classes em problemas multi-classe e multirrótulo.

Os métodos de classificação multirrótulo geralmente estão associados a tarefas de classificação de textos, mas também estão presentes em áreas como Bioinformática e classificação de imagens [22, 23]. Esses métodos podem ser divididos em três grupos, são eles [24]:

- **Transformação de problema:** é uma abordagem que modifica o problema multirrótulo para se adaptar a qualquer algoritmo tradicional de AM, ou seja, consiste na transformação de uma tarefa multirrótulo para uma ou mais tarefas de um único rótulo (ou simples-rótulo) [22, 25].
- **Adaptação de algoritmos:** é uma abordagem que usa algoritmos específicos para lidar com problemas multirrótulo. Neste caso, os problemas multirrótulo não são alterados, e com isso, tendem a apresentar melhores resultados que a abordagem de transformação do problema [25, 1].
- **Classificação a partir de ranking:** nesta abordagem, a classificação multirrótulo é dada por meio da ordenação de rótulos por sua relevância, isto é, os dados são ordenados de forma a determinar quais rótulos são considerados mais relevantes para determinado exemplo ou quais exemplos são mais relevantes para determinado rótulo. Assim, a partir de um critério de corte são definidos quais rótulos serão atribuídos na predição das amostras [25].

Para este trabalho, utilizamos dois dos métodos clássicos mais utilizados de transformação de problema e um dos métodos mais conhecidos de adaptação de algoritmo, que serão descritos nas seções 2.3.1, 2.3.2 e 2.3.3.

2.3.1 Binary Relevance

O método de relevância binária (*Binary Relevance* — BR) é o método de transformação de problema mais popular, que decompõe um problema multirrótulo em vários subproblemas binários [22, 24]. Uma vantagem deste método é sua fácil adaptação com treinamentos que possuem ausência de rótulos ou informações incompletas. A Figura 12 exemplifica o funcionamento da transformação de relevância binária em um conjunto de dados.

O processo de classificação do modelo é bastante simples. Basicamente o conjunto de rótulos de uma nova amostra é predito através de uma combinação simples das saídas de cada um dos classificadores binários, gerando assim, uma combinação multirrótulo para cada amostra [26].

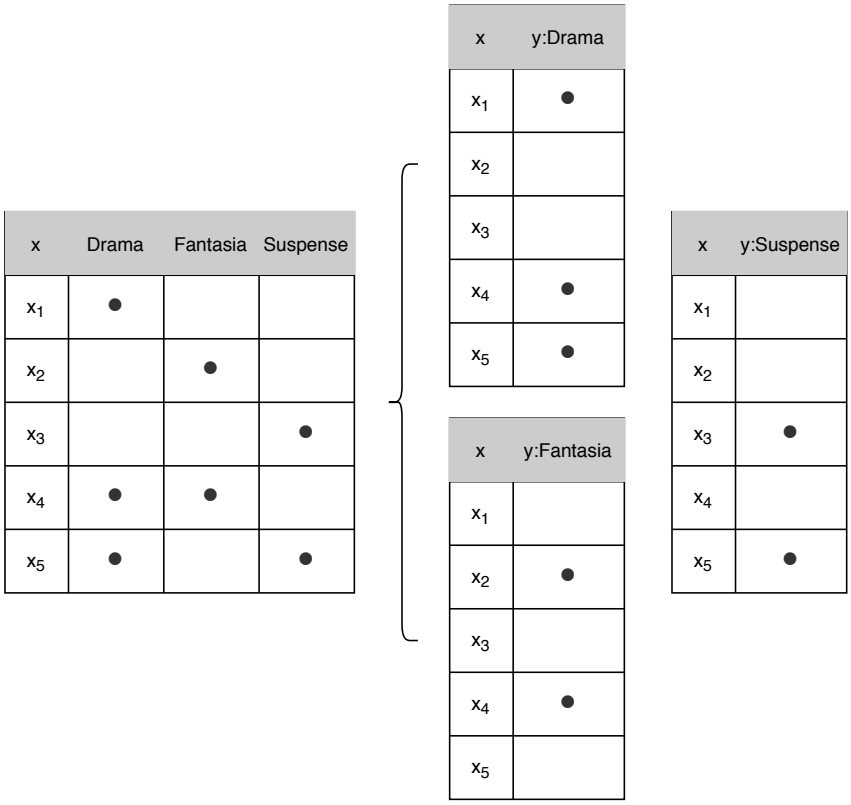

Figura 12 – Exemplificação de transformação de problema BR.

Além disso, como este método trabalha com subproblemas, estes subproblemas podem ser operados em paralelo, resultando na diminuição do tempo para a construção do modelo. Entretanto, é importante ressaltar que este método não considera a dependência de rótulos na construção do modelo, logo, cada classificador binário é construído de forma independente dos demais, o que pode resultar em um método com pouca capacidade de generalização [22, 24, 1].

2.3.2 Label Powerset

O *Label Powerset* (LP) é um método de transformação de problema considerado simples. Sua função é reduzir um problema multirrótulo para um problema monorrótulo do tipo multi-classe, ou seja, ele considera cada conjunto possível de rótulos no conjunto de treinamento como uma nova classe que possui um único rótulo que representa o conjunto de rótulos originais, transformando assim, um problema multirrótulo em multi-classe como mostra a Figura 13 [22, 26].

O processo de classificação do modelo é bastante simples, uma vez que o conjunto de dados original tenha passado pelo processo de transformação de problema, é possível classificar o problema através do uso de um algoritmo multi-classe comum, gerando como saída uma classe que corresponde ao conjunto de rótulos existente no problema original [26].

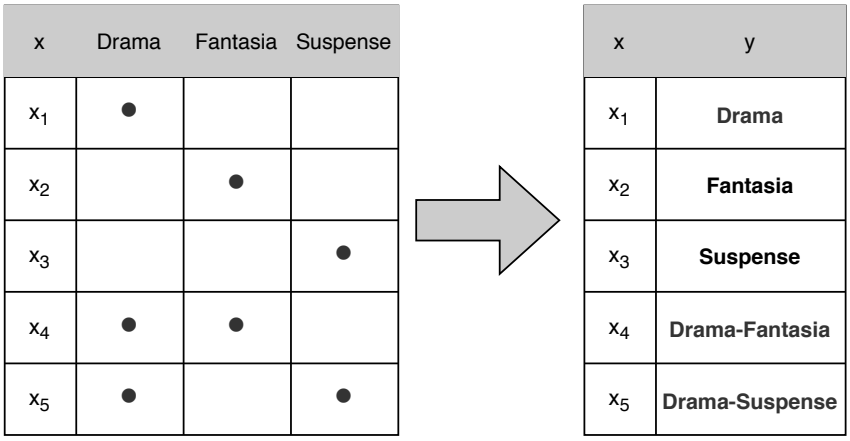

Figura 13 – Exemplificação de transformação de problema LP

Entretanto, algumas desvantagens devem ser destacadas. Uma delas é que dependendo do número de combinações multirrótulo existentes neste conjunto, o número de classes geradas aumenta significativamente, o que resulta em uma tarefa com muitas classes a serem preditas. Além disso, outra desvantagem é a possibilidade de desbalanceamento entre as classes, onde algumas podem possuir uma frequência maior na classificação, comparado a outras, o que pode ser um problema no momento da classificação [22].

2.3.3 Multi-label k-Nearest Neighbor

Um dos métodos de adaptação de algoritmos mais conhecidos é o Multi-label K-Nearest Neighbor (ML-k-NN) baseado no kNN e no *Naive Bayes* (NB) [27]. A primeira etapa do método é exatamente igual ao kNN original descrito anteriormente. A diferença entre a versão original e a adaptada se encontra na maneira de lidar com o conjunto de rótulos a ser predito, onde o ML-kNN usa o princípio *maximum a posteriori* baseado em NB para definir o conjunto de classes de um novo exemplo [1, 25].

Primeiramente, os k vizinhos mais próximos de uma amostra do conjunto de dados são identificados. Após esta fase, para cada rótulo pertencente ás amostras vizinhas, é empregada a fórmula de Bayes para estimar se este rótulo pode ou não pertencer à amostra analisada [26].

2.4 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) estuda a comunicação humana usando métodos computacionais, ou seja, esta subárea da IA se atenta a todos os aspectos do uso de um texto natural por computadores em qualquer idioma de linguagem natural, como o Inglês e o Espanhol, por exemplo.

As linguagens naturais são mais difíceis para o computador interpretar, pois a linguagem humana não envolve apenas o entendimento das palavras e sim o contexto como um todo, ou seja, o computador precisa interpretar aspectos como ambiguidade, onde uma simples palavra pode significar outra coisa se estiver encaixada em um contexto diferente. Por isso, são necessárias técnicas que possibilitem a desambiguação, para que assim, possam ser interpretadas pelos computadores [28].

Além disso, o PLN pode ser categorizado de acordo com dois aspectos, onde é considerado se o idioma em questão é escrito ou falado, além de especificar se a tarefa desejada consiste em processar a linguagem natural ou gerar um texto como saída [29]. Logo, este trabalho está focado na linguagem escrita e no aspecto de processamento de linguagem natural.

Ainda, pode se considerar que o PLN funciona através de uma divisão entre níveis de processamento e tipos de abordagem onde os níveis de processamento representam os níveis de linguagem presentes em um idioma [29], sendo eles:

- **Fonologia:** que é responsável pela interpretação dos sons das palavras;
- **Morfologia:** que estuda a estrutura das palavras e sua composição;
- **Léxico:** que é responsável pela separação de sentenças em palavras;
- **Sintático:** que estuda a relação entre as palavras de uma frase;
- **Semântico:** que visa compreender o sentido completo de um texto com base em seu conteúdo;
- **Discurso:** que estuda o significado do texto como um todo;
- **Pragmático:** que visa tentar buscar significados que não estão presentes de forma explícita no texto (as entrelinhas).

Os tipos de abordagem são formas nas quais os níveis de processamento serão tratados pelo software [29], estes níveis podem ser divididos em:

- **Simbólico:** que é baseado em regras linguísticas que não possuem ambiguidade, ou seja, esta abordagem possui exemplos e contra exemplos facilmente diferenciados com o uso da lógica, sendo considerada a abordagem com processamento mais simples;
- **Estatístico:** onde é possível deduzir o uso correto dos níveis de processamento por intermédio de modelos matemáticos, através de um modelo estatístico para encontrar uma hipótese que possua maior proximidade com o exemplo analisado;
- **Conexionista:** combina o aprendizado estatístico com teorias de representação do conhecimento, de forma a desenvolver modelos genéricos para a linguagem. Este modelo é baseado em redes neurais artificiais;
- **Híbrido:** mescla o uso das todas as abordagens anteriores, a fim de tratar os problemas de PLN de forma flexível e efetiva.

Para que a máquina consiga compreender a linguagem natural, são necessárias etapas de pré-processamento para estruturar a linguagem natural, fazendo com que apenas o que é realmente relevante continue para as etapas seguintes. Para isso, são usadas algumas tarefas de pré-processamento conhecidas no estado da arte, como [30]:

- **Normalização:** consiste em padronizar alguns aspectos do texto, abrangendo tarefas como tokenização, que é responsável pela separação de palavras ou sentenças em unidades, onde cada palavra se torna um *token* no texto, por exemplo: “O céu é azul” se torna [“O”, “céu”, “é”, “azul”], remoção de *tags* (HTML, CSS, etc.), remoção de caracteres especiais, transformação de letras maiúsculas para minúsculas, etc.
- **Remoção de Stopwords:** é um método que elimina todas as palavras que não possuem alguma informação relevante para a construção do modelo. Geralmente são palavras bastante frequentes, como, por exemplo: “a”, “o”, “de”, “da”, “que” presentes na língua portuguesa. Porém, a remoção de *stopwords* só deve ser feita se a análise realmente não necessitar de informações como essas, onde, por exemplo, em análise de sentimentos se mostra relevante a palavra “não”, por manifestar conotação negativa, o que neste caso, indica um sentimento transmitido.
- **Remoção de Numerais:** remove numerais presentes no texto e possíveis símbolos que possam acompanhá-los, como: cifras de dinheiro, quilometragem, metragem, etc.

- **Correção Ortográfica:** identifica e corrige possíveis erros de digitação, vocabulário informal e abreviações, de forma a padronizar o texto para que palavras erradas não sejam confundidas com novos *tokens*, o que aumentaria a esparsidade dos dados.
- **Stemming e Lemmatization:** o *stemming* (stemização) é um processo que consiste na redução de uma palavra ao seu radical, isto é, palavras do português como "menino", "menina", "meninos" e "meninas" são reduzidos apenas para "menin". Já no caso de *lematization* (lematização), as palavras são reduzidas ao seu modo infinitivo, isso é, palavras como "tenho", "tinha", "terei" e "tem" são consideradas como "ter".

O uso de PLN durante a etapa de pré-processamento auxilia na melhora da qualidade dos resultados produzidos pela máquina, onde os termos são identificados usando o contexto linguístico agregando valores semânticos que auxiliam positivamente nos resultados obtidos pela máquina.

3 TRABALHOS CORRELATOS

Nos últimos anos, uma extensa literatura foi desenvolvida sobre as abordagens de detecção de *fake news*. Em sua maioria, os trabalhos podem ser divididos em duas categorias principais: baseadas em conteúdo de notícias ou em contexto social [2]. Este trabalho enfoca nas abordagens baseadas em conteúdo de notícias, onde o conteúdo de um texto noticioso pode ser analisado para se decidir sobre sua falsidade e objetividade.

Shu et al. [2] propuseram uma análise abrangente da detecção de *fake news* nas mídias sociais, considerando características como o conceito de *fake news* nas mídias tradicionais e sociais. Também foi utilizada uma classificação binária, obtendo uma lista de atributos significativos, como título, corpo do texto, possíveis imagens, etc. A partir disso, os autores citaram possíveis meios de resolver as técnicas de AM, deixando formas abertas de explorar esse problema com Mineração de Dados (*Data Mining* — DM).

Bajaj [31] propôs o uso de PLN e modelos de classificação para detectar *fake news* baseadas apenas no conteúdo da notícia. Para tal, apenas o conteúdo textual das notícias foi considerado e processado com a *Recurrent Neural Network* (RNN) e o *Gated Recurrent Units* (GRU). O resultado não explorou completamente o conjunto de dados utlizado nos experimentos, o que resultou em um modelo pouco abrangente.

Melhorando a previsão de *fake news* em notícias, Singhania et al. [32] desenvolveram um classificador hierárquico de três níveis (palavras, frases e manchetes) 3HAN baseado em uma rede de atenção hierárquica de dois níveis (Hierarchical Attention Network — HAN). A proposta de classificação dos autores referente a diversos aspectos do modelo obteve resultados satisfatórios, onde o 3HAN superou os resultados obtidos com HAN e seus derivados, demonstrando que a adição de uma terceira camada melhorou a acurácia da predição. No entanto, sua contribuição é limitada a uma classificação binária.

Zhou et al. [33] propuseram um modelo que investiga o conteúdo de notícias falsas nos níveis, léxico, sintático e semântico baseado no conteúdo da notícia, de modo a detectar notícias falsas ainda em seu estágio inicial. Como resultado, o modelo proposto pelos autores superou o estado da arte, no entanto, os autores consideraram apenas a detecção de falsidade em uma notícia (verdadeiro ou falso), não considerando os possíveis níveis de falsidade ou verdade que uma notícia pode conter.

As redes sociais são uma fonte comum de *fake news*. Considerando esse cenário, Shao et al. [34] identificaram a origem de potenciais *bots*[1] existentes para disseminar *fake news* no *Twitter*. A ferramenta proposta reconhece a disseminação de informações

[1] Diminuitivo de *robot*, é uma aplicação construída para simular ações humanas repetidamente de forma padrão, da mesma forma como faria um robô.

enganosas, rastreando as contas responsáveis pela disseminação inicial de notícias e alguns padrões relacionados. Uma discussão importante desse trabalho é o fato de pessoas comuns compartilharem notícias com amigos e seguidores sem verificar sua veracidade, onde quem recebe o conteúdo duvidoso vindo de uma fonte conhecida acaba confiando que o conteúdo da notícia é real, compartilhando novamente com outras pessoas. Esse fenômeno, em larga escala, compromete a identificação de uma classe conceitual real, pois notícias com conteúdo duvidoso são tomadas como verdade sem a devida análise.

O *Twitter* também foi o foco do estudo sobre o comportamento dos usuários, proposto por Ruchansky et al. [35]. Os autores propuseram um modelo que combina três características (onde são analisados o texto do artigo a ser analisado, a resposta do usuário ao artigo e se a resposta foi feita por um único usuário) para prever comunicações falsas. Os resultados contribuíram para representar usuários e artigos na identificação dos principais agentes de risco, demonstrando padrões de conversas reais ou por possíveis *bots*.

Além disso, Monteiro et al. [36] criaram uma ferramenta com o intuito de auxiliar a população brasileira em tempo real para a detecção de *fake news*, onde com uma simples interação com o *WhatsApp* é possível enviar o conteúdo textual de uma notícia e obter como resposta se é ou não uma notícia falsa. Entretanto, este projeto consegue identificar apenas resultados binários, ou seja, a notícia pode ser considerada totalmente falsa ou totalmente verdadeira.

No entanto, apenas a detecção pura de *fake news* é uma tarefa desafiadora, uma vez que o conceito de *fake news* ainda não é totalmente compreendido como observado por Ruchansky et al. [35]. De acordo com Rubin et al. [37], notícias sarcásticas e irônicas também podem ser um tipo de *fake news*, pois o leitor pode tomar uma notícia feita de cunho irônico como verdade, dependendo de como a notícia chegou a seu conhecimento somado ao viés do ser humano em tomar naturalmente como verdade aquilo que se encaixa com suas crenças. Tayal et al. [7] analisaram os *tweets* com base em duas medidas propostas, a primeira verifica se um determinado *tweet* é sarcástico e a segunda identifica a polaridade nos *tweets* sarcásticos de cunho político.

González-Ibánez et al. [38] criaram um mecanismo de busca para *tweets* com conteúdo sarcástico. Com o objetivo de examinar o impacto de fatores lexicais e pragmáticos, os autores fizeram uma comparação entre técnicas de AM como SVM e Regressão Logística (*Logistic Regression* — LR) e seres humanos na classificação de sentimentos. Como resultado, houve por uma pequena diferença, um melhor desempenho obtido por seres humanos. Entretanto, a precisão obtida de maneira geral no experimento foi baixa devido às dificuldades em identificar o sarcasmo nos dois casos.

Seguindo uma linha de pensamento analítico, Poria et al. [39] propuseram o uso de uma Rede Neural Convolucional (*Convolutional Neural Networks* — CNN) para extrair sentimentos, emoções e personalidade na detecção de sarcasmo no Twitter, onde os resultados obtidos superaram o estado da arte. No entanto, vale ressaltar que os experimentos foram realizados com uma única fonte de notícias.

Hossain et al. [40] propuseram um conjunto de dados para auxiliar em estudos de análise de *fake news* para idiomas com poucos recursos como o Bengalês[2]. Para isso, os autores consideraram 22 portais de notícias confiáveis mais populares de Bangladesh incluindo notícias legítimas e falsas onde os autores consideraram as notícias satíricas e *clickbait*[3] como pertencentes a categoria falsa. Os autores descobriram que o conjunto de dados apresentou melhores resultados através do uso de classificadores lineares de AM comparado a abordagem de Redes Neurais. Ainda, as *features* a nível de caractere se mostraram mais importantes que as *features* a nível de palavra para este idioma, onde a quantidade de pontuação em notícias falsas se mostrou mais frequente que em notícias verdadeiras.

Estudos mais recentes trabalharam para aumentar a interpretabilidade dos sistemas de detecção de *fake news*. FakeNewsTracker [41] e dEFEND [42], são exemplos de métodos para tornar esses DSSs explicáveis e permitir a visualização dos resultados dessas decisões, onde apresentam diferentes modelos com resultados promissores para o estado da arte.

Considerando as propostas relacionadas à MT, existem algumas pesquisas na literatura que tratam da classificação multirrótulo [43, 44, 45, 46]. Grande parte delas é dedicada à análise de sentimentos e a múltiplas classificações de tópicos. É importante mencionar a contribuição de Almeida et al. [46] em comparação com uma ampla gama de técnicas que fornecem *insights* sobre o viés das técnicas multirrótulo.

A maior parte dos trabalhos relacionados foi baseado em classificação binária (verdadeiro ou falso), suportados por *features* textuais e não textuais, e focados em fontes específicas como o *Twitter*. Por outro lado, nosso DSS é baseado apenas em *features* textuais extraídas das notícias coletadas através de um *pipeline* de MT. Avaliamos nossa proposta com diferentes fontes de notícias para reduzir o viés de um único portal. Além disso, nossa proposta aborda a classe multi conceitual de uma única notícia, conforme declarado na definição multirrótulo apresentada. E com isso, este trabalho segue uma abordagem diferente comparada ao estado da arte atual, visando demonstrar uma nova forma de resolução para o problema de detecção de notícias falsas .

---

[2] Uma língua indo-ariana falada pelas populações de Bangladesh.

[3] Notícias que usam títulos chamativos e sensacionalistas para chamar a atenção do leitor e atrair cliques ao conteúdo duvidoso.

4 SISTEMA DE APOIO À DECISÃO

Este capítulo apresenta uma ideia geral do funcionamento do DSS proposto neste trabalho, que é baseado em um *pipeline* para classificar documentos de notícias usando recursos estilométricos extraídos do texto. O objetivo dessa abordagem é classificar os documentos em 2 classes conceituais: falso/legítimo e satírico/objetivo, o que perfaz um total de 4 combinações de classes possíveis (satírico-falso, satírico-legítimo, objetivo-falso e objetivo-legítimo).

Todo o DSS pode ser dividido em duas partes: a criação do DSS e sua execução para obter uma previsão. A Figura 14 ilustra as etapas de criação do DSS, nas quais o modelo é construído a partir dos dados extraídos anteriormente. Este processo será descrito com mais detalhes na Seção 5. A Figura 15 refere-se ao processo de execução desse sistema criado no conjunto de validação para avaliar o resultado ou em um ambiente de produção com novos dados.

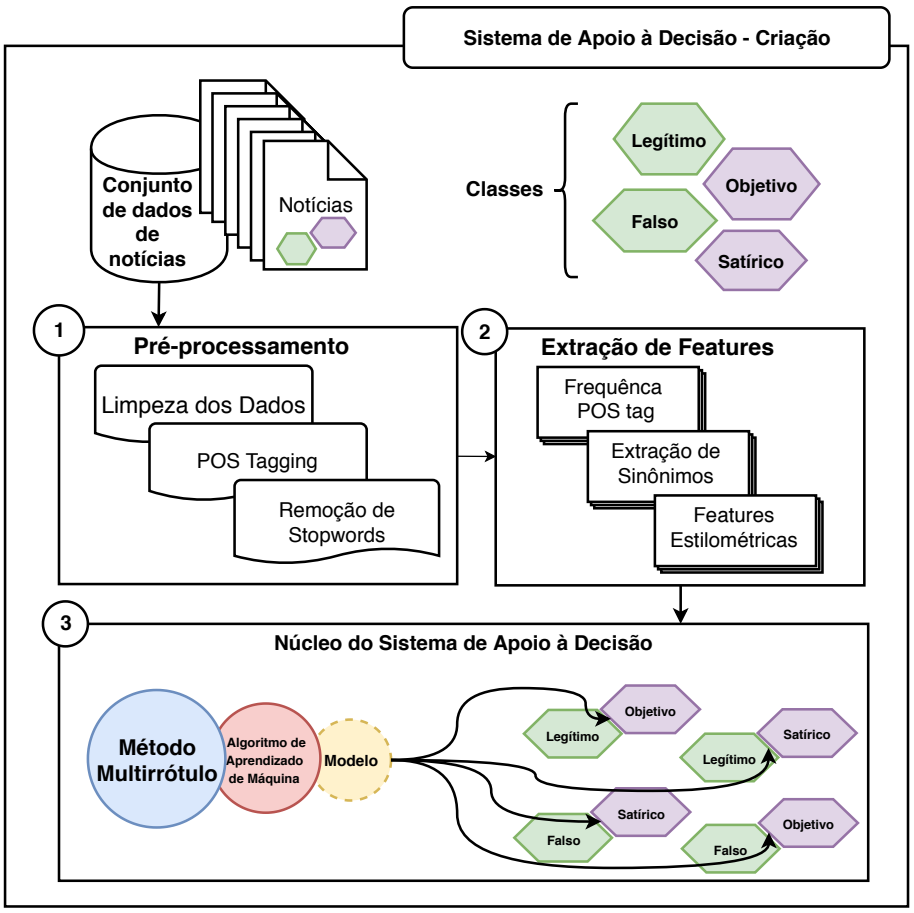

Figura 14 – Criação do DSS para a detecção de notícias falsas, legítimas, satíricas ou objetivas.

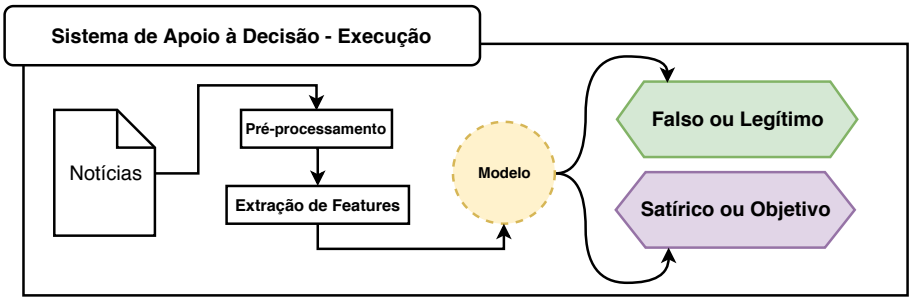

Figura 15 – Execução do DSS para a detecção de notícias falsas, legítimas, satíricas ou objetivas.

Além da criação e execução deste DSS, focamos em analisar como as *features* estilométricas extraídas do texto influenciam o resultado da classificação dos documentos. Portanto, essa análise pode se dividir em duas partes: a obtenção e a divisão das *features* do grupo e a análise dessas *features* onde cada grupo é analisado de forma separada e combinada com outros grupos com o intuito de descobrir se determinado grupo de *features* tende a obter bons resultados sem o auxílio de outros grupos.

4.1 Pré-processamento

Como visto na Figura 14, para que seja possível a criação do DSS é necessária a etapa de coleta de dados, que envolve a seleção de notícias que irão compor o conjunto de dados usados neste trabalho, conhecida na literatura como *corpus* ou *corpora*. Mais detalhes sobre a formação do *corpus* deste trabalho estão disponíveis na Seção 5.1.

Com isso, referindo-se à criação do DSS, o primeiro passo é o pré-processamento do conjunto de dados bruto. O objetivo dessa etapa é preparar o conjunto de dados para que a máquina possa compreender e analisar estes dados através do uso de técnicas de pré-processamento, proporcionando melhoria na qualidade dos dados para as etapas seguintes. Para isso são executadas etapas como:

- **Limpeza do texto:** nesta primeira etapa é feito o processo de normalização descrito com mais detalhes na Seção 2.4, onde todo tipo de ruído ou dado irrelevante é excluído além da separação dos dados textuais através do processo de tokenização, do inglês *tokenization* separando palavras ou sentenças em unidades [46, 47].
- **Etiquetagem POS:** após a limpeza do texto é realizada a etiquetagem POS (*Part-of-speech* (POS) *tagging*), que consiste em efetuar a identificação sintática de cada *token* extraído do *corpus*, onde cada *token* é classificado como pertencente a uma categoria morfológica ou a uma classe sintática, que na maioria das linguagens são representadas por: adjetivos, advérbios, conjunções, determinantes, nomes, preposições, pronomes e verbos [47, 48].

- **Remoção de *stopwords*:** após os processos anteriores é feita a remoção de *stopwords*, descrito com mais detalhes na Seção 2.4, que consiste na remoção de *tokens* que não agregam nenhuma informação relevante para as etapas seguintes, onde geralmente são palavras usadas apenas para compreensão geral do texto [48, 47]. Para este processo usamos a biblioteca NLTK em Python [49].

Como as *stopwords* possuem alto índice de frequência entre os documentos, podem ser consideradas ruídos presentes nos dados do texto por possuírem pouco poder discriminativo e, frequentemente, sua remoção melhora o desempenho do modelo [50]. Portanto, escolhemos a remoção dessas palavras junto às outras etapas pelo fato dessas palavras não serem úteis para nossa finalidade, pois não focamos em análise de sentimentos e sim estilometria textual.

4.2 Extração de Features

Após a etapa de pré-processamento é realizada a segunda etapa do DSS, responsável pelo processo de extração das *features* textuais. Para este processo, foram considerados o número médio de sinônimos por termo, a frequência de POS *tags* além da obtenção de outras *features* estilométricas baseadas no estado da arte que serão descritas com mais detalhes na sequência.

A Tabela 2 lista as *features* extraídas do conjunto de dados e faz referência aos trabalhos que basearam ou inspiraram seu método de extração, perfazendo um total de 29 *features* textuais, sendo que, 9 delas (em negrito), foram utilizadas apenas na etapa de análise de *feature importance*.

A adição destas novas *features* possibilitou a divisão das *features* extraídas em grupos com o mesmo escopo, de modo a compreender qual a real influência de cada grupo de *features* para o processo de classificação, por meio de uma análise profunda do comportamento e influência de cada grupo nos resultados.

Com isso, todas as *features* extraídas foram divididas em cinco grupos: *Complexity*, *Stylistic*, *POS tag*, *Corpus Statistics* e *Others*. A maior parte destas *features* foram extraídas de outros trabalhos do estado da arte, exceto quatro que propusemos, são elas: `avgPar`, `missWordC`, `missWordR` and `sumRed` que serão descritas com mais detalhes a seguir.

4.2.1 Features Complexity

As *features* do tipo *Complexity* são voltadas a captura da complexidade geral de um artigo, sua ideia é baseada em cálculos profundos onde são observados graus de complexidade a nível de frase e palavra [51]. Propusemos o uso de palavras médias por

Tabela 2 – Lista de *features* extraídas

No	Tipo	Nome	Descrição	Referência
1	Complexity	avgPar	Average words per paragraph	Proposto
2	Complexity	avgSen	Average words per sentence	[51]
3	**Complexity**	**avgWordSize**	**Average words per size**	**[52, 53]**
4	**Complexity**	**sentences**	**Sentences**	**[54]**
5	**Complexity**	**ttr**	**Type-token ratio**	**[52, 55]**
6	Stylistic	missWordC	Out-of-vocabulary (OOV) words count	Proposto
7	Stylistic	missWordR	Out-of-vocabulary (OOV) words ratio	Proposto
8	**Stylistic**	**upperCase**	**Uppercase letters**	**[56]**
9	**Stylistic**	**quotesCount**	**Quotation marks count**	**[51]**
10	POS tag	ratioADJ	ADJ label frequency	[2]
11	POS tag	ratioADP	ADP label frequency	[2]
12	POS tag	ratioADV	ADV label frequency	[2]
13	POS tag	ratioAUX	AUX label frequency	[2]
14	POS tag	ratioCCONJ	CCONJ label frequency	[2]
15	POS tag	ratioDET	DET label frequency	[2]
16	POS tag	ratioINTJ	INTJ label frequency	[2]
17	POS tag	ratioNOUN	NOUN label frequency	[2]
18	POS tag	ratioPRON	PRON label frequency	[2]
19	POS tag	ratioPROPN	PROPN label frequency	[2]
20	POS tag	ratioPUNCT	PUNCT label frequency	[2]
21	POS tag	ratioSCONJ	SCONJ label frequency	[2]
22	POS tag	ratioSYM	SYM label frequency	[2]
23	POS tag	ratioVERB	VERB label frequency	[2]
24	**Corpus Statistics**	**thanking**	**Thanking words**	**[57]**
25	**Corpus Statistics**	**whQuestions**	**Wh-Questions**	**[57]**
26	**Corpus Statistics**	**apoWords**	**Apology words**	**[57]**
27	Others	sumRed	Summary reducing rate	Proposto
28	Others	avgSyn	Average synonyms	[37]
29	**Others**	**emotiveness**	**Emotiveness words**	**[58]**

parágrafo (avgPar) e por sentença (avgSen) com base em recursos estilométricos [2, 51], que são computados com tokenização de palavras do texto e quebra por frases e linhas.

Com as palavras tokenizadas, foi extraída a média de palavras por tamanho (avgWordSize) e o número total de sentenças (sentences). O type-token ratio (ttr), foi extraído com a finalidade de capturar a diversidade lexical do vocabulário contido em um documento [52, 55]. Basicamente, se o valor ttr estiver baixo significa que o texto possui mais redundâncias, caso contrário, é um texto com maior diversidade lexical [59].

4.2.2 Features Stylistic

As *features Stylistic* são focadas em entender a sintaxe, os elementos gramaticais e o estilo de cada conteúdo e título contido em um artigo noticioso, sendo geralmente baseadas em PLN [51].

Com as palavras tokenizadas provindas das features de complexidade extraídas, verificamos o Mac-Morpho [60], que é um corpus de mais de 1 milhão de palavras em português do Brasil disponível no *Natural Language Toolkit* (NLTK) [49] e, em seguida,

contamos todas as palavras fora do vocabulário (*Out-of-vocabulary* — OOV) marcadas como adjetivo (ADJ), advérbio (ADV), verbo (VERB) ou substantivo (NOUN) que não foram encontradas no conjunto, supondo que seja uma palavra informal ou um neologismo. Vale ressaltar que neste caso, as palavras com grafia incorreta podem ser consideradas palavras fora do vocabulário.

Em seguida, essa contagem é usada para extrair o número total de palavras OOV (missWordC) e a proporção de palavras OOV para o número total de tokens (missWordR). Além disso, a contagem de letras maiúsculas (upperCase) e a contagem de aspas (quotesCount) foram extraídas.

4.2.3 Features POS tag

A frequência de etiquetagem POS *tag*, consiste na extração da frequência de cada etiqueta extraída no processo de POS *tagging* para todo o documento. Conforme mostrado em [2, 51, 61], as *POS tags* são usadas como descritores linguísticos na literatura de detecção de *fake news*.

As *POS tags* escolhidas para este trabalho, como visto na Tabela 2 foram: adjetivo (ADJ), adposição (ADP), advérbio (ADV), verbo auxiliar (AUX), conjunção coordenativa (CCONJ), determinante (DET), interjeição (INTJ), substativo (NOUN), pronome (PRON), nome próprio (PROPN), pontuação (PUNCT), conjunção subordinada (SCONJ), símbolo (SYM) e verbo (VERB).

4.2.4 Features Corpus Statistics

As *features* de *Corpus Statistics* são baseadas na comunicação linguística, onde são consideradas ações como desculpas, agradecimentos, promessas, etc [57, 62]. Para este artigo usamos apenas 3 comunicações linguísticas: palavras de agradecimento (thanking), pronomes e advérbios interrogativos (whQuestions) e palavras relacionadas a desculpa (apoWords).

Todas as *features* deste grupo foram originalmente pensadas e implementadas para a língua inglesa e, para este trabalho, as *features* thanking, whQuestions e apoWords foram adaptadas à sua equivalência na língua portuguesa, respeitando a diferença gramatical existente entre os idiomas Inglês e Português.

4.2.5 Others

Este último grupo consiste em *features* que não se encaixaram em nenhum grupo anterior, e cada uma das *features* possui uma característica particular.

A *feature summary reducing rate* (sumRed) foi proposta neste trabalho considerando a hipótese de que jornalistas profissionais em veículos de mídia tradicionais escrevem

os parágrafos principais (geralmente o primeiro parágrafo de um texto jornalístico contendo as informações mais importantes sobre o texto [63]) diferentemente das notícias escritas por não profissionais. Assim, geramos um resumo automatizado, obtido através de uma variação do algoritmo TextRank, que é um algoritmo baseado em gráfico independente de idioma, pois não utiliza conhecimento linguístico como base [64].

O TextRank determina a relação de similaridade entre duas frases com base em seu conteúdo, determinando assim quais elementos descrevem melhor o texto, e com isso, o algoritmo consegue criar resumos sem a necessidade um *corpus* de treinamento ou rotulagem, que se encaixa na estilometria de escrita [64].

Para determinar a relação, o TextRank modela qualquer documento como um grafo usando sentenças como nós, onde uma função é necessária para calcular a similaridade das sentenças e construir bordas entre elas. E com isso, a função é utilizada para ponderar as bordas do grafo e quanto maior a similaridade entre as sentenças, mais importante será a borda entre elas no grafo [64]. A partir disso é feita a comparação do resultado do resumo com o tamanho do artigo original, onde um artigo possuirá um resumo de seu conteúdo caso seu primeiro parágrafo possua similaridade com o restante do texto.

Os sinônimos são obtidos usando um modelo pré-treinado word2vec [65] contando o número de termos mais semelhantes com uma medida de similaridade superior a um limite. Depois disso, é obtida uma média da contagem de sinônimos (avgSyn) para o documento. Esse recurso está relacionado aos recursos de validade semântica propostos por Rubin et al. [37], que consideram a ambiguidade e a incoerência de conceitos como uma característica que pode estar relacionada a textos satíricos.

E por fim as palavras emotivas (emotiveness), que são a proporção de modificadores para palavras de conteúdo, podendo ser formalmente definida da seguinte forma [58].

$$Emotiveness = \frac{\sum adjetivos \ \& \ advérbios}{\sum substantivos \ \& \ verbos} \tag{4.1}$$

4.3 Kernel do Sistema de Apoio a Decisão

Após todo o processo de extração descrito na etapa (2), são gerados vetores destas *features* onde cada instância é equivalente a um documento do conjunto de dados bruto. Cada instância possui dois rótulos, um para cada classe conceitual e são usados para induzir o modelo de decisão.

Então, na terceira etapa, é criado um modelo de predição usando um dado método multirrótulo com estes vetores de *features*. Este modelo é o *kernel* de decisão do processo, onde o modelo de AM extrai padrões para distinguir as classes criadas por um domínio com vários rótulos.

Como estamos classificando documentos que podem ser legítimos ou falsos, satíricos ou objetivos, a abordagem multirrótulo é mais apropriada do que uma simples classificação multi-classe. Em vez de pertencer a uma única categoria, um documento é rotulado como pertencente a duas classes ao mesmo tempo, por exemplo, falso e satírico. Logo, a ideia deste trabalho é fazer uma avaliação de diferentes algoritmos multirrótulo, de forma a discutir e avaliar seu desempenho ao longo da dissertação.

4.4 Execução do Sistema de Apoio à Decisão

O objetivo da fase de execução do DSS é determinar as classes de novos documentos que não foram avaliados pelo sistema durante a fase de criação. Durante esta fase, as mesmas operações iniciais de pré-processamento e extração de *features* da criação do DSS são executadas para gerar os vetores de *features*. A etapa final desta fase consiste na execução de um modelo de AM, criado na fase anterior. Este modelo gera uma previsão para ajudar na decisão da classe de um documento textual, e após todo o estágio de execução do DSS, é analisada a importância das *features* (*feature importance*) extraídas na etapa (2), com a intenção de mostrar quais grupos de *features* são mais relevantes para o modelo de decisão.

5 MATERIAIS E MÉTODOS

Este capítulo descreve os processos utilizados em cada fase do DSS proposto, onde apresenta o processo de coleta do conjunto de dados utilizado, o critério de escolha dos algoritmos utilizados nos experimentos e as métricas de avaliação empregadas para avaliar os diferentes aspectos destes algoritmos.

5.1 Conjunto de Dados

O conjunto de dados utilizado neste estudo foi coletado por vários portais de notícias brasileiras. A coleta foi implementada em duas partes que serão descritas a seguir:

Na primeira parte, foi utilizado um *web crawler*[1] para reunir uma grande quantidade de artigos de notícias de cada site. Um *web crawler* ou simplesmente *crawler* é o nome dado aos robôs que possuem capacidade de navegar na Internet de forma autônoma, coletando dados de forma automática mediante instruções pré-definidas tais como: quais hiperlinks serão visitados e quais os critérios de captura de dados são abordados para cada site [47].

Para cada combinação de classes, exceto as de notícias falsas e objetivas, foram selecionados portais com sua finalidade conhecida e foi aplicado um filtro para capturar apenas notícias pertencentes a categoria política. Para notícias objetivas e legítimas, os sites selecionados foram o G1[2] e o UOL Notícias[3], dois dos sites mais visitados no Brasil segundo o *ranking* Alexa[4]. Notícias satíricas e falsas foram coletadas a partir do Sensacionalista e Diário Pernambucano[5], que são sites satíricos que zombam de notícias reais, caçoando de assuntos atuais. Para notícias satíricas e legítimas, este estudo considerou sites que reúnem eventos incomuns ou inesperados com um tom humorístico, como Surrealista[6], UOL Tabloide[7] e Planeta Bizarro[8].

Na segunda parte, foram utilizadas agências de checagem de fatos para a coleta de notícias objetivas e falsas, onde não foi possível a coleta através do *crawler*, pois não há nenhum portal que contenha assumidamente notícias deste tipo a conhecimento público. As agências de checagem de fatos publicam artigos que verificam a veracidade das notícias que são difundidas nas redes sociais ou em sites duvidosos. Para reunir os documentos

[1] https://scrapinghub.com/
[2] https://g1.globo.com/
[3] https://noticias.uol.com.br/politica/
[4] https://www.alexa.com/
[5] http://www.diariopernambucano.com.br/
[6] https://www.surrealista.com.br/
[7] https://noticias.uol.com.br/tabloide/
[8] https://g1.globo.com/planeta-bizarro/

utilizados no *corpus* deste estudo foram escolhidas as agências Lupa[9] e Boatos[10].

Como conjuntos de dados desequilibrados geralmente causam problemas no treinamento de um modelo de AM [66], selecionamos apenas as verificações que foram checadas em relação aos documentos textuais, totalizando 58 documentos. O *corpus* de notícias objetivo-falso coletado contém documentos provindos de 30 sites diferentes, principalmente relacionados à política e às eleições brasileiras de 2018. Notícias sobre imagens ou manchetes simples não foram incluídas no *corpus* porque vão além do escopo desta pesquisa.

Finalmente, o conjunto de dados foi construído por amostragem aleatória para cada combinação de classes. A versão usada para os experimentos deste trabalho, possui 70 documentos satírico-legítimo, 70 satírico-falso, 70 objetivo-legítimo e 58 objetivo-falso, conforme representado na Figura 16, que demonstra a disposição e as correlações existentes entre estes documentos, ou seja, cada uma das classes é representada por um segmento ao longo da circunferência (separado por cor) e cada uma das possíveis correlações entre as classes é representada através dos arcos existentes (ou curvas de Bézier). Além disso, a espessura de cada arco está diretamente ligada a quantidade de documentos que representam cada uma das correlações.

De acordo com a figura, é possível perceber que há representantes de todas as combinações de classes conceituais, onde percebe-se que pela espessura do arco e sua área de representação que a classe objetivo-falso tem uma menor quantidade de documentos em relação as outras classes. No entanto, mesmo com a menor quantidade de exemplos, o conjunto de dados ainda possui dados balanceados o que é extremamente importante para a realização dos experimentos, e esta diferença, se deve a maior dificuldade em encontrar notícias assumidamente objetivo-falso. Exemplos do conteúdo dos documentos coletados de cada uma das classes são mostrados na Tabela 3.

O conjunto de dados atual está disponível no GitHub[11] e contém um total de 268 documentos rotulados, com uma média de 370 *tokens* por documento e um desvio padrão de 296, onde a menor instância possui 36 *tokens* e a maior, 1805.

5.2 Decisão do Modelo

Os métodos de comparação utilizados no DSS baseiam-se na transformação de problemas multirrótulo, adaptação de problemas multirrótulo e algoritmos de classificação multi-classe. Para a transformação e adaptação de problemas multirrótulo foram usados três algoritmos de AM multi-classe como classificadores de base, estes algoritmos também

[9] https://piaui.folha.uol.com.br/lupa/
[10] https://www.boatos.org/
[11] https://github.com/hugoabonizio/fake-news-multilabel

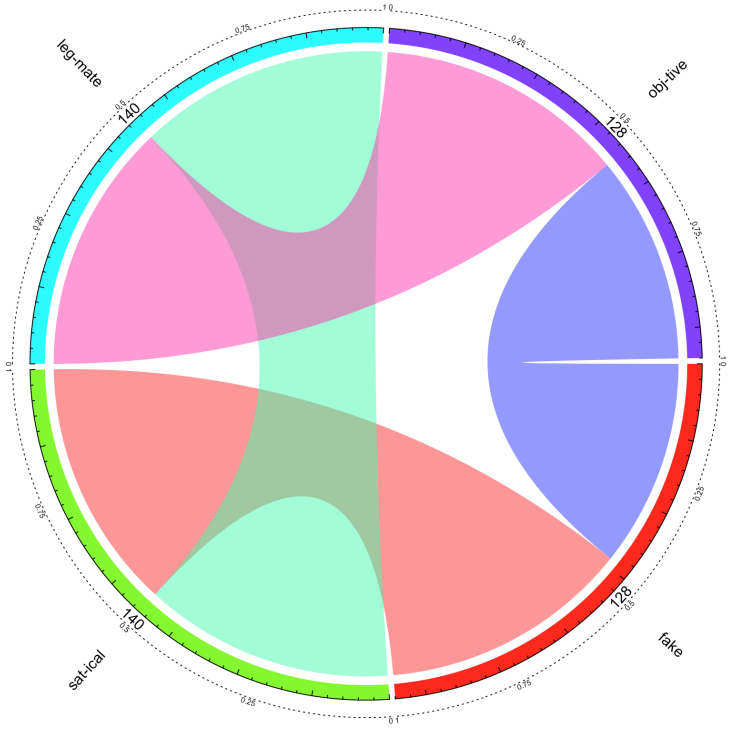

Figura 16 – Relação das classes conceituais multirrótulo: Falso, Legítimo (leg-mate), Objetivo (obj-tive) e Satírico (sat-ical)

foram utilizados para a classificação multi-classe, são eles: RF [15], SVM [67] e kNN [68], baseados em diferentes vieses e ramos de AM.

Existe uma grande variedade de algoritmos multirrótulo [69], mas neste estudo nos concentramos em avaliar os representantes de métodos de transformação de problemas e adaptação de algoritmos para problemas multirrótulo. As técnicas de transformação de problemas usadas nos experimentos foram BR e LP [27], para que possamos avaliar e comparar a abordagem multirrótulo em diferentes métodos multi-classe.

O método *Binary Relevance*, como dito anteriormente, decompõe os rótulos q em q classificadores binários independentes que preveem se uma instância tem um rótulo correspondente [27]. Este método possui uma desvantagem quando os rótulos têm correlações, o que não é o caso neste trabalho, que classifica em duas classes conceituais independentes. O método *Label Powerset* converte cada combinação de rótulo exclusiva em uma classe única e, em seguida, cria um conjunto em que cada componente tem como alvo um subconjunto aleatório do problema, que aborda a desvantagem da *Binary Relevance* de não considerar correlações entre rótulos.

A combinação de algoritmos de AM como classificadores base e métodos multirrótulo é referida neste trabalho como: BR_kNN, BR_RF, BR_SVM, LP_kNN, LP_RF e LP_SVM. Para os métodos de adaptação do algoritmo, utilizamos o ML-kNN [70], que é uma adaptação do algoritmo kNN para dados multirrótulo.

Tabela 3 – Exemplo do conteúdo de noticias das classes conceituais

Classes Conceituais		**Conteúdo**
Objetivo	Legítimo	TSE apresenta previsão do tempo de propaganda no rádio e na TV para cada candidato à Presidência O Tribunal Superior Eleitoral (TSE) apresentou nesta quinta-feira (23) o tempo previsto para a propaganda no rádio e na televisão de cada um dos 13 candidatos à Presidência da República, para a campanha do primeiro turno das eleições deste ano. (...)
Objetivo	Falso	MST promete guerra civil em caso de prisão de Lula À medida que cresce a força de Lula no seio do eleitorado brasileiro cresce, também, a perseguição movida contra ele pela Operação Lava-Jato e pela mídia golpista. (...)
Satírico	Legítimo	Assaltantes perdem dinheiro de roubo após rajada de vento “Dinheiro na mão é vendaval” é uma grande mentira? Neste caso, um vendaval tirou o dinheiro da mão de bandidos que assaltaram uma agência de viagens em Droylsden, na região da Grande Manchester, na Inglaterra. (...)
Satírico	Falso	Após fim de supletivo em Economia, Bolsonaro dará aulas na UFRJ Após contratar Adolfo Salsisa, professor de economia básica para supletivo dos políticos do DEM, Bolsonaro já tem indicação da Escola Sem Partido para lecionar no Instituto de Economia da UFRJ. Apesar das queixas do professor acerca dos cochilos do aprendiz, Salsisa prevê um futuro presidente bastante graduado em Economia, quiçá mais preparado que Ciro Gomes. (...)

O passo final consiste em treinar os modelos e recuperar suas performances sobre o conjunto de testes, o que é feito através de um processo de validação cruzada estratificada (*stratified cross-validation*) [71], onde cada métrica de desempenho foi calculada após dez iterações da validação cruzada com 10 partições aleatórias do conjunto de treinamento (10-*fold cross-validation*). Além disso, a validação cruzada estratificada garante que todos os algoritmos tenham acesso mesma partição de treinamento e teste durante os experimentos, assegurando uma comparação justa entre os métodos multi-classe e multirrótulo.

5.3 Métricas

Para avaliar a abordagem proposta, testamos um conjunto de modelos e algoritmos *base learners* e comparamos seus resultados. As métricas usadas nessa comparação são acurácia e *F1-score*, disponíveis para classificação multirrótulo e multi-classe.

A classificação multirrótulo não possui o conceito “verdadeiro ou falso” presente na classificação binária, portanto as métricas para este tipo de classificação consideram a possibilidade de um exemplo ser classificado de maneira parcialmente errada ou parcialmente correta.

Para definir a acurácia da classificação multirrótulo, seja D um conjunto de avaliação multirrótulo, Y seja o conjunto verdadeiro de rótulos e Z, seja o conjunto previsto de rótulos, Tsoumakas e Katakis [21] definem a acurácia como:

$$Acurácia = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Y_i \bigcup Z_i|} \tag{5.1}$$

Para definir o *F1-score*, primeiro precisamos definir as métricas de precisão e revocação (*recall*), definidas por Tsoumakas and Katakis [21] como:

$$Precisão = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Z_i|} \tag{5.2}$$

$$Revocação = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \bigcap Z_i|}{|Y_i|} \tag{5.3}$$

Já para definir as métricas multi-classe, onde o resultado pode ser positivo ou negativo para cada classe separadamente, é possível encontrar a Acurácia e o *F1-score* a partir da matriz de confusão que ilustra o número de predições corretas e incorretas de cada classe.

As linhas da matriz representam as classes verdadeiras e as colunas as classes que foram preditas pelo classificador, além disso, os termos verdadeiro positivo (*true positive* - TP), verdadeiro negativo (*true negative* - TN), falso positivo (*false positive* - FP) e falso negativo (*false negative* - FN), ilustram a quantidade de erros e acertos presentes no problema, como pode ser visto na Tabela 4 [1].

Tabela 4 – Matriz de confusão para duas classes.

		Classe Predita	
		+	-
Classe Verdadeira	+	VP	FN
	-	FP	VN

Com isso, Olson and Delen [72] e Fawcett [73] definem a Acurácia, a Precisão e a Revocação como:

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

$$Precisão = \frac{TP}{TP + FP} \tag{5.5}$$

$$Revocação = \frac{TP}{TP + FN} \tag{5.6}$$

Por fim, a partir das métricas de precisão e revocação, tanto para a classificação multirrótulo quanto para a multi-classe, o *F1-score* é definido por:

$$F1 = \frac{2 * Precisão * Recall}{Precisão + Recall} \tag{5.7}$$

6 RESULTADOS E DISCUSSÕES

Os experimentos deste estudo possuem dois resultados: o desempenho do DSS proposto para classificação de notícias, onde comparamos a abordagem multirrótudo com algoritmos multi-classe explanando os resultados obtidos, e as informações que podem ser extraídas através destes resultados, onde é realizada uma análise de *feature importance* de modo a compreender a importância de cada *feature* para o estudo. Para o desempenho do modelo proposto, foram utilizadas como métricas de avaliação a acurácia (*accuracy*) e o *F1-score*, como mostra a Figura 17.

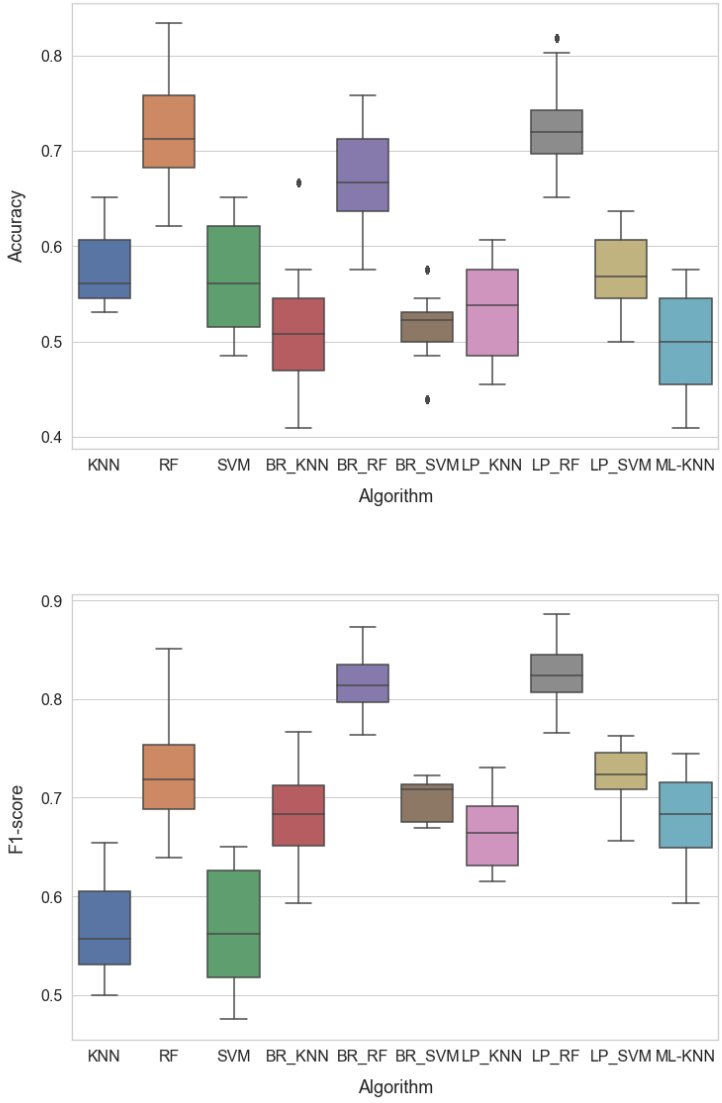

Figura 17 – Resultado do processo de *cross-validation* através de diferentes algoritmos usando as métricas de acurácia e *F1-score* representadas através de um gráfico de caixas (*boxplot*).

Conforme visto na Figura 17, a acurácia de LP_RF e RF obtiveram as duas pontuações mais altas (70% e 69%, respectivamente), com a diferença entre elas sendo estatisticamente irrelevante. Isso significa que o algoritmo RF obteve os melhores resultados tanto em sua abordagem clássica (segunda pontuação mais alta), quanto unido aos métodos de transformação de problema multirrótulo LP e BR (primeira e terceira pontuação mais altas, respectivamente).

Este é um resultado razoável, porque a RF é uma abordagem de *ensemble* que cria classificadores fracos (*weak classifiers*) randômicos que, por sua vez, votam na decisão final, fazendo com que o modelo evite o excesso de ajustes e seja robusto a valores discrepantes (*outliers*) e ruídos [15], mostrando-se adequado até mesmo em abordagem multi-classe.

No entanto, no *F1-score*, a diferença entre as abordagens multi-classe e multirrótulo é mais significativa, com a RF simples obtendo 68% e LP_RF alcançando 81%. Os dois *F1-scores* mais altos são dos métodos multirrótulo LP (81%) e BR (79%) usando RF como classificador base, seguido por LP_SVM (69%) ocupando o terceiro lugar.

Com este resultado, podemos afirmar que a abordagem multirrótulo proposta neste trabalho, usando os métodos de transformação de problemas *Label Powerset* ou *Binary Relevance*, é adequada para o problema, pois os métodos multirrótulos vinculados com algoritmos multi-classe se mostraram superiores à maioria dos métodos clássicos tanto em acurácia quanto em *F1-score*.

Contudo, é importante salientar qual é a diferença entre as implementações multi-classe clássica e multirrótulo usando *label powerset*. A abordagem multi-classe, basicamente usa um único modelo implementado em python (com combinações pré-definidas e imutáveis) e o LP lida com todos os modelos possíveis (*all vs. all*). Por isso, houve uma diferenciação com melhor resultado na abordagem LP, onde foram obtidos resultados mais promissores se comparado a abordagem multi-classe.

O desempenho das combinações de algoritmos SVM e kNN foi significativamente pior do que as combinações equivalentes de RF, mas suas versões clássicas (multi-classe) foram as que demonstraram o pior desempenho tanto em acurácia quanto em *F1-score*. Dada a natureza determinística da SVM e do kNN, suas caixas presentes na Figura 17, geralmente são mais curtas porque apresentam uma variação menor, exceto no caso da SVM clássica, onde a caixa possui uma alta variação o que indica que os hiperparâmetros do modelo precisam ser ajustados de modo a alcançar um melhor resultado. A natureza não determinística da RF explica a alta variação nos resultados, que segue uma distribuição normal aproximada.

Apesar da SVM apresentar desempenho pior que a maioria os algoritmos, na medida *F1-score* sua combinação com LP obteve um resultado superior (69%) ao algoritmo

RF clássico (68%). No entanto, esta diferença não tem relevância estatística. Contudo, comparando o desempenho entre LP_SVM e a abordagem SVM clássica (50%) é possível observar o aumento de desempenho que uma abordagem multirrótulo é capaz de alcançar.

Os *outliers* que aparecem nos resultados da pontuação, indicam que as divisões do conjunto de dados feitas durante o processo de validação cruzada estratificada influenciaram diretamente na facilidade ou dificuldade de trabalho para alguns modelos. Isso demonstra como cada modelo lida de forma diferente com o conjunto de dados durante sua execução. Esses *outliers* podem ser evitados usando um conjunto de dados maior.

O desempenho do modelo ML-kNN no *F1-score* demonstrou que esse algoritmo adaptado, apesar de apresentar melhores resultados que as abordagens multi-classe, ainda carrega as limitações presentes no kNN clássico quando comparado a um método com hiperparâmetros definidos (RF).

Uma pergunta importante que procuramos responder é a importância das *features* extraídas do conjunto de dados e seu impacto na decisão da classe de um documento. Para responder a essa pergunta, extraímos a *feature importance* baseada nos resultados da *Random Forest* obtidas nos experimentos, de modo a fornecer um *ranking* da importância de cada variável preditora, considerando as árvores criadas pelo algoritmo [74].

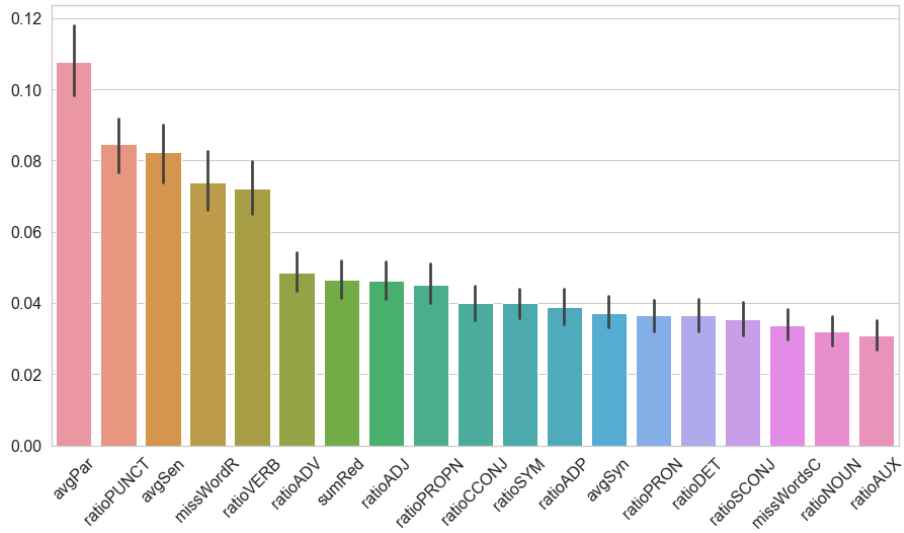

Figura 18 – RF *importance* das *features* textuais exploradas.

A Figura 18 mostra o *ranking* de importância variável onde é possível notar que a média da quantidade de palavras por parágrafo (avgPar) e por sentença (avgSen) junto à frequência de palavras rotuladas com pontuação (ratioPUNC) estão entre as *features* mais importantes para descrever o conjunto de dados, indicando que há uma diferença no tamanho e na densidade do texto que divide as classes. As *features* seguintes na ordem de importância foram: a proporção de palavras fora de vocabulário (missWordR) e a frequência de palavras rotuladas com verbo (ratioVERB) para a contagem total de

palavras.

As features de *POS tags* se mostraram importantes variáveis preditoras (onde as *features* com classificação mais alta no *ranking* foram respetivamente: pontuação (PUNCT), verbo (VERB) e advérbio (ADV)) para classificar notícias falsas e satíricas, confirmando os resultados encontrados por Shu et al. [2] e Horne and Adali [51].

A Tabela 5 mostra a média e o desvio padrão dos valores das *features* agrupadas por combinações de rótulos. É possível afirmar pelos resultados que, com relação ao número de palavras por parágrafo (avgPar), há pouca diferença entre notícias objetivas e artigos satírico-falso, mas os documentos satírico-legítimo têm parágrafos claramente menores e uma variação maior tanto em avgPar quanto em ratioPUNCT. O avgSen, que conta a média de palavras por sentença, indica que os artigos objetivo-falso possuem sentenças substancialmente menores, sendo um sinal de que esse tipo de documento possui uma linguagem menos complexa, visando ser mais acessível e superficial.

Tabela 5 – Valor médio e desvio padrão das *features* extraídas agrupadas pelas combinações de rótulos.

Nome da Feature	**Objetivo Legítimo**	**Objetivo Falso**	**Satírico Falso**	**Satírico Legítimo**
avgPar	40,685 (11,520)	42,167 (13,140)	43,532 (16,351)	28,590 (22,570)
ratioPUNCT	0,137 (0,027)	0,126 (0,023)	0,105 (0,022)	0,116 (0,041)
avgSen	21,571 (3,395)	16,815 (4,014)	17,5236 (5,478)	20,602 (7,310)
missWordR	0,008 (0,004)	0,022 (0,017)	0,020 (0,015)	0,027 (0,043)
ratioVERB	0,108 (0,019)	0,138 (0,023)	0,134 (0,021)	0,121 (0,030)
ratioADV	0,035 (0,010)	0,042 (0,020)	0,046 (0,020)	0,038 (0,014)
sumRed	0,284 (0,065)	0,230 (0,092)	0,217 (0,091)	0,232 (0,096)
ratioADJ	0,043 (0,013)	0,036 (0,018)	0,043 (0,018)	0,048 (0,022)
ratioPROPN	0,109 (0,040)	0,079 (0,030)	0,115 (0,049)	0,110 (0,082)
ratioCCONJ	0,021 (0,006)	0,021 (0,010)	0,021 (0,011)	0,024 (0,011)
ratioSYM	0,016 (0,017)	0,013 (0,009)	0,011 (0,010)	0,016 (0,014)
ratioADP	0,146 (0,021)	0,138 (0,021)	0,138 (0,027)	0,130 (0,028)
avgSyn	6,842 (0,463)	7,050 (0,580)	6,783 (0,617)	6,636 (0,883)
ratioPRON	0,025 (0,012)	0,033 (0,016)	0,030 (0,014)	0,032 (0,017)
ratioDET	0,096 (0,017)	0,108 (0,016)	0,105 (0,022)	0,105 (0,022)
ratioSCONJ	0,012 (0,007)	0,013 (0,008)	0,014 (0,009)	0,012 (0,008)
missWordsC	5,442 (4,325)	5,414 (4,882)	4,514 (4,705)	9,586 (17,561)
ratioNOUN	0,175 (0,029)	0,187 (0,027)	0,177 (0,028)	0,177 (0,035)
ratioAUX	0,017 (0,007)	0,020 (0,013)	0,019 (0,012)	0,001 (0,009)

Com relação a essas variáveis, os resultados demonstraram que as notícias objetivo-falso têm uma média de palavras significativamente menor por parágrafo e mais palavras fora do vocabulário (OOV). As *features* propostas (avgPar, missWordR e sumRed) foram importantes descritores de objetividade e legitimidade dos documentos de notícias.

Com base nos resultados obtidos nos experimentos descritos, fizemos uma análise detalhada de *feature importance*, onde foram adicionadas além das 20 *features* utilizadas nos experimentos anteriores, mais 9 *features*. Para esta análise, foi escolhida uma combinação multirrótulo de *Binary Relevance* com *Random Forest*, pois foi uma das combinações que obteve um dos melhores resultados nos experimentos. Ainda, analisamos as combinações de cada grupo de *features* separadamente e todas as combinações possíveis entre esses grupos, que podem ser vistas na Figura 19.

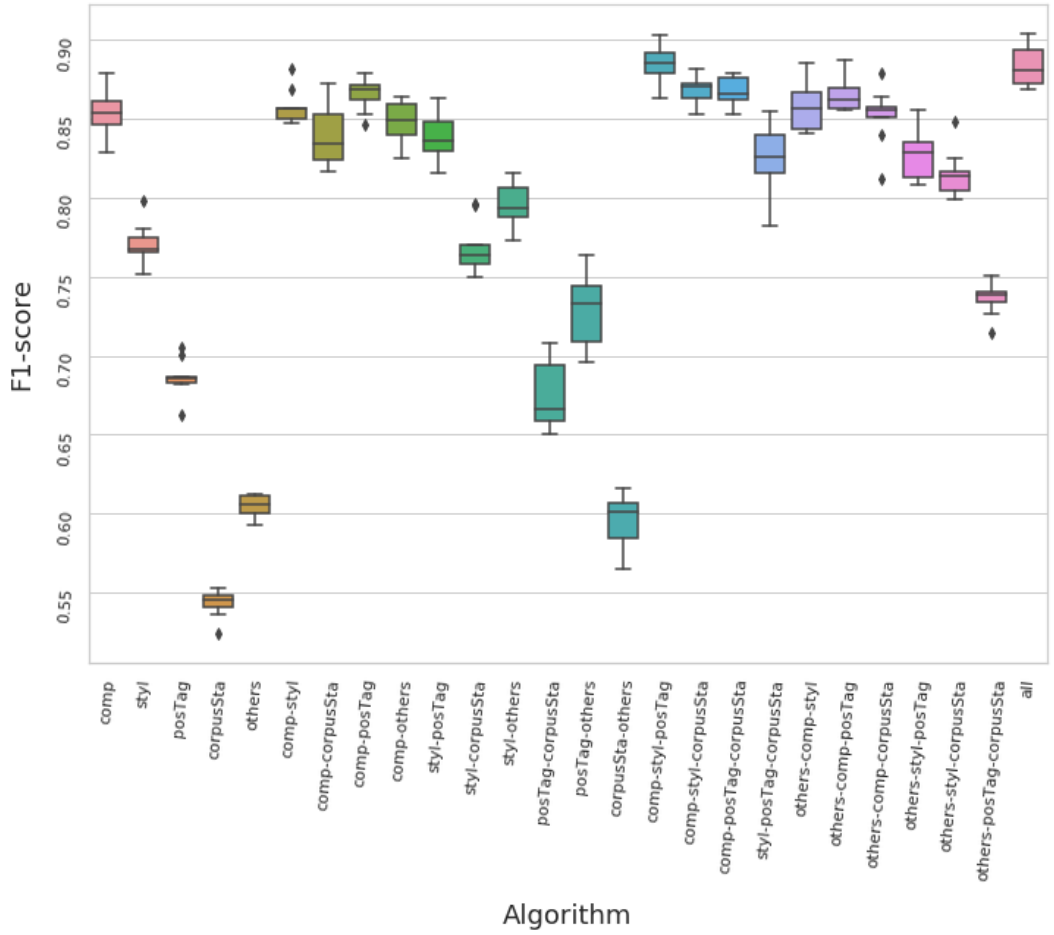

Figura 19 – Resultado do processo de validação cruzada com 10 dobras (10-*fold cross-validation*), usando BR_RF e a métrica *F1-score* para cada combinação de *features* através de um *boxplot*.

Como observado, os grupos de *features* que contêm *features* de *Complexity* têm um *F1-score* mais alto. Grupos contendo *features Stylistic* também têm desempenho significativo, seguido de *POS tags* e *Others*. Os grupos que contêm as *features* de *Corpus Statistics* não se mostraram relevantes, o que pode ser visto com mais clareza quando somente as *features* de *Corpus Statistics* são analisadas, onde é visto o pior resultado em comparação com os outros grupos. Isso mostra que a complexidade textual e suas características sintáticas se mostraram mais relevantes para os resultados do que o tipo de comunicação linguística utilizado em cada notícia.

Também pode ser observado na Figura 19 que a maioria das caixas possuem uma pequena amplitude, o que indica que na maioria dos grupos comparados houve uma pequena variação de valores. É possível notar também que a maioria das caixas com maior amplitude contém *features* de *Corpus Statistics*, o que pode indicar uma variação maior de valores além do seu desempenho inferior aos outros grupos.

Para visualizar a distribuição das amostras no espaço de *features*, aplicamos o *t*-SNE [75], uma técnica de redução de dimensionalidade não linear que gera uma projeção bidimensional do conjunto de dados. O *t*-SNE funciona através da otimização da divergência de Kullback-Leibler entre duas distribuições, onde a distribuição de probabilidade gaussiana é baseada na relação entre cada ponto no espaço original e a distribuição Student-t recria a distribuição em um espaço de dimensão inferior. Esta técnica se difere de outras técnicas de redução de dimensionalidade, como a Análise de Componentes Principais (*Principal Component Analisys* — PCA), uma vez que o *t*-SNE mapeia relacionamentos não lineares complexos da estrutura de dados local e global. Para a geração do *t*-SNE, utilizamos como base a biblioteca *scikit-learn* [76] sem alteração dos parâmetros padrões disponibilizados na biblioteca. Além disso, como o *t*-SNE possui uma função de custo com diferentes inicializações, os resultados gerados tendem a ser diferentes em cada rodada, e para o nosso problema consideramos o *t*-SNE com o melhor resultado obtido, baseado no melhor resultado em n rodadas do algoritmo. A Figura 20 mostra cada amostra como um ponto no espaço de *feature*, destacando a separabilidade existente nas classes legítimo/falso mostrando a existência de um lado predominante para cada uma das classes. No entanto, os dados são dispersos e não existe uma separabilidade precisa entre legítimo e falso. As classes objetivo/satírico assim como legítimo/falso também possuem uma separabilidade visível, porém nota-se que a separação entre estas duas classes é um pouco mais homogênea comparado a legítimo/falso.

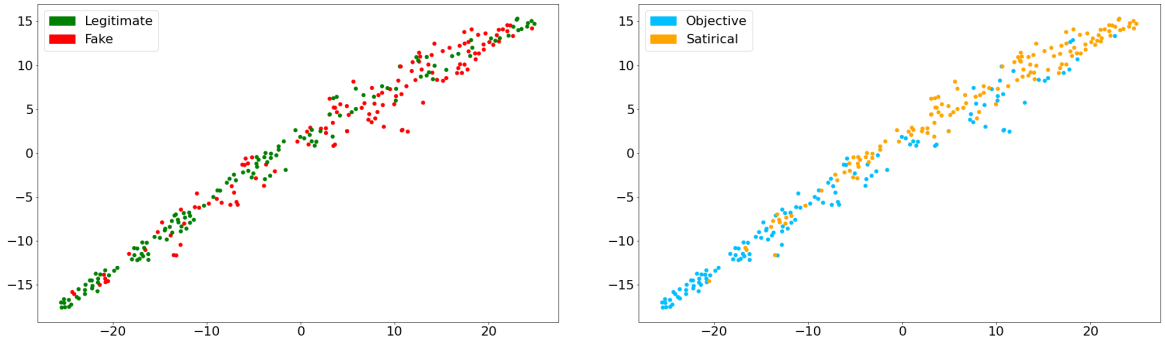

Figura 20 – Visualização da distribuição de amostras sobre o espaço de *features* usando *t*-SNE para a redução de dimensionalidade.

As Figuras 21 e 22 mostram o teste Nemenyi para a acurácia e *F1-score*, onde as diferenças estatísticas de cada um dos grupos analisados e suas respectivas combinações são mostradas. Cada um dos grupos conectados pela barra horizontal ondulada faz parte da mesma barra de distância crítica (*Critical Distance* — CD), ou seja, esses grupos interconectados não são estaticamente diferentes um do outro e, como visto na Figura 19, os grupos que contêm as *features* de *Complexity* obtiveram um resultado mais preciso.

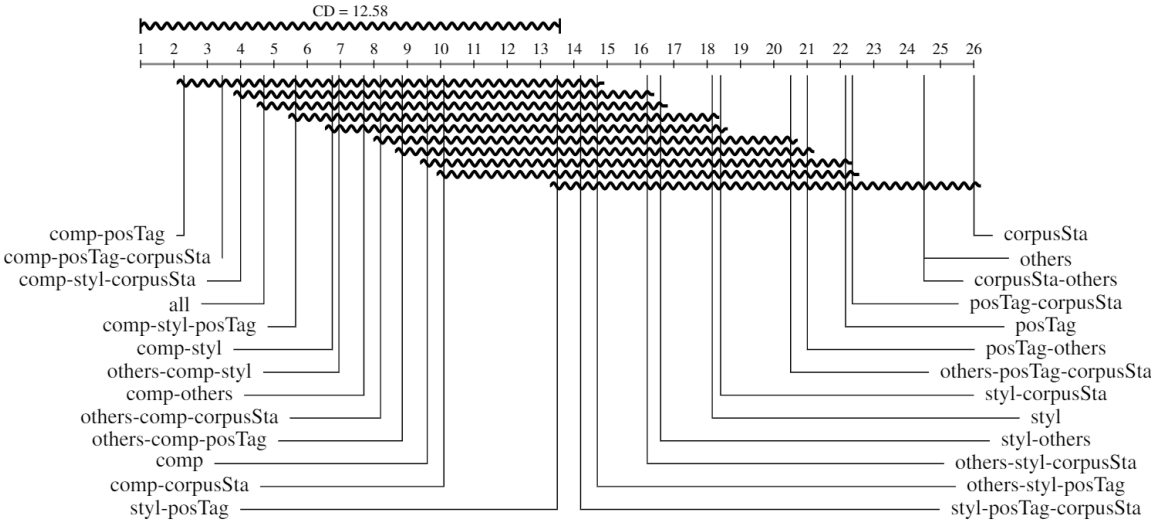

Figura 21 – Comparação dos modelos treinados com cada combinação de grupo de *features* de acordo com o teste Nemenyi, levando em consideração a métrica de acurácia. Os resultados dos grupos conectados não são significativamente diferentes (em $\alpha = 0,05$).

Durante a análise de Nemenyi, tanto em relação à acurácia quanto ao *F1-score*, é possível perceber a divisão das combinações de *features* em dois grandes grupos, o primeiro contendo combinações de grupos incluindo as *features* de *Complexity* que fazem parte da mesma CD à esquerda e o segundo contendo as combinações de *features* de *Corpus Statistics* à direita.

Apesar da distinção notável que confirma a maior diferença entre *features* de *Complexity* e *Corpus Statistics*, é possível notar também que a CD possui escalas, onde algumas combinações de grupos de *features* que pertencem aos dois grandes grupos possuem níveis estatísticos parecidos, considerando a CD entre eles.

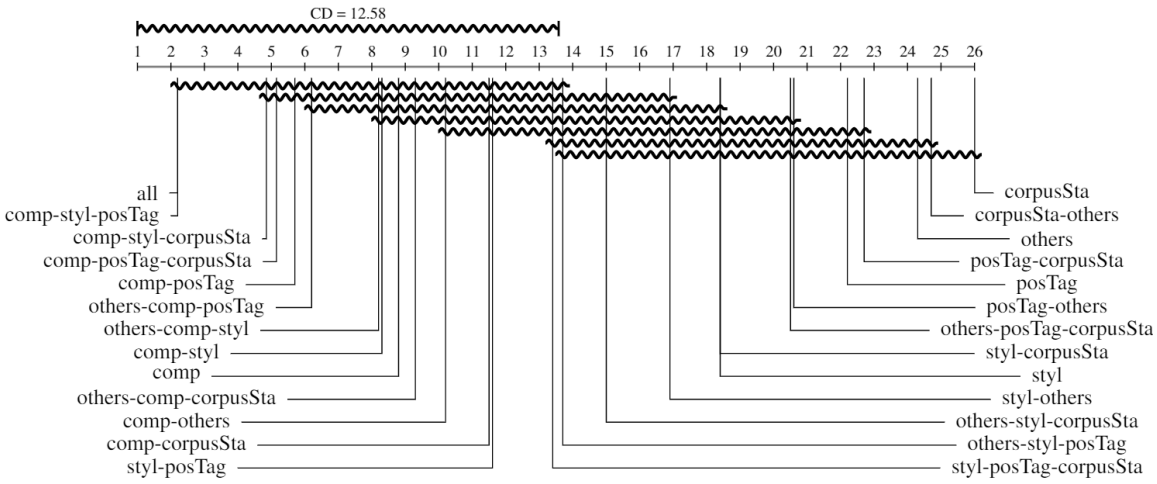

Figura 22 – Comparação dos modelos treinados com cada combinação de grupo de *features* de acordo com o teste Nemenyi, levando em consideração a métrica de *F1-score*. Os resultados dos grupos conectados não são significativamente diferentes (em $\alpha = 0,05$).

6.1 Discussões e Limitações

Após todos os resultados apresentados podemos extrair algumas informações relacionados as nossas descobertas. Nosso DSS se mostrou relevante e a nossa abordagem multirrótulo se mostrou superior às abordagens multi-classe clássicas do estado da arte para o problema. A partir disso, é possível destacar o desempenho do algoritmo *Random Forest* nas abordagens multirrótulo e multi-classe, que obteve os melhores resultados em relação aos outros algoritmos utilizados nos experimentos. Além disso, é importante salientar que a classificação textual é extremamente complicada para a máquina e os resultados apontam que a característica não determinística da RF ajudou no entendimento da abstração deste problema.

A análise de *feature importance* é um ponto considerável em nossa pesquisa, onde a ideia de detectar notícias falsas a partir de características textuais possibilita a independência de características linguísticas de cada idioma. Nosso processo de análise considerou uma gama de possíveis *features* de modo a demonstrar "quais seriam" ou "quais não seriam" relevantes para o problema, e o critério de escolha utilizado foi baseado em trabalhos do estado da arte. Além das 20 *features* utilizadas nos experimentos iniciais, adicionamos 9 *features* para o processo de análise de *feature importance*, com o intuito de balancear e possibilitar a divisão das *features* em 5 categorias de modo a compreender quais categorias mostram-se mais relevantes durante o processo de classificação.

No entanto, nosso trabalho possui algumas limitações importantes, onde o conjunto de dados coletado é limitado e possui poucas amostras, o que torna o estudo não tão abrangente principalmente em relação à acurácia. Outra limitação é a quantidade

de sites extraídos como representantes de cada uma das classes conceituais. Em teoria, a diferença estrutural de cada site pode influenciar nos resultados e para contornar este problema tratamos todo o conjunto de dados na fase de pré-processamento descaracterizando possíveis influências estilométricas de cada site para que uma característica específica não fosse utilizada como parâmetro para as decisões tomadas durante os experimentos.

Embora os resultados tenham demonstrado que a abordagem multirrótulo em nosso DSS é adequada para o problema apresentado, é essencial destacar que a pesquisa trata apenas do idioma português do Brasil. No entanto, os resultados são promissores e oferecem novas perspectivas para pesquisas futuras em diferentes idiomas.

Além disso, nosso estudo não abordou experimentos do nosso DSS proposto em outro domínio, onde focamos apenas na temática política, além do uso de apenas nossos resultados como parâmetro para o processo de análise de *feature importance*. No entanto, é importante observar que as *features* escolhidas são independentes de idioma e podem ser usadas em problemas multi-idiomas, respeitando as particularidades e as possíveis adaptações para cada domínio.

Por fim, a análise de *feature importance* mostrou quais categorias de *features* mostraram-se mais relevantes para o problema abordado neste estudo. E como o problema abordado tem um nicho específico e um conjunto de dados limitado, seus resultados não podem ser tomados como conclusivos para a grande área. Contudo, esta análise abre caminho para análises futuras destas categorias em diferentes conjuntos de dados, idiomas e domínios.

7 CONCLUSÃO

Neste trabalho, propusemos um Sistema de Apoio à Decisão para auxiliar na classificação de notícias considerando duas classes conceituais: objetivo/satírico e legítimo/falso. Além disso, propusemos uma abordagem multirrótulo baseada em algoritmos de transformação de problema (BR e LP) e transformação de algoritmo (ML-k-NN) para enfrentar o desafio de classificar as quatro combinações possíveis destas classes: objetivo-legítimo, objetivo-falso, satírico-legítimo e satírico-falso.

Para tanto, exploramos um cenário realista com base em um conjunto de dados coletado de diferentes sites de notícias brasileiros, com exemplos de cada uma das combinações de classes possíveis e propusemos quatro novas *features* textuais (avgPar, missWordC, missWordR e sumRed) com o intuito de melhorar o desempenho preditivo. Ainda, comparamos o desempenho da abordagem multirrótulo com algoritmos multi-classe do estado da arte (kNN, RF e SVM). Ainda, utilizamos 16 features textuais para a classificação de notícias provindas do estado da arte junto a 4 features textuais propostas.

O algoritmo que obteve o melhor resultado foi o LP_RF, que obteve o maior desempenho nas abordagens multi-classe e multirrótulo. Os melhores desempenhos (*F1-score*) foram alcançados por abordagens multirrótulo, sendo as três pontuações mais altas LP_RF (0,81), BR_RF (0,79) e BR_SVM (0,69) respectivamente prevalecendo sobre o melhor algoritmo multi-classe RF (0,71).

Com isso, podemos afirmar que a abordagem multirrótulo é adequada para este problema, pois se mostrou superior as abordagens multi-classe em diversos cenários, considerando tanto a medição da acurácia quanto de *F1-score*. Além disso, podemos afirmar também que o algoritmo RF é o algoritmo mais adequado para o nosso problema, pois obteve os melhores resultados tanto em sua versão clássica (multi-classe) quanto em sua versão multirrótulo (LP_RF e BR_RF).

Ainda, analisamos a eficácia das features propostas em nosso trabalho, e para isso, utilizamos a RF *importance*, que demonstrou que a feature proposta avgPar superou a importância das *features* tradicionais, seguido das features ratioPUNCT e avgSen, o que mostra que a quantidade de palavras e de sentenças e a frequência de pontuação foram as features mais importantes para descrever o conjunto de dados. A *feature* proposta missWordR também pode ser destacada, pois se encontra entre as *features* de maior relevância do *ranking*.

Finalmente, 9 features textuais foram adicionadas e todo o grupo de features (29 features) foi dividido em cinco categorias, onde cada grupo foi analisado de forma separada e combinada a outros grupos, com o objetivo de analisar como cada grupo influencia

no problema. Os resultados mostraram que as *features* do grupo *Complexity* tem uma influência significativa nos resultados, seguido das *features* do grupo *Stylistic*. Além disso, o grupo de *features Corpus Statistics* mostrou que não é relevante para os resultados, apresentando o pior desempenho em comparação com os outros.

Com os resultados obtidos foi possível analisar que em nosso problema, a complexidade de uma notícia está mais presente em notícias objetivo-legítimo, o que mostra que portais tradicionais de notícias mantém textos maiores e com um vocabulário mais rico e complexo. Ainda, notícias objetivo-falso tendem a utilizar trechos de notícias legítimas com o intuito de trazer maior credibilidade ao conteúdo disseminado. No entanto, a construção textual geralmente não possui vocabulário complexo, demonstrando geralmente um vocabulário mais acessível e superficial.

Notícias satírico-falso apresentaram em seus resultados características bastante parecidas com os resultados obtidos em objetivo-falso. Em contrapartida, notícias satírico-legítimo não demonstram um padrão específico em sua construção, onde os parágrafos são relativamente menores que os encontrados em notícias objetivo-legítimo, além de possuírem uma alta variação em seus valores, tornando sua classificação mais difícil.

Como próximos passos, seria interessante avaliar como o nosso DSS se comporta em outros cenários através de testes com diferentes conjuntos de dados e idiomas, de modo a analisar se a abordagem obtém resultados satisfatórios em condições adversas. Os resultados obtidos neste trabalho não são suficientes para afirmar que sua aplicação em problemas reais garante a detecção precisa de notícias falsas, porém, sua contribuição para o estado da arte pode proporcionar outras perspectivas nas pesquisas futuras relacionadas, onde o auxílio de um especialista humano pode contribuir para o treinamento e entendimento do modelo.

REFERÊNCIAS

[1] CARVALHO, A. et al. Inteligência artificial–uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2011.

[2] SHU, K. et al. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, ACM, v. 19, n. 1, p. 22–36, 2017.

[3] LAZER, D. M. et al. The science of fake news. *Science*, American Association for the Advancement of Science, v. 359, n. 6380, p. 1094–1096, 2018.

[4] PARISER, E. *The filter bubble: What the Internet is hiding from you*. [S.l.]: Penguin UK, 2011.

[5] KRESS, G. *Literacy in the new media age*. [S.l.]: Routledge, 2003.

[6] MCCOMBS, M. E.; SHAW, D. L. The agenda-setting function of mass media. *Public opinion quarterly*, Oxford University Press, v. 36, n. 2, p. 176–187, 1972.

[7] TAYAL, D. K. et al. Polarity detection of sarcastic political tweets. In: IEEE. *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*. [S.l.], 2014. p. 625–628.

[8] PELLICCIARI, V. *Machine Learning: Fundamental Algorithms for Supervised and Unsupervised Learning With Real-World Applications*. first. [S.l.]: Edição Kindle, 2017.

[9] RASCHKA, S. *Python machine learning*. [S.l.]: Packt Publishing Ltd, 2015.

[10] MITCHELL, T. *Machine Learning*. McGraw-Hill Education, 1997. (McGraw-Hill international editions - computer science series). ISBN 9780070428072. Disponível em: <https://books.google.com.br/books?id=xOGAngEACAAJ>.

[11] FIORE, U. et al. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, Elsevier, v. 122, p. 13–23, 2013.

[12] BISHOP, C. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. 738 p.

[13] LUNARDI, A. de C.; VITERBO, J.; BERNARDINI, F. C. Análise de sentimentos utilizando técnicas de classificação multiclasse. *Tópicos em Sistemas de Informação: Minicursos SBSI 2016*, p. 1.

[14] KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, v. 160, p. 3–24, 2007.

[15] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

[16] VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013.

[17] BOSWELL, D. Introduction to support vector machines. *Departement of Computer Science and Engineering University of California San Diego*, 2002.

[18] DEKA, P. C. et al. Support vector machine applications in the field of hydrology: a review. *Applied soft computing*, Elsevier, v. 19, p. 372–386, 2014.

[19] BORCHANI, H. et al. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 5, n. 5, p. 216–233, 2015.

[20] ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2009.

[21] TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, IGI Global, v. 3, n. 3, p. 1–13, 2007.

[22] KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. Multilabel text classification for automated tag suggestion. In: *Proceedings of the ECML/PKDD*. [S.l.: s.n.], 2008. v. 18, p. 5.

[23] CERRI, R. *Técnicas de classificação hierárquica multirrótulo*. Tese (Doutorado) — Universidade de São Paulo, 2010.

[24] TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Data mining and knowledge discovery handbook. *Mining multi-label data*, 2010.

[25] CHERMAN, E. A. *Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo*. Tese (Doutorado) — Universidade de São Paulo, 2013.

[26] GONÇALVES, E. C. Introdução à classificação multirrótulo.

[27] ZHANG, M.-L.; ZHOU, Z.-H. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, IEEE, v. 26, n. 8, p. 1819–1837, 2014.

[28] PÉREZ-MARÍN, D.; PASCUAL-NIETO, I.; RODRÍGUEZ, P. Computer-assisted assessment of free-text answers. *The Knowledge Engineering Review*, Cambridge University Press, v. 24, n. 4, p. 353–374, 2009.

[29] MANNING, C. D.; MANNING, C. D.; SCHÜTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999.

[30] SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. Introduction to information retrieval. In: *Proceedings of the international communication of association for computing machinery conference*. [S.l.: s.n.], 2008. p. 260.

[31] BAJAJ, S. *The Pope Has a New Baby! Fake News Detection Using Deep Learning*. [S.l.]: Tech. rep. Technical Report, Stanford Univ, 2017.

[32] SINGHANIA, S.; FERNANDEZ, N.; RAO, S. 3han: A deep neural network for fake news detection. In: SPRINGER. *International Conference on Neural Information Processing*. [S.l.], 2017. p. 572–581.

[33] ZHOU, X. et al. Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, ACM New York, NY, USA, v. 1, n. 2, p. 1–25, 2020.

[34] SHAO, C. et al. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, arXiv, 2017.

[35] RUCHANSKY, N.; SEO, S.; LIU, Y. Csi: A hybrid deep model for fake news detection. In: ACM. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. [S.l.], 2017. p. 797–806.

[36] MONTEIRO, R. A. et al. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2018. p. 324–334.

[37] RUBIN, V. et al. Fake news or truth? using satirical cues to detect potentially misleading news. In: *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. [S.l.: s.n.], 2016. p. 7–17.

[38] GONZÁLEZ-IBÁNEZ, R.; MURESAN, S.; WACHOLDER, N. Identifying sarcasm in twitter: a closer look. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*. [S.l.], 2011. p. 581–586.

[39] PORIA, S. et al. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*, 2016.

[40] HOSSAIN, M. Z. et al. Banfakenews: A dataset for detecting fake news in bangla. *arXiv preprint arXiv:2004.08789*, 2020.

[41] SHU, K.; MAHUDESWARAN, D.; LIU, H. Fakenewstracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, v. 25, n. 1, p. 60–71, Mar 2019. Disponível em: <https://doi.org/10.1007/s10588-018-09280-3>.

[42] SHU, K. et al. defend: Explainable fake news detection. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, 2019. (KDD '19), p. 395–405. ISBN 978-1-4503-6201-6. Disponível em: <http://doi.acm.org/10.1145/3292500.3330935>.

[43] ISHITA, E. et al. Investigating multi-label classification for human values. *Proceedings of the American Society for Information Science and Technology*, Wiley Online Library, v. 47, n. 1, p. 1–4, 2010.

[44] BHOWMICK, P. K. Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Computer and Information Science*, v. 2, n. 4, p. 64, 2009.

[45] LI, X. et al. Weighted multi-label classification model for sentiment analysis of online news. In: IEEE. *Big Data and Smart Computing (BigComp), 2016 International Conference on*. [S.l.], 2016. p. 215–222.

[46] ALMEIDA, A. M. et al. Applying multi-label techniques in emotion identification of short texts. *Neurocomputing*, Elsevier, v. 320, p. 35–46, 2018.

[47] JUNIOR, J. R. C. Desenvolvimento de uma metodologia para mineração de textos. *Departamento de Engenharia Elétrica, Pontífica Universidade Católica do Rio de Janeiro*, 2007.

[48] IGAWA, R. A. et al. Adaptive distribution of vocabulary frequencies: A novel estimation suitable for social media corpus. In: IEEE. *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. [S.l.], 2014. p. 282–287.

[49] BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009.

[50] SAIF, H. et al. On stopwords, filtering and data sparsity for sentiment analysis of twitter. *Ninth International Conference on Language Resources and Evaluation*, p. 810—-817, 2014.

[51] HORNE, B. D.; ADALI, S. This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *arXiv preprint arXiv:1703.09398*, 2017.

[52] LYNCH, G.; VOGEL, C. The translator's visibility: Detecting translatorial fingerprints in contemporaneous parallel translations. *Computer Speech & Language*, v. 52, p. 79 – 104, 2018. ISSN 0885-2308.

[53] CHEN, Y.; CONROY, N. J.; RUBIN, V. L. Misleading online content: Recognizing clickbait as false news. In: ACM. *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*. [S.l.], 2015. p. 15–19.

[54] QIN, T. et al. Modality effects in deception detection and applications in automatic-deception-detection. In: IEEE. *Proceedings of the 38th annual Hawaii international conference on system sciences*. [S.l.], 2005. p. 23b–23b.

[55] ZHOU, L. et al. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group Decision and Negotiation*, v. 13, n. 1, p. 81–106, Jan 2004. ISSN 1572-9907. Disponível em: <https://doi.org/10.1023/B:GRUP.0000011944.62889.6f>.

[56] CASTILLO, C.; MENDOZA, M.; POBLETE, B. Predicting information credibility in time-sensitive social media. *Internet Research*, Emerald Group Publishing Limited, v. 23, n. 5, p. 560–588, 2013.

[57] REGANTI, A. et al. Open secrets and wrong rights: automatic satire detection in english text. In: ACM. *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. [S.l.], 2017. p. 291–294.

[58] PISKORSKI, J.; SYDOW, M.; WEISS, D. Exploring linguistic features for web spam detection: a preliminary study. In: ACM. *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*. [S.l.], 2008. p. 25–28.

[59] DILLARD, J. P.; PFAU, M. *The persuasion handbook: Developments in theory and practice*. [S.l.]: Sage Publications, 2002.

[60] FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, Springer, v. 21, n. 1, p. 2, 2015.

[61] CONROY, N. J.; RUBIN, V. L.; CHEN, Y. Automatic deception detection: Methods for finding fake news. In: AMERICAN SOCIETY FOR INFORMATION SCIENCE. *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. [S.l.], 2015. p. 82.

[62] LEECH, G.; WEISSER, M. Generic speech act annotation for task-oriented dialogues. In: LANCASTER: LANCASTER UNIVERSITY. *Proceedings of the corpus linguistics 2003 conference*. [S.l.], 2003. v. 16.

[63] BELL, A. *The language of news media*. [S.l.]: Blackwell Oxford, 1991.

[64] BARRIOS, F. et al. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*, 2016.

[65] MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119.

[66] BATISTA, G. E.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, ACM, v. 6, n. 1, p. 20–29, 2004.

[67] CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

[68] AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. *Machine learning*, Springer, v. 6, n. 1, p. 37–66, 1991.

[69] SOROWER, M. S. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, v. 18, 2010.

[70] ZHANG, M.-L.; ZHOU, Z.-H. A k-nearest neighbor based algorithm for multi-label classification. In: IEEE. *Granular Computing, 2005 IEEE International Conference on*. [S.l.], 2005. v. 2, p. 718–721.

[71] KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.

[72] OLSON, D. L.; DELEN, D. *Advanced data mining techniques*. [S.l.]: Springer Science & Business Media, 2008.

[73] FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

[74] GENUER, R.; POGGI, J.-M.; TULEAU-MALOT, C. Variable selection using random forests. *Pattern Recognition Letters*, Elsevier, v. 31, n. 14, p. 2225–2236, 2010.

[75] MAATEN, L. v. d.; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. Nov, p. 2579–2605, 2008.

[76] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

TRABALHOS PUBLICADOS PELA AUTORA

Trabalhos publicados pela autora durante o programa.

Publicações principais do trabalho.

1. Janaína Ignácio de Morais, Hugo Queiroz Abonizio, Gabriel Marques Tavares, André Azevedo da Fonseca, Sylvio Barbon Jr., **Deciding among Fake, Satirical, Objective and Legitimate news: A multi-label classification system**, Proceedings of the XV Brazilian Symposium on Information Systems, 05/2019, ACM, págs. 22-28, ISBN: 978-1-4503-7237-4, (Qualis CC 2019, B1)

2. Hugo Queiroz Abonizio, Janaína Ignácio de Morais, Gabriel Marques Tavares, Sylvio Barbon Jr., **Language-Independent Fake News Detection: English, Portuguese, and Spanish Mutual Features**, Future Internet, 05/2020, MDPI, pág 87, (Qualis CC 2019, B1)

3. Janaína Ignácio de Morais, Hugo Queiroz Abonizio, Gabriel Marques Tavares, André Azevedo da Fonseca, Sylvio Barbon Jr., **A Multi-label Classification System to Distinguish among Fake, Satirical, Objective and Legitimate News in Brazilian Portuguese**, iSys - Brazilian Journal of Information Systems, 07/2020, CESI/SBC, 1-25, (Qualis CC 2019, B3)