UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Douglas Vitor Pontes

O uso de técnicas de Aprendizado de Máquina em grafos para predição da popularidade de atos normativos brasileiros sobre segurança alimentar

Belo Horizonte
2024

Douglas Vitor Pontes

O uso de técnicas de Aprendizado de Máquina em grafos para predição da popularidade de atos normativos brasileiros sobre segurança alimentar

Versão Parcial

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Dr. Adriano Alonso Veloso
Coorientadora: Dra. Fabiana de Menezes Soares

Belo Horizonte
2024

[Ficha Catalográfica em formato PDF]

A ficha catalográfica será fornecida pela biblioteca. Ela deve estar em formato PDF e deve ser passada como argumento do comando `ppgccufmg` no arquivo principal `.tex`, conforme o exemplo abaixo:

```
\ppgccufmg{
      ...
      fichacatalografica={ficha.pdf}
}
```

[Folha de Aprovação em formato PDF]

A folha de aprovação deve estar em formato PDF e deve ser passada como argumento do comando `ppgccufmg` no arquivo principal `.tex`, conforme o exemplo abaixo:

```
\ppgccufmg{
     ...
     folhadeaprovacao={folha.pdf}
}
```

A minha amada mãe Rosa Maria Pontes (in memoriam), que fez tanto por mim ao longo da sua vida e que continua a me guiar de onde quer que esteja.

Agradecimentos

Finalizada uma etapa particularmente importante da minha vida, não poderia deixar de expressar o mais profundo agradecimento a todos os que me apoiaram nesta longa caminhada e contribuíram para a realização deste trabalho.
A Deus, sempre comigo, guiando e protegendo e a quem tanto recorri durante esta jornada.

À minha esposa Roseane e aos meus pequenos, Cecília e Bento pela compreensão, dedicação, apoio e carinho nos momentos de tempestade e por sempre acreditarem em mim. Minha eterna gratidão.

Aos meus pais e irmãos, agradeço pelo amor incondicional, pelo apoio e coragem que sempre me transmitiram.

À minha família linda, minha base, meu refúgio, meu acalanto e meu desassossego.

À Universidade Federal de Minas Gerais (UFMG), ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) por toda infraestrutura oferecida.

Ao meu orientador, Professor Doutor Adriano Alonso Veloso, por sua paciência, ensinamentos e acima de tudo pela confiança no desenvolvimento deste trabalho. Meus sinceros agradecimentos pela oportunidade.

A todos os professores da Pós-Graduação, pelos conhecimentos e competências que me transmitiram ao longo deste percurso acadêmico, que culminaram na elaboração desta tese.

À Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo incentivo a esta pesquisa.

Aos meus amigos que me apoiaram incondicionalmente e entenderam minha ausência.

À Professora Doutora Fabiana de Menezes Soares, por todo o aprendizado, paciência, empatia e atenção ímpar.

Por último, mas não menos importante, agradeço aos meus colegas de Technium, em especial ao Gibram Raul, Vitor Hanriot, Welerson Melo e Welton Augusto pelos momentos disponibilizados e por partilharem do seu saber.

"*Grandes espíritos sempre encontraram violenta oposição de mentes medíocres. A mente medíocre é incapaz de compreender o homem que se recusa a se curvar cegamente aos preconceitos convencionais e escolhe expressar suas opiniões com coragem e honestidade.*"

(Albert Einstein)

Resumo

A crescente demanda e disponibilização de dados em larga escala de atos normativos brasileiros apresentam oportunidades desafiadoras para a sociedade, particularmente na construção de sistemas computacionais que possam aprender, raciocinar e realizar inferências com base em conhecimentos prévios. Nesse contexto, as bases de conhecimento são ativos de extrema importância para a representação e o raciocínio automatizado do conhecimento em diversos domínios de aplicação. Um exemplo relevante é a inferência de informações a partir de sua representação em redes (grafos de conhecimento), que tem ganhado notoriedade acadêmica e industrial nos últimos anos. Diante disso, esta dissertação visa desenvolver um modelo de Inteligência Artificial utilizando técnicas de Aprendizado de Máquina em grafos para a predição da popularidade de atos normativos brasileiros específicos da área de segurança alimentar. O desenvolvimento do projeto exigiu a implementação de um fluxo de processamento (pipeline) estruturado e abrangente, com o intuito de realizar análises detalhadas e produzir resultados relevantes. A base de dados utilizada foi composta por atos normativos brasileiros voltados para a segurança alimentar, obtidos no site oficial do Ministério da Agricultura e Pecuária (MAPA). A coleta dos dados normativos foi realizada por meio da técnica de web scraping, que permitiu a captura estruturada e sistemática de 320 atos normativos em formato PDF. As etapas do estudo incluíram: tratamento dos dados, modelagem de tópicos, criação de matriz de similaridade, construção de grafos e suas características, geração de *embeddings* do grafo, e experimentos com modelos de Aprendizado de Máquina utilizando *embeddings* e features do grafo, além da aplicação do método *SHAP*. Os resultados evidenciam que a análise dos atos normativos através do *BERTopic* permitiu a identificação de tópicos relevantes, enquanto a construção de grafos e a aplicação de técnicas de Aprendizado de Máquina possibilitaram a predição da popularidade desses atos normativos. Conclui-se que a metodologia aplicada não só fornece uma análise detalhada e robusta da popularidade dos atos normativos, mas também contribui significativamente para o campo de pesquisa ao demonstrar a eficácia das técnicas de Aprendizado de Máquina em grafos no contexto jurídico e normativo.

Palavras-chave: Inteligência Artificial. Aprendizado de Máquina. Grafos. Legislação Brasileira. Segurança Alimentar.

Abstract

The growing demand and availability of large-scale data on Brazilian normative acts present challenging opportunities for society, particularly in the construction of computational systems that can learn, reason, and make inferences based on prior knowledge. In this context, knowledge bases are extremely important assets for the representation and automated reasoning of knowledge in various application domains. A relevant example is the inference of information from its representation in networks (knowledge graphs), which has gained academic and industrial notoriety in recent years. Therefore, this dissertation aims to develop an Artificial Intelligence model using Machine Learning techniques in graphs to predict the popularity of Brazilian normative acts specific to the area of food security. The development of the project required the implementation of a structured and comprehensive processing pipeline to perform detailed analyses and produce relevant results. The database used was composed of Brazilian normative acts focused on food security, obtained from the official Ministério da Agricultura e Pecuária (MAPA) website. The collection of normative data was carried out using web scraping techniques, which enabled the structured and systematic capture of 320 normative acts in PDF format. The study stages included: data processing, topic modeling, creation of a similarity matrix, construction of graphs and their characteristics, generation of graph embeddings, and experiments with Machine Learning models using embeddings and graph features, in addition to the application of the SHAP method. The results show that the analysis of normative acts through BERTopic allowed the identification of relevant topics, while the construction of graphs and the application of Machine Learning techniques enabled the prediction of the popularity of these normative acts. It is concluded that the applied methodology not only provides a detailed and robust analysis of the popularity of normative acts but also significantly contributes to the research field by demonstrating the effectiveness of Machine Learning techniques in graphs in the legal and normative context.

Keywords: Artificial Intelligence. Machine Learning. Graphs. Brazilian Legislation. Food Security.

Lista de Figuras

2.1 Pirâmide de Kelsen. Estrutura geral da pirâmide normativa. 20
2.2 Cidade de Königsberg, com suas sete pontes destacadas. 26
2.3 Exemplo de grafos não direcionado e direcionado. 27
2.4 Exemplo de *random walk*. 30
2.5 Ilustração do procedimento da *random walk* no *Node2Vec*. 31
2.6 Blocos, *message Passing*, *Aggregationh*, *Update* para construção do GNN. . . . 33
2.7 Arquiteturas *convolutional*, *attentional* e *message-passing*. 34
2.8 Processo de modelagem de tópicos usando *BERTopic*. O modelo *BERT* é utilizado para gerar *embeddings*, seguido pela redução de dimensionalidade com *UMAP*. A clusterização é realizada com *HDBSCAN* e a importância das palavras em cada tópico é avaliada usando *c-TF-IDF*. 37

4.1 Diagrama do *pipeline* do projeto, incluindo: coleta de dados, conversão de formato, pré-processamento, modelagem de tópicos, construção do grafo, geração de *embeddings* e treinamento do modelo. 42
4.2 Exemplo da pesquisa no Google para o título "PORTARIA SDA Nº 605, DE 23 DE JUNHO DE 2022"mostrando aproximadamente 49.800 resultados. . . . 48

5.1 Nuvem de tags representando os termos mais frequentes nos atos normativos sobre segurança alimentar. Os termos destacados, como "produto", "estabelecimento", "registro"e "instrução normativa", refletem as principais áreas de foco das regulamentações. 52
5.2 Gráficos de barras mostrando os termos-chave de tópicos gerados pelo *BERTopic*, com base nas pontuações *c-TF-IDF*. 53
5.3 Dendrograma ilustrando a clusterização hierárquica dos tópicos gerados pelo *BERTopic*. Os tópicos são agrupados com base na similaridade, facilitando a compreensão das relações entre eles. 54
5.4 Parte do grafo acíclico direcionado mostrando a inter-relação entre atos normativos. Vértices representam atos normativos e arestas indicam tópicos relevantes que conectam esses atos. 55
5.5 *Boxplots* comparando os quartis preditos utilizando *embeddings* de 8 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos. 56

5.6 *Boxplots* comparando os quartis preditos utilizando *embeddings* de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos. 57
5.7 *Boxplots* comparando os quartis preditos utilizando *embeddings* de 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos. 58
5.8 *Boxplots* comparando os quartis preditos utilizando o somatório dos *embeddings* de 8, 64 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*. 60
5.9 *Boxplots* comparando os quartis preditos utilizando a subtração dos *embeddings* de 8, 64 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*. 61
5.10 *Boxplots* mostrando a distribuição dos quartis previstos com base em métricas do grafo: *degree centrality, closeness centrality, load centrality, harmonic centrality, betweenness centrality, average neighbor degree, clustering e pagerank*. . 63
5.11 Gráfico de valores de *SHAP* mostrando a importância das características do grafo: *degree centrality, closeness centrality, load centrality, harmonic centrality, betweenness centrality, average neighbor degree, clustering* e *pagerank*, na previsão do modelo. 64

Lista de Abreviaturas e Siglas

BERT	*Bidirectional Encoder Representations for Transformers*
BERTopic	*BERTopic* é uma técnica de modelagem de tópicos
BFS	*Breadth-First Search*
BOW	*Bag of Words*
c-TF-IDF	*Class-based Term Frequency-Inverse Document Frequency*
DCN	*Deep Convolutional Networks*
DFS	*Depth-First Search*
GNN	*Graph Neural Network*
HDBSCAN	*Hierarchical Density-Based Spatial. Clustering of Applications with Noise*
IA	*Inteligência Artificial*
LDA	*Latent Dirichlet Allocation*
LightGBM	*Light Gradient Boosting Model*
LSTM	*Long Short-Term Memory*
MAE	*Mean Absolute Error*
MAPA	Ministério da Agricultura e Pecuária
NLP	*Natural Language Processing*
Node2Vec	Algoritmo para aprendizado de representações em grafos
PDF	*Portable Document Format*
RNN	*Recurrent Neural Network*
SHAP	*SHapley Additive exPlanations*
SVM	*Support Vector Machines*
TF-IDF	*Term Frequency – Inverse Document Frequency*
TXT	*Text File Format*
UFMG	Universidade Federal de Minas Gerais
UMAP	*Uniform Manifold Approximation and Projection*

Sumário

1 Introdução **15**

1.1 Problema de pesquisa . . . 17
1.2 Objetivos . . . 18
1.2.1 Objetivo Geral . . . 18
1.2.2 Objetivos Específicos . . . 18

2 Referencial Teórico **19**

2.1 Norma Jurídica e atos normativos . . . 20
2.2 Inteligência Artificial . . . 22
2.2.1 Aprendizado de Máquina . . . 23
2.2.2 Processamento de Linguagem Natural . . . 24
2.3 Modelagem de Tópicos . . . 25
2.4 Teoria dos Grafos . . . 26
2.5 *Graph Embedding* . . . 28
2.5.1 *Node2Vec* . . . 29
2.5.1.1 *Random Walks* . . . 30
2.5.1.2 *Search bias a* . . . 30
2.5.1.3 Parâmetro de retorno . . . 31
2.5.1.4 Parâmetro de entrada-saída . . . 32
2.5.1.5 Redes Neurais de Grafos . . . 32
2.5.1.6 *Message passing* . . . 35
2.5.1.7 *Aggregation* . . . 35
2.5.1.8 *Update* . . . 36
2.6 *BERTopic* . . . 37

3 Trabalhos Relacionados **39**

4 Metodologia **41**

4.1 Delineamento da pesquisa . . . 41
4.2 Coleta de dados . . . 42
4.3 Conversão dos Dados . . . 43
4.4 Tratamento dos Dados . . . 43
4.5 Modelagem de Tópicos . . . 44
4.6 Matriz de similaridade . . . 45

4.7 Grafo e suas característica 46
4.8 Geração de *embeddings* do grafo 46
4.9 Tratamento do modelo de *Machine Learning* 47
4.9.1 Experimento 1: utilização de *embeddings* 48
4.9.2 Experimento 2: utilização de *features* do grafo 49
4.9.3 *SHAP* 49

5 Resultados e discussão **51**

6 Conclusão **65**

7 Sugestões para Estudos Futuros **67**

Referências **68**

Capítulo 1

Introdução

No Brasil mais de 5 milhões de regras jurídicas geradas por cerca de 5 mil entes legislativos nas três esferas normativas diferentes - União, Estados e Municípios - com competências concorrentes. Isso causa confusão, tanto por parte da administração pública ao criar leis contraditórias e ambíguas, quanto instabilidade e insegurança jurídica para os cidadãos (Soares, 2009).

O Brasil se caracteriza por intensa atividade legislativa e, talvez por isso, sofre as consequências negativas dessa produção massiva, como falta de planejamento adequado e rigor. Desta maneira, é crucial a vigilância sobre a qualidade da elaboração normativa. A preocupação com a qualidade das leis e dos atos normativos no âmbito dos Poderes Executivo e Legislativo no Brasil tem ganhado relevância na academia, mas ainda se encontra em estágios iniciais de desenvolvimento. Refletir sobre a qualidade desses atos é essencial para garantir a segurança jurídica, promover a transparência do Estado e proteger os direitos e deveres dos cidadãos. Portanto, a discussão sobre a qualidade dos atos normativos no Brasil é fundamental para melhorar o sistema legal e fortalecer o Estado de Direito.

Dentre a seara legislativa, é objeto deste estudo os atos normativos, emitidos e disponibilizados pelo Ministério de Agricultura e Pecuária desta Federação, que tratam sobre segurança alimentar. Esse tema é relevante pois segurança alimentar e desempenha um papel vital para garantir que todos tenham acesso a alimentos seguros, saudáveis e adequados para consumo. As leis que tratam sobre segurança alimentar são voltadas para a proteção da saúde e bem-estar da sociedade.

Uma maneira de avaliar o impacto de uma lei é a frequência ou popularidade com que um ato normativo é buscado e discutido online. Tal popularidade serve como um indicador de relevância, mostrando que um ato normativo está sendo amplamente buscado e discutido online, o que sugere seu interesse para a sociedade. A popularidade também pode ajudar na avaliação do impacto de uma lei ou regulamento, pois atos normativos que geram muita discussão geralmente têm um impacto significativo na vida das pessoas, seja positivo ou negativo. Além disso, a popularidade pode ser um sinal de alto engajamento público, essencial em democracias, onde é importante que os cidadãos estejam atentos e envolvidos nos processos legislativos e regulatórios. Para pesquisadores e legisladores, a

popularidade pode ajudar a priorizar quais atos normativos precisam de maior atenção ou revisão, permitindo uma análise mais detalhada das suas implicações e possíveis melhorias. Porém a popularidade não garante sua qualidade.

Uma das técnicas possíveis para avaliar legislações é o uso de Aprendizado de Máquina. Essa abordagem é particularmente relevante devido à quantidade de entes legisladores, o volume de legislação vigente, a linguagem técnica utilizada, a diversidade de formatação e a constante evolução das leis. Técnicas de Aprendizado de Máquina se mostram eficazes na análise de leis por sua capacidade de processar grandes volumes de dados, automatizar tarefas repetitivas e extrair informações relevantes de textos complexos.

Além disso, algoritmos de Aprendizado de Máquina podem prever resultados de casos judiciais e identificar tendências em decisões anteriores, auxiliando advogados e juízes na tomada de decisões informadas. Isso torna a análise legal mais eficiente, precisa e informada, beneficiando o trabalho jurídico.

Essas técnicas também permitem a personalização de recomendações legais, detecção de anomalias e fraudes, e o aprimoramento da qualidade legislativa. Ao analisar o impacto de leis existentes, o Aprendizado de Máquina fornece insights para melhorias, tornando a criação e aplicação de legislações mais eficazes e eficientes. Isso resulta em maior precisão, eficiência e suporte na análise de complexidades jurídicas.

Associado ao Aprendizado de Máquina, a técnica de Processamento de Linguagem Natural (NLP) possibilita a análise textual automatizada, célere, confiável afim de extrair as *features*. Essas variáveis, presentes intrinsecamente nos dados, será utilizada para a realizar as classificações, além de serem essenciais na identificação de padrões pelo algoritmo pela aprendizagem de máquinas.

Diante deste panorama da intensa atividade legislativa brasileira, e os possíveis conflitos decorrentes, e entendendo a relevância do tema da segurança alimentar, acredita-se que a aplicação das técnicas de NLP e Aprendizado de Máquina podem contribuir na discussão do tema por meio do destaque da popularidade dos atos normativos.

As motivações desta pesquisa foram duas. Uma é pessoal, pois minha filha e eu fomos diagnosticados com doença crônica autoimune de origem genética e mecanismo inflamatório deflagrado pela ingestão alimentar (doença celíaca) em 2016, para qual não há medicamento, tratamento ou cura, apenas controle de danos e sintomas a partir da supressão de alimentos que contenham glúten. Percebemos, no cotidiano, que a segurança alimentar é frequentemente comprometida pela rotulagem incorreta, pela falta de informação adequada sobre a composição dos alimentos e pela ausência de rigor na fiscalização das normas de segurança alimentar. Por essa razão, busquei compreender melhor sobre a elaboração e fiscalização de atos normativos brasileiros relacionados à segurança alimentar.

Em segundo lugar em razão de duas disciplinas isoladas que cursei no Departamento de Ciência da Computação (DCC) da Universidade Federal de Minas Gerais

(UFMG): Aprendizado de Máquina e Processamento de Linguagem Natural, sendo que a segunda me despertou muito interesse. Posteriormente, aprovado como aluno regular do Mestrado, procurei o Professor Dr. Adriano Alonso Veloso, pela referência em trabalhos com NLP.

Neste mesmo período, tive a oportunidade de conhecer a Professora Dra. Fabiana de Menezes Soares, que desenvolvia o Pós-doutorado no Laboratório de Inteligência Artificial (LIA) no DCC/UFMG juntamente com o professor Adriano. A Professora Dra. Fabiana possui um vasto conhecimento sobre legislação alimentar e ao analisar seu projeto de Pós-Doutorado da Professora Dra. Fabiana, intitulado: "Análise preditiva aplicada a sistemas normativos complexos: IA para detecção de riscos ao direito à alimentação", surgiu a oportunidade de juntar o meu cotidiano de restrição alimentar com o uso da IA, como uma forma de extensão dos estudos realizados pela Professora Dra. Fabiana e para entender sobre a elaboração e fiscalização de atos normativos brasileiros relacionados à segurança alimentar.

Debater esse problema é fundamental para melhorar a análise e a gestão dessas normas. Predizer a popularidade dos atos normativos é importante porque permite identificar quais normas têm maior impacto e aceitação entre o público, o que pode orientar políticas públicas e estratégias de comunicação mais eficazes. Além disso, a popularidade dos atos normativos pode influenciar a sua implementação e o cumprimento, tornando a predição uma ferramenta valiosa para a tomada de decisões no âmbito governamental.

1.1 Problema de pesquisa

A utilização de técnicas de Aprendizado de Máquina em grafos pode auxiliar na predição da popularidade de atos normativos brasileiros do MAPA?

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um modelo de Inteligência Artificial com o uso de técnicas de Aprendizado de Máquina em grafos para predição da popularidade de atos normativos brasileiros específicos para a área de segurança alimentar.

1.2.2 Objetivos Específicos

- Revisar a literatura acadêmica relevante para construir uma base teórica sólida sobre a aplicação de técnicas de Aprendizado de Máquina em grafos, focando na predição da popularidade de atos normativos brasileiros relacionados à segurança alimentar;
- Coletar e organizar todos os atos normativos emitidos pelo Ministério da Agricultura, Pecuária e Abastecimento (MAPA), que serão utilizados como dados para a análise;
- Aplicar o método *BERTopic* para identificar e agrupar tópicos presentes nos atos normativos, facilitando a análise temática desses documentos;
- Construir um grafo onde os nós representam os atos normativos e as arestas representam os tópicos identificados e a relação entre esses atos normativos, utilizando a biblioteca *NetworkX*;
- Empregar técnicas de *Graph Neural Networks* (GNN) e *Node2Vec* para gerar *embeddings*, que são representações vetoriais dos grafos, capturando a estrutura e as características dos nós e arestas; e
- Desenvolver e treinar um modelo de Aprendizado de Máquina que utilize os *embeddings* gerados e *features* derivadas dos grafos para prever a popularidade dos atos normativos, classificando-os em quartis.

Capítulo 2

Referencial Teórico

Haja vista que os pontos focais para o entendimento desta pesquisa são correlacionar o que é segurança alimentar, atos normativos e Inteligência Artificial, os temas serão abordados a seguir, para que seja possível franquear embasamento teórico aos resultados alcançados.

Segurança alimentar é um conceito crucial que envolve garantir que todas as pessoas tenham acesso a alimentos seguros, nutritivos e em quantidade suficiente para uma vida saudável. Este tema é de suma importância devido aos seus impactos na saúde pública, qualidade de vida das populações e sustentabilidade ambiental. A falta de segurança alimentar pode levar a má nutrição, obesidade, doenças crônicas e deficiências de micronutrientes. Garantir alimentos de qualidade previne doenças e fraudes alimentares, protegendo especialmente grupos vulneráveis como crianças e idosos. Práticas sustentáveis são essenciais para preservar recursos e garantir o futuro alimentar. A relevância da segurança alimentar é amplamente discutida na literatura científica, destacando a interconexão entre saúde, desenvolvimento sustentável e políticas públicas eficientes.

A relevância da segurança alimentar é amplamente discutida na literatura científica, com diversos estudos sublinhando sua importância para a saúde pública e o desenvolvimento sustentável. A Organização das Nações Unidas para a Alimentação e a Agricultura (FAO), em seu relatório de 2019, aborda de forma abrangente a interconexão entre segurança alimentar, saúde e desenvolvimento sustentável, ressaltando a necessidade de sistemas alimentares resilientes e inclusivos para garantir o acesso universal a alimentos nutritivos.

Pinstrup-Andersen (2009) destaca a complexidade da segurança alimentar, considerando fatores como a disponibilidade de alimentos, o acesso econômico e físico a esses alimentos, a utilização nutricional adequada e a estabilidade desses elementos ao longo do tempo. Seu trabalho enfatiza a importância de políticas públicas eficazes que possam mitigar os riscos associados à insegurança alimentar, especialmente em regiões vulneráveis.

Godfray *et al.* (2010) exploram os desafios e estratégias para alimentar uma população global crescente, abordando questões como a intensificação sustentável da agricultura, a redução de perdas e desperdícios de alimentos, e a adaptação às mudanças climáticas. Este estudo enfatiza a necessidade de políticas integradas que promovam a segurança alimentar em um contexto de pressões ambientais crescentes, propondo soluções

inovadoras e colaborativas para garantir a sustentabilidade dos sistemas alimentares globais.

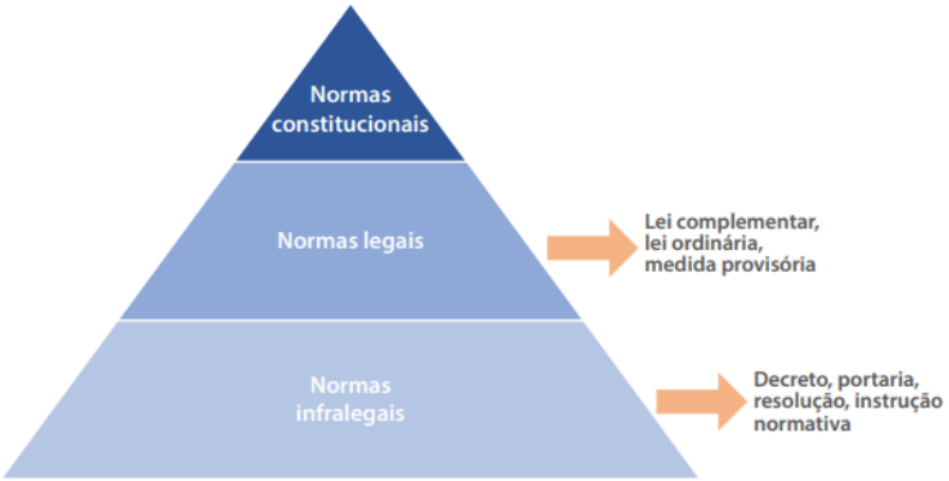
Esses estudos, em conjunto, fornecem uma visão abrangente e multidimensional da segurança alimentar, destacando a necessidade de abordagens holísticas e interdisciplinares para enfrentar os desafios contemporâneos e futuros nesta área crucial.

2.1 Norma Jurídica e atos normativos

No contexto jurídico brasileiro, os textos produzidos são organizadas de maneira hierárquica, em que normas de diferentes níveis de importância e abrangência se sobrepõem ordenadamente, garantindo a coerência e a estabilidade do sistema legal. A hierarquia das normas é essencial para compreender como os atos normativos infralegais, objeto deste estudo, se inserem no arcabouço jurídico e sua importância para a efetivação das políticas públicas e a regulação de diversas atividades sociais.

Segundo Kelsen, as normas hierarquizadas se apresentam em formato de pirâmide, na qual as normas inferiores devem respeitar as normas superiores. No ápice da pirâmide está a Constituição da República Federativa do Brasil de 1988, que é o fundamento de validade de todo o ordenamento jurídico.

Figura 2.1: Pirâmide de Kelsen. Estrutura geral da pirâmide normativa.



Fonte: Manual de elaboração de atos normativos. Ministério da Saúde (2021)

A Constituição Federal de 1988 ocupa o topo da pirâmide normativa no Brasil, sendo a norma fundamental que confere validade a todas as demais normas do sistema. Abaixo da Constituição estão as emendas constitucionais, que são instrumentos destinados a alterar ou complementar o texto constitucional. Em um nível inferior, encontram-se

as leis complementares, leis ordinárias e leis delegadas, que são elaboradas pelo Poder Legislativo e têm como objetivo regulamentar matérias de competência da União.

Seguindo na estrutura hierárquica, encontramos as medidas provisórias, que são editadas pelo Presidente da República em situações de relevância e urgência, com força de lei, mas sujeitas à posterior aprovação pelo Congresso Nacional. Em um patamar subsequente, estão os decretos legislativos e as resoluções, que também são atos do Poder Legislativo, porém com funções específicas e limitadas.

Os atos normativos infralegais, que incluem decretos, portarias, instruções normativas, resoluções administrativas e outros atos administrativos, situam-se na base da pirâmide normativa. Embora ocupem uma posição hierárquica inferior, esses atos desempenham um papel crucial na aplicação e execução das leis, detalhando procedimentos, estabelecendo regras complementares e adaptando as normas gerais às especificidades de diferentes contextos administrativos e setoriais.

Os decretos são editados pelo Presidente da República ou por autoridades delegadas e têm a função de regulamentar as leis, especificando as condições de sua aplicação. As portarias, por sua vez, são expedidas por ministros de Estado e outras autoridades administrativas, visando detalhar e orientar a execução de políticas públicas e a administração de serviços públicos. As instruções normativas e as resoluções administrativas são emitidas por órgãos e entidades da administração pública para orientar a atuação de seus servidores e assegurar a uniformidade e eficiência dos procedimentos administrativos.

A importância dos atos normativos infralegais reside em sua capacidade de conferir agilidade e flexibilidade à administração pública, permitindo que o Poder Executivo ajuste as normas às realidades dinâmicas e complexas do dia a dia administrativo. Esses atos complementam as leis e viabilizam sua execução prática, assegurando a implementação eficaz das políticas públicas e a prestação de serviços à sociedade.

Em síntese, a compreensão da hierarquia das normas e da função dos atos normativos infralegais é fundamental para a análise e o desenvolvimento de qualquer estudo jurídico. Esse entendimento é particularmente relevante quando se trata de áreas específicas e cruciais, como a segurança alimentar.

A importância dos atos normativos infralegais reside em sua capacidade de conferir agilidade e flexibilidade à administração pública, permitindo que o Poder Executivo ajuste as normas às realidades dinâmicas e complexas do dia a dia administrativo. Esses atos complementam as leis e viabilizam sua execução prática, assegurando a implementação eficaz das políticas públicas e a prestação de serviços à sociedade.

Em síntese, a compreensão da hierarquia das normas e da função dos atos normativos infralegais é fundamental para a análise e o desenvolvimento de qualquer estudo jurídico. Esse entendimento é particularmente relevante quando se trata de áreas específicas e cruciais, como a segurança alimentar.

Na pesquisa, foram utilizadas as normas jurídicas Decreto-Lei, Lei, Portaria, Ins-

trução Normativa, Decreto Legislativo, Medida Provisória, Decreto, Ato, Diário Oficial, Memorando Circular, Decisão, Norma Operacional, Protocolo, Resolução MERCOSUL, e Ofício-Circular, além de uma cartilha contendo orientações sobre a inscrição de espécies no RNC.

2.2 Inteligência Artificial

Não há na literatura uma definição clara a respeito de Inteligência Artificial (IA). Esta indefinição reside nas diversas dimensões em que este tema pode ser abordado. Segundo Norvig; Russell (2013), IA pode ser definida ao longo de duas dimensões: as que se relacionam ao processo de pensamento e raciocínio e as que se referem ao comportamento. Assim, foi adotada a definição de Nilsson (1998) "IA... está relacionada a um desempenho inteligente de artefatos", associada à dimensão do comportamento inteligente.

O desenvolvimento da AI tem experimentado vários ciclos de avanço desde seu início em 1943. Atualmente, as principais forças motrizes de avanço da IA são dados, processadores e algoritmos de aprendizado (Xuejiao; Xiaofeng; Yang, 2013). Desde 2010, a quantidade de dados produzidos no mundo atingiu o nível de zettabyte (ZB). O surgimento de processadores específicos melhorou a eficiência do processamento de dados de IA. Estes processadores aceleram a velocidade de treinamento e iteração dos cálculos e promovem o desenvolvimento da indústria de IA. Conforme Farid (2017) o computador infere características do próprio objeto a partir do banco de dados e, em seguida, identifica o objeto de acordo com a regra da característica. Segundo este mesmo autor, este aspecto permite a eliminação de gargalos de processamento da IA.

Um atributo que está diretamente vinculado ao que a IA visa alcançar é a criação de sistemas que possam perceber seu ambiente e, consequentemente, tomar medidas para aumentar as chances de sucesso (Dopico *et al.*, 2016). Como as formas de consumo estão mudando, os fabricantes buscam concentrar-se em demandas cada vez mais individualizadas para alcançar o máximo de clientes em potencial. Assim, o ambiente industrial também deve tornar-se variável. Portanto, a indústria precisa fornecer uma linha de produção dinâmica, onde não apenas os produtos são feitos, mas uma combinação de produtos e serviços são oferecidos para obter vantagem contra seus concorrentes, o que leva a produção a mudar constantemente (Lee; Wang; Su, 2015).

Para conseguir isso, deve-se buscar um grau de automação flexível, onde a computação sensível e coleta de informações do ambiente, deve poder prever as próximas etapas da produção com quase nenhuma interação com o operador, da mesma maneira que a IA preconiza (Zhang *et al.*, 2019). Devido ao avanço das tecnologias de IA e

o desenvolvimento contínuo da fabricação industrial inteligente no mundo, as empresas começaram gradualmente a integrar as tecnologias de IA às atividades industriais. Este processo de integração é chamado de Inteligência Artificial Industrial (*Industrial Artificial Intelligence* – IAI) (Zhang *et al.*, 2019). Da perspectiva industrial, é possível definir a IAI a partir dos requisitos de aplicação industrial, tecnologias e funções da IA. Funções inteligentes estão ligadas a capacidade dos softwares de aprendizado e previsibilidade das próximas etapas, cuja tecnologia é conhecida como Aprendizado de Máquina, uma das linhas de estudo da AI, conforme apresentado a seguir.

2.2.1 Aprendizado de Máquina

Aprendizado de Máquina (do inglês, *Machine Learning*) é uma subárea da Inteligência Artificial (IA), campo de pesquisa centrado na interseção de áreas como Estatística e Ciência da Computação e pode ser vista como sendo uma área de estudos que objetiva a criação de sistemas de computação capazes de realizar tarefas de forma inteligente (Carvalho; Pereira; Cardoso, 2019). Esse termo foi proposto inicialmente em 1956 pelo pesquisador John McCarthy. Apesar da simplificação do termo aqui proposta, esse é um conceito difícil de ser definido, principalmente porque a área está em constante evolução e o entendimento sobre o que ele significa evolui ao longo do tempo.

Outra razão para a dificuldade em definir IA é a natureza interdisciplinar do campo. Antropólogos, biólogos, cientistas da computação, linguistas, filósofos, psicólogos e neurocientistas contribuem para o campo da IA, e cada grupo traz sua própria perspectiva e terminologia (Luckin *et al.*, 2016). A discussão se aprofunda ainda mais e se torna filosófica quando tentamos definir o que significa ser "inteligente". Uma boa definição para "ser inteligente" é "ser racional". Assim, um sistema é inteligente e, ao mesmo tempo, racional, se "faz tudo certo" com os dados que tem (Russel, 2004). A Inteligência Artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana (Gomes, 2010).

De uma forma geral, algoritmos de Aprendizado de Máquina podem ser vistos como sendo funções que buscam fazer o mapeamento entre um conjunto de características, utilizadas como entrada, para extrair algum tipo de aprendizado. Os algoritmos de Aprendizado de Máquina são frequentemente divididos em dois grupos: Aprendizado Supervisionado e Aprendizado Não-Supervisionado. Na primeira classe, os algoritmos que possuem a propriedade de utilizar rótulos previamente conhecidos para induzir funções que relacionem o conjunto de características de entrada com o atributo alvo.

Seja X o espaço de entrada e Y o espaço de saída, o objetivo do Aprendizado

Supervisado é aprender uma função $f : X \rightarrow Y$ (Luxburg; Schölkopf, 2011). Na segunda classe, os algoritmos relacionados com Aprendizado Não-Supervisionado lidam apenas com o espaço de entrada X, uma vez que os rótulos das instâncias não são conhecidos (exemplos não são rotulados). Este tipo de algoritmo é utilizado para clusterização ou agrupamento de dados e redução da dimensão do espaço de entrada. Ainda sobre Aprendizado Supervisionado, os algoritmos são comumente divididos em duas subcategorias, conforme a natureza do problema em que serão utilizados:

- **Algoritmos de Classificação:** utilizados em problemas onde o atributo alvo pode ser descrito por classes ou assumem valores discretos. O principal exemplo de problema dentro dessa subcategoria é a classificação binária, onde existem duas classes possíveis para rotulagem das instâncias. Exemplos de algoritmos: árvores de decisão e *support vector machines*.
- **Algoritmos de Regressão:** aplicados em problemas onde o atributo alvo é um valor numérico (contínuo). O objetivo dos algoritmos é induzir funções (lineares ou não-lineares) que aproximem ao máximo os atributos de entrada à variável de saída. Um exemplo de problema dentro dessa subcategoria é a previsão do preço de ações ou previsão de temperatura. Exemplos de algoritmos: regressão linear e regressão logística.

2.2.2 Processamento de Linguagem Natural

Segundo Chowdhury (2005), o Processamento de Linguagem Natural, comumente conhecido como NLP (em inglês *Natural Language Processing*), é uma área de pesquisa e exploração de mecanismos que possibilitam a manipulação de texto falado e escrito pelos computadores. Uma definição mais formal que denota o termo NLP em (Liddy, 2001) expressa que NLP é um grupo de técnicas computacionais para analisar e representar naturalmente um ou mais níveis de análise linguística a fim de alcançar uma aparência humana no processamento da língua em várias tarefas e aplicações.

De fato, a língua é um mecanismo bastante variado, dependente da geografia e vasto para ser facilmente compreendido pelas máquinas, gerando um interesse não só na reprodução no interior dos computadores para compreensão do que se é dito, mas também na reprodução da língua em aplicações, como os famosos *chatbots*.

2.3 Modelagem de Tópicos

A modelagem de tópicos engloba um conjunto de algoritmos baseados em modelos estatísticos, que processam documentos para identificar suas estruturas temáticas em tópicos, podendo envolver ainda uma análise de como estes se relacionam uns com os outros e como se alteram em cada intervalo de tempo (Blei, 2012). Esses algoritmos partem da perspectiva de que um documento pode ser compreendido como uma distribuição de tópicos, enquanto estes representam uma distribuição de palavras. Mais especificamente, os tópicos consistem em estruturas latentes num documento que podem ser reveladas por modelos de Aprendizado de Máquina e, assim, permitem descrever por meio de um conjunto de palavras as temáticas tratadas no documento analisado (Blei; Jordan, 2003; Blei, 2012).

Existem diversas técnicas computacionais apropriadas para a modelagem de tópicos, sendo uma das mais conhecidas a *Latent Dirichlet Allocation* (LDA), um algoritmo de Aprendizado de Máquina Não-Supervisionado que utiliza como base um modelo probabilístico generativo para captar a permutabilidade de palavras e documentos, sem considerar a ordem que os documentos são apresentados nem das palavras que aparecem em cada documento, baseando-se, então, na representação de *BOW*. Assim, a LDA parte de um número pré-definido de tópicos e atribui – por meio de tratamentos estatísticos – cada palavra a um ou mais tópicos, e cada tópico a um ou mais documentos (Blei; Jordan, 2003; Murphy, 2014).

Apesar de ser uma técnica muito utilizada, a LDA pode ser insuficiente quando a pesquisa possuir como finalidade uma análise mais detalhada das relações existentes na ocorrência de tópicos, já que ela parte do pressuposto que a ordem dos documentos e das palavras não importam, e por não observar relações além da pura modelagem de tópicos (Blei, 2012).

Diante dessas limitações, pode-se levar em consideração algoritmos que estendem a LDA, diminuindo suas pressuposições e aumentando as variáveis de análise (Blei, 2012). Alguns exemplos desses modelos são o *Correlated Topic Model* (Blei; Lafferty, 2005), que observa a correlação entre os tópicos, e o *Dynamic Topic Model* (Blei; Lafferty, 2006), que leva em consideração a ordem dos documentos e analisa as mudanças que ocorrem em cada tópico no decorrer do tempo.

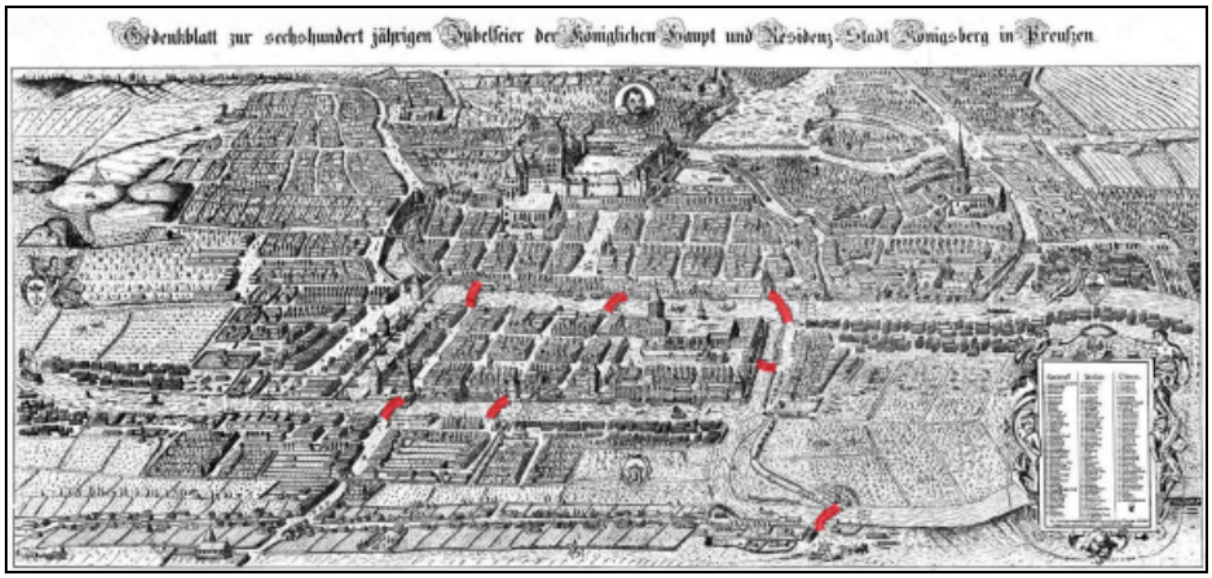
Existem ainda técnicas mais complexas que vão além das modelagens exclusivamente probabilísticas citadas acima e fazem uso de *embeddings*, obtidos por meio de modelos de linguagem baseados em *transformers*, para agrupar documentos com base na similaridade semântica existente entre eles, adicionando como variável o contexto que cada palavra se insere no documento. Exemplos de aplicações desse tipo são o *Top2Vec* (Angelov, 2020), o *BERTopic* (Grootendorst, 2022) e o *Combined Topic Model* (Bianchi;

Terragni; Hovy, 2021). Essas técnicas neurais podem aumentar a interpretabilidade dos tópicos, já que suas palavras-chave passam a conter informações contextuais latentes – inexistentes na abordagem com *BOW*.

2.4 Teoria dos Grafos

Teoria de Grafos teve sua primeira aparição com o famoso problema das sete pontes de Königsberg, enfrentado por Leonhard Euler (1707-1783), um renomado matemático e geômetra. Durante sua estadia em Königsberg (atualmente Kaliningrado), ele deparou-se com esse desafio aparentemente simples, mas de solução elusiva até então. O problema das sete pontes, ilustrado na Figura 2.2, consistia em determinar se era viável percorrer todas as pontes sem repeti-las e retornar ao ponto de partida a partir de terra firme.

Figura 2.2: Cidade de Königsberg, com suas sete pontes destacadas.



Fonte: Goldbarg; Goldbarg (2012).

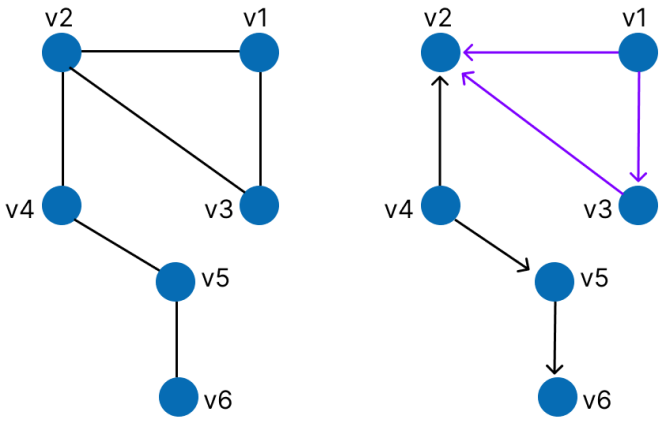
Euler, ao associar pontes a arestas e regiões de terra firme a vértices, concluiu que o grafo correspondente teria que ser um grafo euleriano, onde a condição fundamental é que todos os vértices possuem grau par, ou seja, cada vértice deve ter um número par de conexões.

Formalmente, um Grafo G é um conjunto de três elementos $\{\mathrm{V(G)}, \mathrm{E(G)}, \psi G\}$, onde V(G) é um conjunto não vazio e finito de vértices e $|V(G)| = n$ e $|E(G)| = m$, sendo $E(G)$ um conjunto disjunto de $V(G)$ que representam as arestas e ψG sendo uma função incidente que associa com cada aresta de G um par não ordenado de vértices de G que

não precisam ser distintos. Um exemplo pode ser visto abaixo:

$V(G) = \{v1, v2, v3, v4, v5, v6\}$
$\mathrm{V(E)} = \{\mathrm{e1}, \mathrm{e2}, \mathrm{e3}, \mathrm{e4}, \mathrm{e5}, \mathrm{e6}\}$
$\psi G(e1) = v1v2, \psi G(e2) = v1v3, \psi G(e3) = v3v2, \psi G(e4) = v4v2, \psi G(e5) = v4v5, \psi G$
$(e6) = v5v6$

Figura 2.3: Exemplo de grafos não direcionado e direcionado.



Fonte: Elaborado pelo autor.

Os grafos podem ser ou não ser direcionados. Em grafos direcionados, as arestas possuem uma direção representada por uma seta indicando o vértice em que a aresta incide. Em grafos não direcionados as arestas não possuem direção e indicam um relacionamento mútuo. A Figura 2.3 (esquerda) mostra um exemplo de um grafo não direcionado e a Figura 2.3 (direita) mostra um exemplo do mesmo grafo, porém direcionado. Um estudo mais detalhado pode ser visto em J.A. Bondy e U.S.R. Murty, 1976. Normalmente, utiliza-se uma representação gráfica (geométrica) de um grafo.

Em um grafo direcionado, considera-se um ciclo como um caminho fechado direcionado. Um grafo direcionado é cíclico se os vértices inicial e final coincidirem, caso contrário, é denominado grafo direcionado acíclico, não formando ciclo conforme as setas roxas indicadas na Figura 2.2 (Aloise; Cruz, 2001).

Um grafo G é considerado bipartido quando se é possível particionar o seu conjunto V de vértices em dois subconjuntos V_1 e V_2, de modo que cada aresta deve possuir, obrigatoriamente, uma ponta em cada um destes dois subconjuntos. Geralmente é utilizada a notação $G = (V_1 \cup V_2, E)$ para representar um grafo bipartido (Szwarcfiter, 1984). Para determinar se um grafo é bipartido deve-se verificar se os ciclos contidos em G possuem comprimento par, pois, caso contrário, o caminho não terminaria no mesmo vértice de início, já que em cada passo da travessia o vértice atual está no conjunto oposto ao que se encontra o vértice anterior, e, portanto, não é possível chegar ao vértice de início com um

ciclo de comprimento ímpar. Um grafo é denominado bipartido completo quando existe uma aresta conectando cada um dos vértices dos conjuntos $V_1 e V_2$ (Szwarcfiter, 1984).

2.5 *Graph Embedding*

Graph embedding é o processo de transformação de um grafo num conjunto de vetores espaciais de baixa dimensionalidade. O espaço gerado pelo conjunto de vetores preserva propriedades dessa rede, de modo a garantir que nós mais similares, que compartilhem relações, fiquem mais próximos entre si nesse novo espaço. Há uma série de motivos para uso do *embedding*, em contrapartida, de aplicação de métodos sobre o grafo (Godec, 2018).

O primeiro motivo é que há todo um universo já amplamente estudado e um grande conjunto de ferramentas para aplicação de algoritmos em vetores (*embedding*) devido à bagagem de conhecimento acumulado pelos anos de estudo sobre vetores e suas propriedades que possibilitaram alcançar bons resultados em algoritmos que trabalham com esses objetos matemáticos em detrimento de grafos. Outro ponto positivo para uso de *embedding* é sua representação comprimida, no qual comumente exige menor espaço de armazenamento e, por conseguinte, menor tempo de processamento. Além disso, os *embeddings* por serem vetores possibilitam operações matemáticas rápidas e simples, como soma e multiplicação por escalar, se comparadas com operações sobre grafos.

Apesar das vantagens do uso de *embedding*, este possui uma série de desafios que apesar de estarem sendo superados, não estão ainda suplantados. Por ser um *embedding*, há a necessidade de representação num objeto menor, o vetor resultado do processo, o nó ou o grafo de entrada sem que haja grande perda da informação presente. Devido a isso são usadas uma série de considerações de conceitos presentes em grafos e busca-se aplicá-los para uma melhor representação do grafo. Além desses conceitos, o uso de informações como pesos nas arestas do grafo ou a presença de atributos nos nós e relacionamentos podem ser primordiais para um melhor resultado de *embedding*.

Outro ponto desafiador é a eficiência desses algoritmos em grafos de grande dimensão. Grafos muito grandes podem exigir muito esforço computacional para percorrer cada nó/subgrafo presente no grafo e gerar n caminhos aleatórios para representação em vetores. Ainda seguindo nessa área, maiores *embeddings* preservam mais informação, porém induzem mais espaço e tempo de complexidade (128 e 256 são usualmente usados). Em virtude disso, eficiência no consumo de tempo e espaço são questões relevantes no âmbito de geração de *embedding* em grafos. É importante salientar que existe na literatura uma sobrecarga do termo graph *embedding*. Isso se deve pela existência de duas

categorias de *embeddings* que podem ser realizados sobre grafos.

A primeira categoria corresponde a conversão de cada nó do grafo num vetor de espaço vetorial R^t, sendo t um número inteiro positivo. Essa abordagem é a tradicional, usada nas maiorias das situações, e é conhecida pelo vertex/node *embedding*. Enquanto o último, uma abordagem mais recente, o *embedding* é realizado ao nível de subgrafos (não nós), os transformando em vetores de R^t. Tal abordagem pode ser encontrada no mapeamento de compostos químicos, no qual sua natureza pode ser interpretada como um grafo. Contudo, pelas áreas não terem se desenvolvido ao mesmo tempo, o termo graph *embedding* era inicialmente empregado somente para *embeddings* sobre nós. Devido a isso, se usa até hoje o termo *graph embedding* para representá-la.

No contexto da complexa produção legislativa do Brasil, este estudo se propõe a analisar os desafios relacionados à avaliação da popularidade e qualidade dos atos normativos no país, com um foco especial na aplicação de técnicas de Aprendizado de Máquina.

2.5.1 *Node2Vec*

O algoritmo *Node2Vec* foi desenvolvido por Grover e Leskovec (2016), para o aprendizado de características em grafos. Este método é baseado em *random walks*, que foram propostos no método DeepWalk (Perozzi; Al-Rfou; Skiena, 2014). No *DeepWalk*, introduziu-se o conceito da arquitetura *Skip-gram*, proposta em Mikolov *et al.* (2013) e Perozzi, Al-Rfou e Skiena (2014) para métodos de aprendizagem de características no contexto da linguagem natural. Neste método, cada palavra corrente (atual) é usada como entrada para um classificador linear, que prediz as palavras a uma certa distância antes e depois da palavra corrente. O *Node2Vec* é uma adaptação desta arquitetura, usada para o aprendizado de características no contexto de grafos.

Feito isso, também foi necessário modificar as *random walks* para maximizar a extração de características em grafos, gerando o conceito de biased *random walk*. E essas biased *random walks* são usadas para o aprendizado de poderosas representações vetoriais dos nós de grafos.

O *Node2Vec* foi desenvolvido para fazer amostragens da vizinhança de modo flexível alternando entre BFS e DFS, e tal objetivo é obtido por meio de *biased random walk* que explorar a vizinhança no modo de BFS assim como DFS.

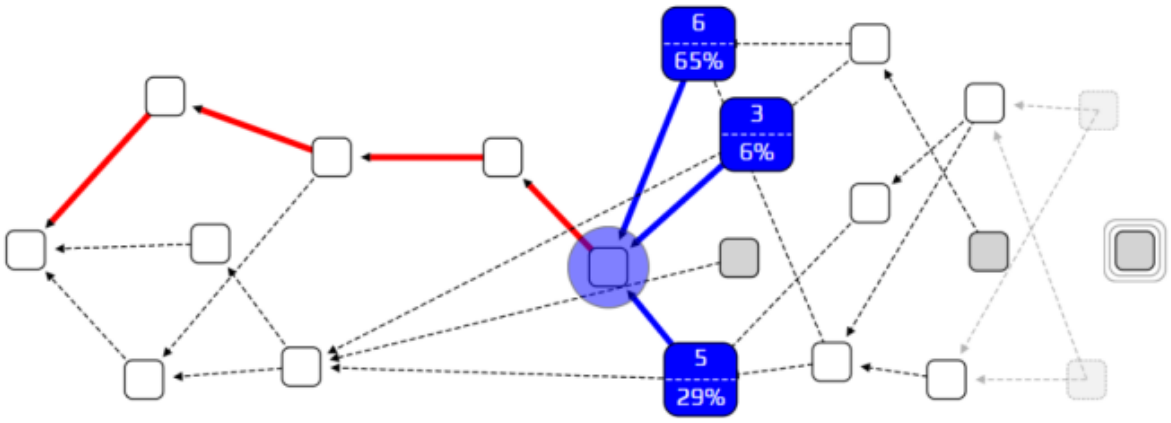
2.5.1.1 *Random Walks*

Dado um nó inicial u, uma simulação de *random walk* de tamanho fixo l é feita. Sendo c_i o i-ésimo nó da *walk*, partindo de $c_0 = u$. Os nós c_i são gerados pela distribuição na equação abaixo, onde π_{vx} é a probabilidade de transição não normalizada entre os nós v e x, e Z é a constante de normalização.

$$P\left(c_i = x \mid c_{i-1} = v\right) = \begin{cases} \frac{\pi_{vx}}{Z}, & \text{se } (v,x) \in E \\ 0, & \text{caso contrário} \end{cases} \tag{2.1}$$

A Figura 2.4 demostra um exemplo de *random walk*, cada transição em azul mostra o peso relacionado à aresta e a probabilidade de transição da escolha do próximo passo da *walk*.

Figura 2.4: Exemplo de *random walk*.



Fonte: Gal (2018).

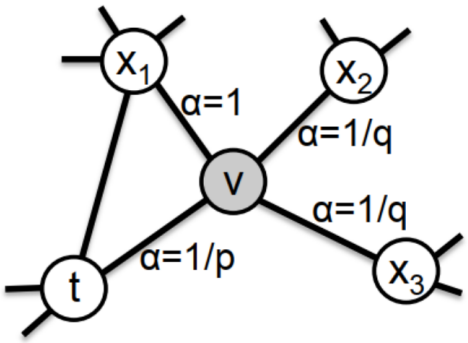
2.5.1.2 *Search bias a*

O modo mais simples de fazer uma *biased random walk* seria mostrar o próximo nó baseado no peso das arestas w_{vx}, por exemplo $\pi_{vx} = w_{vx}$. Entretanto, tal modo não levaria em conta a estrutura do grafo e levaria a busca a explorar diferentes vizinhanças no grafo. Adicionalmente, diferentemente da BFS e da DFS que são paradigmas de amostragem adequados para *structural equivalence* e *homophily* respectivamente, e as *random walks*

não devem se acomodar com essas noções de equivalência, pois grafos do mundo real normalmente exibem uma mistura de ambas.

Uma *random walk* de segunda ordem é defina em Grover e Leskovec (2016) com dois parâmetros p e q que guiam a *walk*: considerando a *random walk* que acaba de cruzar a aresta (t, v) e atualmente está no nó v (Figura 2.5).

Figura 2.5: Ilustração do procedimento da *random walk* no *Node2Vec*.



Fonte: Grover e Leskovec (2016).

A *walk* agora precisa decidir o próximo passo, e então as probabilidades de transição w_{vx} são avaliadas nas arestas (v, x) partindo de v. Logo define-se a probabilidade de transição não normalizada como $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$, onde $\alpha_{pq}(t, x)$ (segundo a equação abaixo) e d_{tx} denotando o menor caminho entre os nós t e x. Observa-se que d_{tx} deve ser 0, 1 ou 2, e consequentemente os dois parâmetros p e q são necessários e suficientes para guiar a *walk*.

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p}, & \text{se } d_{tx} = 0 \\ 1, & \text{se } d_{tx} = 1 \\ \frac{1}{q}, & \text{se } d_{tx} = 2 \end{cases} \tag{2.2}$$

Intuitivamente nota-se que os parâmetros p e q controlam o quão rápido a *walk* explora e deixa a vizinhança do nó inicial u. Mais precisamente, os parâmetros que o procedimento de busca definido interpole entre um BFS e uma DFS e assim torna o procedimento ter mais afinidade para diferentes noções de equivalência de vértices.

2.5.1.3 Parâmetro de retorno

O parâmetro p controla a probabilidade de revisitar imediatamente a nó em uma *walk*. Setá-lo com um alto valor ($> max(q, 1)$) garante que é menos provável de mostrar

um nó já visitado nos próximos dois passos (a menos que o próximo vértice não tenha vizinho). Essa estratégia incentiva uma moderada exploração do grafo e evita redundância 2-hop na amostragem. Caso contrário, se p é pequeno ($< min(q, 1)$), é provável que a *walk* retroceda um passo e mantém a *walk* localmente próxima do nó de partida u.

2.5.1.4 Parâmetro de entrada-saída

O parâmetro q permite que a busca diferencie os nós "internos"e "externos". Nós "internos"referem-se aos nós que estão mais próximos ao nó de origem ou ao nó atualmente considerado na caminhada, enquanto nós "externos"referem-se aos nós que estão mais distantes do nó de origem ou do nó atualmente considerado na caminhada. Olhando a Figura 2.5, se $q > 1$ a *random walk* é provável de explorar os nós próximos de t. Tais *walks* obtêm uma visão local do grafo subjacente ao nó de partida, e a *walk* se aproxima do comportamento da BFS no sentido de que as amostragens abrangem os nós localmente próximos. Em contraposição, se $q < 1$ a *walk* é inclinada a visitar nós mais distantes do nó t.

Tal comportamento corresponde ao da DFS que incentiva a exploração externa. Entretanto, uma diferença essencial é que a exploração ao modo de DFS é obtida usando o método de random walk. Portanto, os nós amostrados não estão necessariamente aumentados de distância do nó de início u, mas por sua vez, há o benefício de um pré-processamento computacionalmente tratável e uma amostragem com eficiência superior das *random walks*.

2.5.1.5 Redes Neurais de Grafos

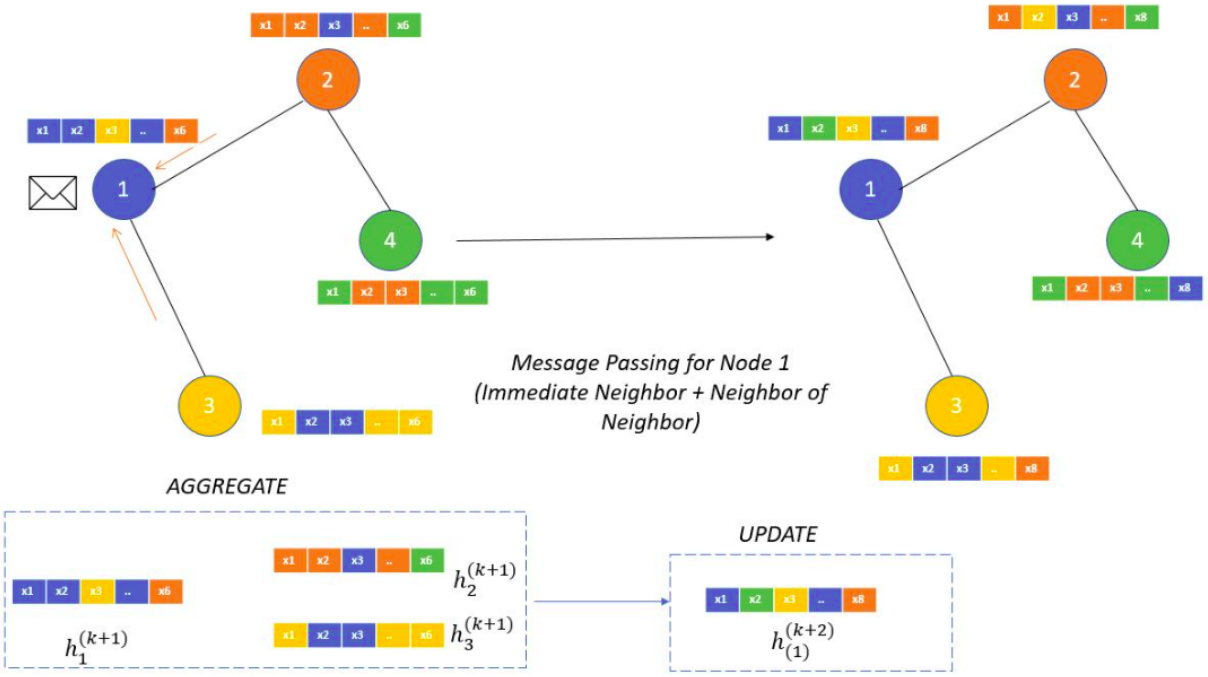
Segundo a definição de Kipf e Welling (2016), uma Rede Neural de Grafos (*Graph Neural Networks* - GNN) é um modelo para aprender um sinal em um grafo G, que recebe como entrada uma descrição de características h_i, $\forall i$ e uma descrição da estrutura do grafo em forma de matriz e produz uma saída de nível de nó Z. A equação abaixo define uma forma geral para o modelo e seus blocos de construção.

O processo para gerar Z é composto por uma série de iterações empilhando camadas e/ou épocas para calcular um *embedding* final h^Z. Na iteração $k+1$, para cada nó $v \in V$, a representação do nó é calculada por:

$$h_u^{k+1} = \sigma\left(h_u^k, \Theta\left(h_v^k, \forall \in \mathcal{N}(u)\right)\right) \tag{2.3}$$

Aqui, h_u^k é a incorporação do nó u na iteração k; θ é a função de agregação que combina informações do nó vizinho do nó u; e σ é o que é conhecido como função de atualização, responsável por transformar o resultado da função de agregação e a representação de incorporação atual do nó u em uma nova incorporação para o nó u. A ideia é ilustrada na Figura 2.6.

Figura 2.6: Blocos, *message Passing*, *Aggregationh*, *Update* para construção do GNN.



Fonte: Aritrasen.com (2022).

A Figura 2.6 ilustra o processo de passagem de mensagens em um grafo para o nó 1, enfatizando como as informações são agregadas dos nós vizinhos e subsequentemente utilizadas para atualizar o estado do nó. O diagrama superior mostra um grafo com quatro nós (1, 2, 3, 4), cada um associado a um vetor de características (x1, x2, x3, x4). As setas indicam a direção da passagem da mensagem para o nó 1 a partir de seus vizinhos imediatos (nós 2 e 3) e do vizinho (nó 4). Na parte inferior esquerda, a seção '*AGGREGATE*' detalha como as características dos nós vizinhos são agregadas, cada uma transformada por uma função h específica do seu nível de conexão com o nó 1 (k+1 para vizinhos imediatos, k+2 para o vizinho do vizinho). Na parte inferior direita, o passo '*UPDATE*' mostra o vetor de características do nó 1 atualizado para uma nova versão h(k+2), integrando as informações agregadas de seus vizinhos.

As Redes Neurais de Grafos são uma generalização da maioria das arquiteturas atuais de aprendizado profundo. Abordagens como redes convolucionais profundas (DCN)

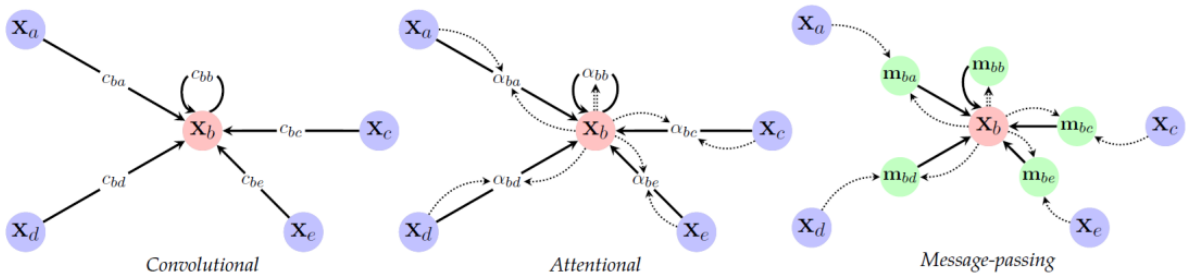
e redes neurais recorrentes (RNN) podem ser representadas como arquiteturas GNN com a adição de informações estruturais. Em particular, pode-se argumentar que a DCN é um caso particular de GNN para grafos de grade (*pixels* de imagens) e que a RNN é um caso particular de GNN para grafos de linha.

Além da generalização geométrica, a ideia central dos modelos GNN é gerar representações dependentes tanto da estrutura do grafo quanto das características do nó. Dessa forma, a combinação da posição estrutural dos nós com as características oferece uma vantagem significativa ao comparar os modelos GNN com técnicas de incorporação estrutural que produzem incorporações de baixa dimensão utilizando somente a estrutura da rede, resultando em um vetor de incorporação único para cada nó.

Segundo Bronstein *et al.* (2021), a grande maioria dos trabalhos baseados em GNN são derivados de três tipos de camadas de GNN, que são: *convolutional*, *attentional*, e de *message passing*. A diferença entre esses tipos de GNN está na estratégia usada para agregar nós vizinhos. No tipo convolutional (Kipf e Welling, 2016), a comunicação direta (*one-hop*) é agregada usando a soma normalizada dos atributos de nó dos vizinhos. No tipo *attentional* (Velickovi *et al.*, 2017), as interações são implícitas, e os nós vizinhos são agregados conforme os coeficientes de atenção aprendidos α. Finalmente, o tipo *message passing* se resume a calcular vetores arbitrários ao longo das arestas.

É importante observar que os três tipos de GNN, representados na Figura 2.7, estão relacionados de forma hierárquica, com *convolutional* $\subseteq$ *attentional* $\subseteq$ *message passing*. As GNNs de *attentional* podem representar as GNNs *convolutional* utilizando uma tabela de pesquisa com $\alpha_{u,v} = c_{u,v}$, onde α é a atenção entre os nós u e v w é o peso entre nós. Ademais, os modelos de *attentional* e *convolutional* podem ser representados como casos específicos de *message passing*, em que as mensagens correspondem simplesmente às características do nó multiplicadas por algum vetor $m_{u,v}$.

Figura 2.7: Arquiteturas *convolutional*, *attentional* e *message-passing*.



Fonte: Bronstein *et al.*, (2021).

Na Figura 2.7, observa-se que tanto as arquiteturas *convolutional* quanto as *attentional* agregam os vetores de representações dos vizinhos para gerar novas representações dos nós. A arquitetura GNN *convolutional* agrega o nó vizinho conforme os pesos de

aresta, enquanto a arquitetura *attentional* utiliza pesos de agregação aprendidos. Por outro lado, a arquitetura *message passing* agrega vetores arbitrários gerados por cada vizinho.

2.5.1.6 *Message passing*

O conceito subjacente à *message passing* do GNN é que, em cada ciclo, cada nó vai reunir informações de sua vizinhança local e produzir uma representação com dados que combinam a representação inicial do nó e informações estruturais do grafo. Após k ciclos de iterações de mensagens do GNN, as representações para cada nó contêm informações sobre as características em seu entorno até k-*hop*.

O conteúdo da mensagem é variável, mas geralmente consiste na representação do nó com alguma transformação. Os nós vizinhos mantêm essas mensagens em uma espécie de caixa de correio e as combinam ainda mais utilizando alguma função de agregação variável. É importante notar que a caixa de correio é uma abstração para armazenar temporariamente as mensagens antes da combinação e não faz parte do fundamento teórico das GNNs. Na prática, diferentes estratégias podem ser empregadas para armazenar as mensagens, dependendo da tecnologia alvo ou da aplicação.

2.5.1.7 *Aggregation*

A agregação desempenha um papel fundamental nas Redes Neurais Geométricas (GNNs). Ela consiste em combinar as informações dos nós vizinhos em uma única representação vetorial que será posteriormente incorporada à atualização do próprio nó. Esse processo é central nas GNNs e tem sido amplamente estudado na literatura, devido à sua capacidade de realizar diversas operações de convolução.

Durante a etapa de agregação, a função Θ coleta informações de um conjunto de vizinhos N e gera um único vetor H que codifica de forma eficaz todas as características da vizinhança. Essa função pode variar desde operações simples, como soma e média, até agregações mais complexas que envolvem o uso de redes neurais para combinar as entradas. É importante notar que a função Θ é, essencialmente, uma função que opera sobre um conjunto, o que significa que ela deve ser insensível à ordem das entradas. Isso é crucial no contexto das GNNs, pois não existe uma ordem natural nos vizinhos de um

nó, e a identificação dos nós pode variar dependendo da inicialização do grafo.

Embora a maioria das implementações de GNNs utilize agregações simples e estratégias para lidar com a permutação da ordem, alguns trabalhos, como os citados em Murphy *et al.* (2018) e Xu *et al.* (2018), obtiveram bons resultados aplicando uma ordem canônica aos vizinhos e utilizando funções que consideram a ordem, como as redes LSTM.

Em suma, a agregação é o componente mais crucial de uma GNN, uma vez que, nas implementações mais relevantes, os parâmetros da função de agregação são treinados, e não a representação vetorial inicial dos nós. Esse treinamento torna as GNNs mais eficientes computacionalmente e, acima de tudo, indutivas, em vez de transdutivas. A aprendizagem indutiva é vantajosa, pois permite a geração eficiente de representações para nós não vistos durante o treinamento, ao contrário dos métodos transdutivos, que requerem retrabalho do modelo para acomodar novos nós, o que pode ser inviável em escala industrial. Com a aprendizagem indutiva, uma vez que os pesos de agregação são aprendidos, não é necessário retrabalhar o modelo, a menos que ocorram mudanças nos dados.

2.5.1.8 *Update*

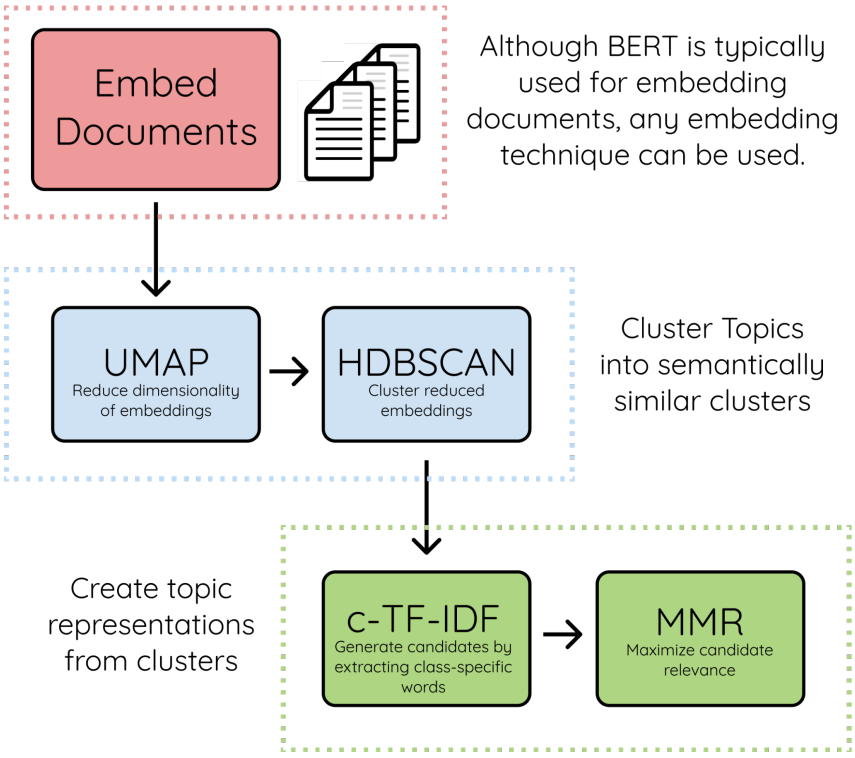
A função de atualização comumente envolve operações simples, como a soma ou a média entre a representação do nó h_u^k e as representações agregadas dos vizinhos $\Theta\left(h_u^k\right)$, seguidas por uma função não-linear σ. Um desafio recorrente ao desenvolver essas funções é chamado de "suavização excessiva", que ocorre quando, após várias iterações de GNN, as representações de todos os nós se tornam muito semelhantes. Isso acontece porque a influência dos vizinhos acaba dominando a representação inicial do nó.

Para combater a suavização excessiva e otimizar a etapa de atualização, trabalhos como Phan *et al.* (2017 e Selsam *et al.* (2018) tentam aplicar analogias aos métodos convencionais de aprendizado profundo, como conexões de salto e conexões controladas. A etapa de atualização é o ponto onde as informações agregadas dos vizinhos do nó se mesclam com a informação do próprio nó. Embora esta etapa seja opcional, é importante mencionar que é possível adicionar auto-laços no grafo e misturar a informação do nó durante a etapa de agregação, mas isso leva a resultados inferiores (Hamilton, 2021) e restringe significativamente a capacidade de priorizar a informação do nó em relação à informação dos vizinhos. A representação vetorial resultante da etapa de atualização é utilizada na próxima iteração durante a etapa de passagem de mensagens.

2.6 *BERTopic*

Segundo Grootendorst (2022), o *BERTopic* é um *framework* que incorpora algoritmos para a busca automática de tópicos densos em uma coleção de documentos, partindo do pressuposto de que documentos semanticamente similares formam tópicos (Figura 2.8).

Figura 2.8: Processo de modelagem de tópicos usando *BERTopic*. O modelo *BERT* é utilizado para gerar *embeddings*, seguido pela redução de dimensionalidade com *UMAP*. A clusterização é realizada com *HDBSCAN* e a importância das palavras em cada tópico é avaliada usando *c-TF-IDF*.



Fonte: Grootendorst (2021).

A primeira etapa do *BERTopic* converte os documentos em dados numéricos utilizando as técnicas de geração de *embeddings* do modelo *BERT*. Neste trabalho foi realizada a redução de dimensionalidade dos dados antes de realizar o agrupamento através do *UMAP*. A segunda etapa é a de clusterização, que é realizada através do *HDBSCAN*, um algoritmo de agrupamento hierárquico por densidade, proposto em Campello, Moulavi e Sander (2013).

Neste algoritmo, os documentos que possuem maior similaridade entre si são agrupados em clusters baseados na estabilidade do cluster. Uma das importantes características do *HDBSCAN* é o fato de ele não forçar a seleção de um dado para um determinado *cluster*. Caso o dado não se encaixe em nenhum grupo por similaridade, ele é considerado um *outlier*.

O modelo *CountVectorizer* foi utilizado para converter os tópicos em vetores de contagem, removendo palavras irrelevantes como *stopwords* em português.

A última etapa do processo é a seleção dos tópicos com base na importância das palavras. Para isso, o autor desenvolveu uma técnica denominada *c-TF-IDF*. Esta técnica funciona de forma parecida com o *TF-IDF* original, que compara a importância das palavras analisando todo o corpus, entretanto, o *c-TF-IDF* utiliza os clusters gerados na etapa de agrupamento aplicando o *TF-IDF* em cada um deles. Esse processo classifica as palavras conforme a importância para cada grupo gerado no processo de agrupamento e extrai os principais tópicos de cada grupo. O cálculo é feito pela fórmula:

$$\mathrm{c-TF-IDF} = \frac{t_i}{w_i} \cdot \log \frac{m}{\sum_j^n t_j} \tag{2.4}$$

Onde a frequência de cada palavra t é extraída para classe i e dividida pelo número total de palavras da classe i. Essa etapa pode ser vista como uma forma de regularização das palavras frequentes na classe. Depois, esse valor é multiplicado pelo logaritmo do número total de documentos (m) dividido pela frequência total da palavra t ao longo de todas as classes n. Assim, é obtido um valor de importância para cada palavra em um *cluster* que será utilizada para criar os tópicos.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados trabalhos recentes, publicados nos últimos 6 anos (2019–2024) referentes à área de Aprendizado de Máquina em grafos. Tais estudos foram escolhidos visando apresentar o que vem sendo publicado atualmente na literatura científica sobre a temática.

Em um estudo relacionado, Kipf e Welling (2019) investigaram a utilização de *Graph Convolutional Networks* (GCNs) para a inferência em grafos de conhecimento. A abordagem propõe a utilização de convoluções em grafos para capturar informações estruturais e semânticas dos nós e arestas, promovendo uma representação mais robusta do conhecimento contido nos grafos. Este método tem se mostrado eficaz em várias aplicações, incluindo a predição de *chatbots* e a classificação de nós em redes complexas (Kipf e Welling, 2019).

Outro trabalho relevante é o de Ma *et al.* (2022), que propuseram a técnica *Node2Vec* para a extração de características de grafos. Esta técnica foi desenvolvida para capturar as relações estruturais e temporais em grafos de conhecimento, permitindo a aplicação em diferentes domínios, como redes sociais e biologia computacional. A técnica demonstrou resultados promissores ao melhorar a precisão das previsões em diversas tarefas de Aprendizado de Máquina aplicadas a grafos (Ma *et al.*, 2022).

O trabalho de Yang *et al.* (2023) introduziu o uso de modelos de Aprendizado Profundo para a segmentação de páginas de documentos digitalizados. Utilizando uma combinação de técnicas de Aprendizado de Máquina e grafos, o estudo aborda o problema da segmentação de documentos, identificando componentes como blocos de texto, figuras e tabelas. Os resultados experimentais, obtidos com imagens de documentos de bancos de dados como o PRIMA *Layout Analysis Datase*, demonstraram o potencial da abordagem proposta em lidar com leiautes diversificados e complexos, além de destacar a vantagem da análise lógica de leiaute na extração de informações de documentos digitalizados (Yang *et al.*, 2023).

O estudo realizado por Zhang *et al.* (2024), intitulado "*Hierarchical Temporal Graph Attention Networks for Popularity Prediction in Information Cascades*", explora a predição da popularidade de cascatas de informação utilizando redes neurais baseadas em grafos temporais hierárquicos. O trabalho aborda a modelagem dinâmica de grafos

inteiros de cascata para capturar tendências de popularidade implícitas em cascatas complexas. Por meio da incorporação de *embeddings* de nós sensíveis ao tempo, mecanismos de atenção em grafos e estruturas de *pooling* hierárquicas, o modelo *HierCas* demonstra desempenho superior em comparação com as abordagens mais avançadas da área, conforme comprovado por experimentos em dois conjuntos de dados do mundo real, disponíveis no repositório *GitHub* (Zhang *et al.*, 2024).

Esses estudos evidenciam a evolução das técnicas de Aprendizado de Máquina em grafos e suas diversas aplicações, desde a predição de popularidade em cascatas de informação até a segmentação de documentos e a inferência em grafos de conhecimento. A contínua pesquisa e desenvolvimento nessas áreas prometem avançar ainda mais a capacidade de análise e processamento de grandes volumes de dados estruturados em grafos.

Capítulo 4

Metodologia

Nesse capítulo, está apresentada a metodologia utilizada para desenvolver esta pesquisa de Mestrado e as etapas desenvolvidas, que possibilitaram o vislumbre dos resultados alcançados.

4.1 Delineamento da pesquisa

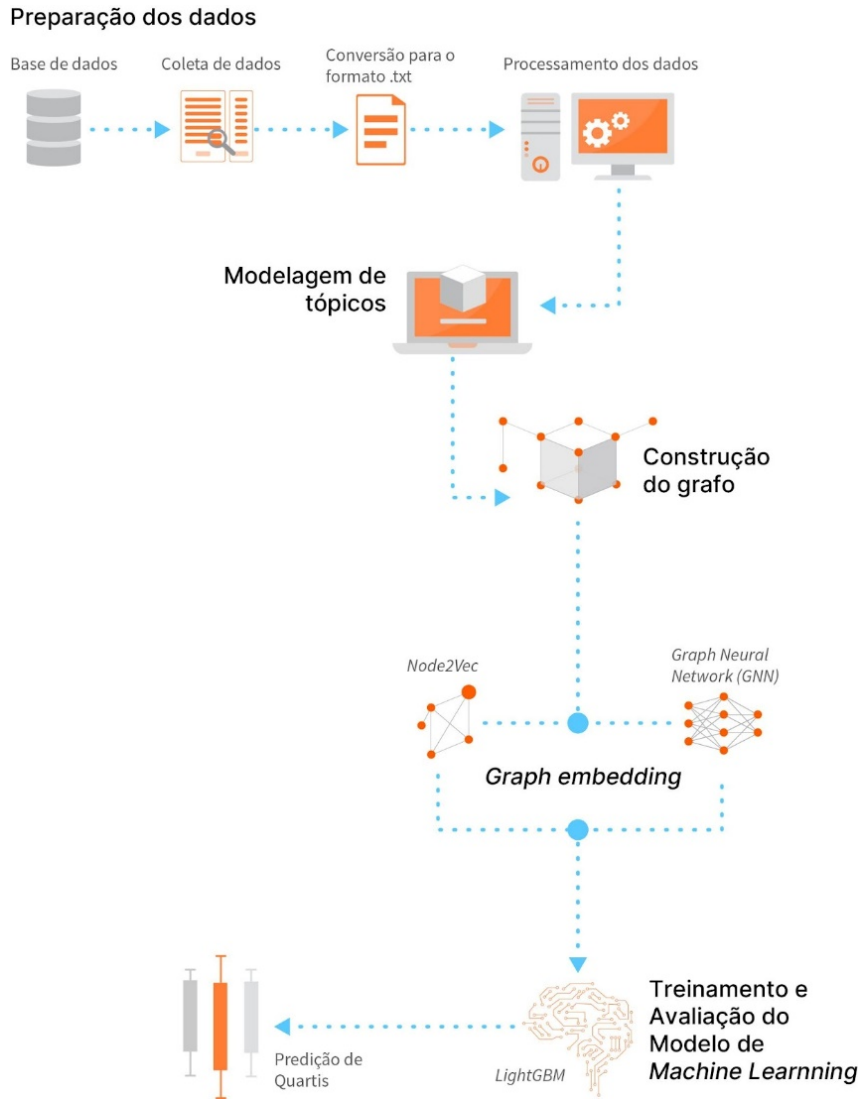
A pesquisa realizada é de natureza quantitativa e descritiva, uma vez que utiliza dados numéricos e técnicas computacionais para análise e predição, bem como busca descrever e analisar a popularidade dos atos normativos utilizando técnicas avançadas de processamento de dados. A abordagem quantitativa é evidenciada pelo uso de algoritmos de Aprendizado de Máquina e análise de grafos para tratar e prever a popularidade dos atos normativos, enquanto a natureza descritiva visa fornecer uma compreensão detalhada das características desses atos e como elas podem influenciar sua popularidade.

Além disso, a pesquisa se caracteriza como documental e bibliográfica. É documental porque se baseia na coleta e análise de atos normativos disponíveis em fontes oficiais. Ao mesmo tempo, é bibliográfica porque envolve a revisão e utilização de literatura acadêmica existente sobre aprendizado de máquina, grafos, Processamento de Linguagem Natural e modelagem de tópicos.

O desenvolvimento da pesquisa demandou a implementação de um fluxo de processamento (*pipeline*) estruturado e abrangente, a fim de realizar análises detalhadas e produzir resultados relevantes. O *pipeline* foi projetado para englobar uma série de etapas cruciais, visando a transformação e análise dos dados textuais coletados. Cada fase do *pipeline* contribuiu de maneira essencial para a obtenção de informações valiosas e aprimoramento das análises realizadas. O fluxo de processamento foi composto pelas seguintes fases apresentadas na Figura 4.1.

A descrição de cada passo da metodologia é descrita a seguir:

Figura 4.1: Diagrama do *pipeline* do projeto, incluindo: coleta de dados, conversão de formato, pré-processamento, modelagem de tópicos, construção do grafo, geração de *embeddings* e treinamento do modelo.



Fonte: Elaborado pelo autor.

4.2 Coleta de dados

A segurança alimentar é um elemento crucial para a saúde pública e o bem-estar social, demandando uma análise rigorosa das regulamentações que a envolvem. Neste estudo, o foco recaiu sobre a análise de atos normativos brasileiros específicos da área de segurança alimentar, utilizando um *corpus* singular e altamente relevante. Os dados, também chamado *corpus*, que formam a base deste estudo, foram coletados diretamente do site oficial do Ministério da Agricultura e Pecuária (MAPA), acessível por meio do endereço eletrônico https://www.gov.br/agricultura/pt-br. Essa fonte foi selecionada pela sua autoridade em informações do setor, e a coleta ocorreu em 3 de setembro de

2022. Foram capturados diversos tipos de normas jurídicas, como Lei, Decreto-Lei, Portaria, Instrução Normativa, Decreto Legislativo, Medida Provisória, Decreto, Ato, Diário Oficial, Memorando Circular, Decisão, Norma Operacional, Protocolo, Resolução MERCOSUL, e Ofício-Circular, além de uma cartilha contendo orientações sobre a inscrição de espécies no RNC, totalizando 320 atos normativos, todos em formato *Portable Document Format* (PDF). A coleta foi realizada utilizando a técnica de *web scraping*, permitindo a extração automatizada de informações do site do MAPA.

4.3 Conversão dos Dados

Após a conclusão da fase de coleta dos dados, a etapa subsequente foi dedicada à conversão dos dados adquiridos para o formato *text file format* (TXT). Essa etapa desempenhou um papel fundamental na preparação dos dados brutos, para facilitar manipulações e análises posteriores. O processo de conversão foi concretizado por meio da elaboração de um *script* desenvolvido utilizando a linguagem de programação *Python*.

A seleção da linguagem *Python* para desenvolver esta pesquisa foi baseada em sua versatilidade e disponibilidade de bibliotecas voltadas para manipulação e transformação de dados.

4.4 Tratamento dos Dados

Em NLP, a etapa de pré-processamento tem a finalidade de criar um conjunto de *tokens* que terão alguma relevância para a modelagem de tópicos e é dividida em algumas tarefas sequenciais, conforme apresentado nos próximos itens.

- **Tokenização:** transformação de caracteres para minúsculos, troca de caracteres acentuados pelo mesmo caractere sem a acentuação e retirada de caracteres que não sejam letras, além da exclusão dos números.
- **Remoção de *stopwords*:** remoção de palavras irrelevantes para a classificação, como as preposições, por exemplo; remoção de nomes próprios e remoção de palavras com menos de 3 caracteres.

- **Stemmização:** redução das palavras ao seu radical, mantendo apenas a parte mais significativa de cada palavra.
- **Remoção de verbos:** a remoção de verbos durante a modelagem de tópicos é fundamental para melhorar a precisão, reduzir o ruído e simplificar o processamento, o que resulta em uma análise mais eficaz e eficiente dos dados de texto.

A etapa de pré-processamento proporcionou um melhor entendimento da base de dados, apresentando as seguintes quantidades de *tokens* por pronunciamento em cada uma das etapas e as seguintes diferenças entre cada classificação na etapa de aplicação de *stemming*.

4.5 Modelagem de Tópicos

Para realizar a tarefa de agrupamento e sumarização de tópicos em um conjunto de 320 documentos previamente normalizados, utilizamos o seguinte procedimento: todos os textos dos atos normativos foram convertidos em uma lista única encadeada e passados como parâmetro para o *BERTopic*. Em seguida, todas as etapas mencionadas anteriormente (1.6) foram realizadas e um modelo foi gerado como resultado. Através desse modelo, pudemos extrair os principais tópicos, analisando a frequência e o valor de importância obtido anteriormente, e gerar um mapa de distribuição de tópicos.

Criou-se uma instância do *BERTopic* em que foram definidos vários parâmetros. O parâmetro '*language*' foi configurado como '*portuguese*', indicando que o algoritmo usará modelos de processamento de linguagem natural pré-treinados em português. Além disso, o parâmetro '*min_topic_size*' foi definido como 320, garantindo que cada tópico resultante contenha pelo menos 320 documentos, o que ajuda a evitar a formação de tópicos com um número muito pequeno de documentos. Esta escolha foi feita empiricamente para assegurar que os tópicos formados sejam significativos e representativos.

O parâmetro '*n_gram_range*' foi configurado como (1, 3), o que significa que o algoritmo considerará sequências de palavras de 1 a 3 palavras para construir os tópicos, levando em consideração não apenas palavras individuais, mas também combinações de até três palavras em sequência.

O parâmetro '*embedding_model*' refere-se ao modelo de incorporação usado para representar os documentos como vetores numéricos, permitindo uma análise mais eficaz da similaridade entre os documentos. O '*umap_model*' é responsável por reduzir a dimensionalidade dos vetores de incorporação, facilitando a interpretação dos *clusters* resultantes. O '*vectorizer_model*' converte os documentos em representações numéricas, possibilitando

a aplicação de cálculos matemáticos sobre os dados textuais. O '*ctfidf_model*' é utilizado para calcular a importância das palavras nos documentos, atribuindo pesos com base na frequência das palavras nos documentos e no *corpus* geral.

Com o parâmetro '*calculate_probabilities*' ativado, o algoritmo calcula as probabilidades de cada documento pertencer a um determinado tópico durante o processo de modelagem. A opção '*verbose*' exibirá informações detalhadas sobre cada etapa do processo durante a execução do algoritmo, úteis para depuração e análise de desempenho. O '*hdbscan_model*' é um algoritmo de clusterização hierárquica, baseado em densidade, capaz de identificar clusters de formatos irregulares e de diferentes tamanhos. O parâmetro '*representation_model*' define como os documentos serão representados como vetores numéricos, facilitando a manipulação eficaz durante o processo de modelagem.

O '*hdbscan_model*' é um algoritmo de clusterização hierárquica, baseado em densidade, capaz de identificar clusters de formatos irregulares e de diferentes tamanhos. O parâmetro '*representation_model*' define como os documentos serão representados como vetores numéricos, facilitando a manipulação eficaz durante o processo de modelagem.

Após estabelecer os argumentos, a função *fit_transform* é convocada e o algoritmo é acionado. Durante este estudo, todo o procedimento foi realizado com base nos parâmetros personalizados do *BERTopic*, destacando-se a habilidade de ajustar o algoritmo conforme as exigências e particularidades específicas do conjunto de dados e do problema em questão.

4.6 Matriz de similaridade

A matriz de similaridade foi construída a partir da matriz de Documentos por Tópicos gerada com dimensões de "Número de Documentos"por "Número de Tópicos", após a análise e extração dos tópicos. Essa matriz proporciona uma visão abrangente sobre quais tópicos estão presentes em cada documento e com qual intensidade, contribuindo para uma compreensão mais completa dos elementos temáticos presentes nos documentos.

A matriz de similaridade foi gerada para quantificar a relação de similaridade entre diferentes pares de documentos por meio da comparação de seus vetores de tópicos. Para isso, foi utilizada a medida de similaridade do cosseno, que calcula o ângulo entre os vetores de tópicos dos documentos. Essa medida é valiosa porque fornece uma maneira precisa de entender como os documentos estão relacionados uns com os outros, identificando quais possuem conteúdo semelhante ou abordam tópicos parecidos.

A importância da matriz de similaridade reside na capacidade de oferecer insights valiosos sobre um grande conjunto de documentos. Ao calcular e representar numeri-

camente as semelhanças entre os documentos, essa matriz possibilita a identificação de padrões, grupos temáticos e relações que poderiam não ser imediatamente perceptíveis através de análises manuais.

4.7 Grafo e suas característica

Orientado pela matriz de similaridade previamente gerada, foi construído um grafo acíclico e direcionado para quantificar as relações de similaridade entre diferentes pares de documentos. Essa construção contou com o *NetworkX*, um pacote *Python* utilizado para manipulação de grafos e redes complexas e se baseia na comparação dos vetores de tópicos associados a cada documento. No âmbito desse processo, cada nó no grafo está vinculado a um documento específico, e as arestas que interligam esses nós refletem as relações de similaridade detectadas na matriz. A direção das arestas é definida pela informação temporal, indicando a ordem dos anos de publicação dos documentos. A resultante representação gráfica oferece uma visão nítida da estrutura subjacente e das interconexões entre os documentos.

Além da criação inicial do grafo, a análise se estende ao cálculo e exploração de diversas métricas do grafo. Essa abordagem aprofundada proporciona *insights* mais detalhados sobre a importância relativa de cada documento dentro da rede.

4.8 Geração de *embeddings* do grafo

Para a geração de *embeddings* do grafo, utilizamos duas abordagens principais: *Node2Vec* e GNN (*Graph Neural Networks*). A escolha destas técnicas se baseou na capacidade comprovada de cada uma em capturar informações estruturais importantes do grafo, e preservar a estrutura local e global do grafo e representá-las de maneira eficaz em um espaço de dimensões reduzidas.

O algoritmo *Node2Vec* foi selecionado devido à sua habilidade em preservar a estrutura e os padrões de conexões entre nós. A flexibilidade do *Node2Vec* em ajustar a exploração e a explotação durante a geração dos caminhos no grafo nos permitiu capturar informações contextuais importantes. Foram realizados experimentos com diferentes tamanhos de *embeddings* (8, 64 e 128 dimensões) para avaliar a qualidade das

representações aprendidas e compreender a captura de padrões estruturais complexos em diferentes espaços dimensionais.

Por outro lado, as *Graph Neural Networks* (GNNs) foram adotadas devido à sua capacidade de aprender representações considerando as interações locais e globais do grafo. A utilização de redes neurais possibilitou a captura de informações complexas e não lineares presentes nas relações entre nós, resultando em representações mais ricas e informativas. Da mesma forma que com o *Node2Vec*, conduzimos experimentos com diferentes tamanhos de *embeddings* (8, 64 e 128 dimensões) para compreender o desempenho da técnica em diferentes espaços dimensionais e comparar as representações aprendidas com aquelas geradas pelo *Node2Vec* em termos de complexidade e expressividade.

Ao adotar essas duas abordagens complementares, buscamos explorar a complementaridade entre a capacidade do *Node2Vec* em capturar informações estruturais de grafos e a habilidade das GNNs em aprender representações complexas, visando a geração de *embeddings* de alta qualidade que possam ser aplicados a uma variedade de tarefas de análise e processamento de grafos.

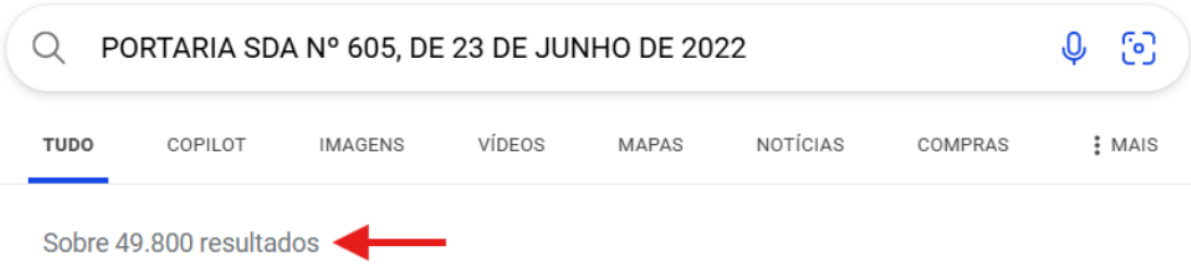
4.9 Tratamento do modelo de *Machine Learning*

Um dos objetivos deste estudo foi treinar um modelo de regressão baseado no algoritmo *LightGBM* para prever quartis e, dessa forma, identificar a posição relativa dos atos normativos dentro de uma distribuição de popularidade. Para isso, realizamos dois conjuntos de experimentos: o primeiro com diferentes dimensões de *embeddings* gerados pelo *Node2Vec* e GNN, e o segundo com características do grafo, incluindo '*degree centrality*', '*closeness centrality*', '*load centrality*', '*harmonic centrality*', '*betweenness centrality*', '*average neighbor degree*', '*clustering*' e '*pagerank*'.

Os valores alvo (y) foram obtidos a partir de pesquisas no Google, ocorrida no dia 23 de agosto de 2023, sobre atos normativos coletados (Figura 4.2). Para realização da pesquisa, foi adotado e utilizado exatamente o título do ato normativo contido dentro do documento. Isso garantiu precisão e relevância nos resultados, permitindo uma medida exata do interesse público. A padronização do método assegurou comparabilidade e consistência nos dados, tornando o processo eficiente e replicável, além de reduzir ruídos e resultados irrelevantes.

Para o processamento, a data da pesquisa no Google foi subtraída da data de criação do ato normativo e, em seguida, dividida pelo número total de pesquisas no Google no mesmo intervalo de anos. Esse procedimento proporcionou uma métrica representativa da frequência com que um ato normativo foi pesquisado em relação ao seu ano de criação.

Figura 4.2: Exemplo da pesquisa no Google para o título "PORTARIA SDA Nº 605, DE 23 DE JUNHO DE 2022"mostrando aproximadamente 49.800 resultados.



Fonte: Elaborado pelo autor.

4.9.1 Experimento 1: utilização de *embeddings*

Neste experimento, conduziu-se uma análise detalhada do treinamento de um modelo de regressão com o uso do *framework LightGBM*, a fim de prever os quartis de atos normativos, baseando-nos em *embeddings* gerados a partir de duas técnicas: *Node2Vec* e GNN (*Graph Neural Networks*). A utilização de *embeddings* visa fornecer representações vetoriais das características dos atos normativos, permitindo ao modelo capturar relações e padrões complexos. Para avaliar o desempenho do modelo em diferentes cenários, foi realizado uma série de experimentos considerando diversas dimensões de *embeddings* e operações de concatenação e subtração entre eles.

Os dados de entrada (X) consistem em *embeddings* gerados por *Node2Vec* e GNN, variando em dimensões de 8, 64 e 128. Adicionalmente, ocorreu a concatenação e subtração de *embeddings* com a mesma dimensão. Os valores alvo (y) foram obtidos a partir de pesquisas no Google sobre atos normativos coletados, conforme descrito no ítem 4.8.

Para treinar o modelo de regressão, inicialmente calcula-se um valor de corte para cada quartil com base na frequência de aparecimento dos atos normativos. Em seguida, divide-se os dados em conjuntos de treinamento e teste. Utilizamos validação cruzada estratificada com 5 dobras para avaliar o desempenho do modelo e evitar qualquer viés na divisão dos dados. O *LightGBM* é um algoritmo de *gradient boosting* que permite otimização de hiperparâmetros. Para encontrar os melhores hiperparâmetros que minimizem o erro absoluto médio (MAE), utilizamosou-se a biblioteca *Optuna*.

Após a realização dos experimentos, identificamos os melhores conjuntos de hiperparâmetros para cada cenário e calculamos o MAE médio no conjunto de validação cruzada. Também calculamos o MAE nos dados de teste para avaliar o desempenho do modelo em um conjunto independente. Os resultados são consistentemente avaliados e comparados entre os diferentes cenários de *embeddings*.

4.9.2 Experimento 2: utilização de *features* do grafo

No segundo experimento, o treinamento do modelo de regressão *LightGBM* foi conduzido visando prever os quartis com base em diversas métricas derivadas do grafo. Essas metricas incluem '*degree centrality*', '*closeness centrality*', '*load centrality*', '*harmonic centrality*', '*betweenness centrality*', '*average neighbor degree*', '*clustering*' e '*pagerank*'. Essas métricas forneceram as *features* de entrada (X) para o modelo, enquanto os resultados das pesquisas no Google serviram como o valor a ser previsto (y).

Para preparar o conjunto de dados de treinamento, os valores de y passaram por um cálculo específico. A diferença entre a data da pesquisa no Google e a data de criação do ato normativo foi estimada, seguida pela divisão entre o número total de pesquisas no Google e a diferença entre os anos.

O treinamento do modelo de regressão *LightGBM* foi conduzido para minimizar o erro absoluto médio (MAE). Durante o processo de treinamento, testamos diferentes conjuntos de hiperparâmetros para determinar os melhores valores. Esse procedimento foi executado para cada dataframe específico, utilizando uma validação cruzada estratificada para garantir a robustez dos resultados. O melhor conjunto de hiperparâmetros foi identificado para cada iteração do processo de treinamento, e o modelo final foi treinado com esses parâmetros otimizados. A avaliação do desempenho do modelo foi realizada usando o MAE nos dados de teste.

Adicionalmente, foram gerados gráficos de boxplot para visualizar a distribuição dos valores previstos em relação aos quartis.

4.9.3 *SHAP*

O método *SHAP (SHapley Additive exPlanations)* é uma técnica avançada de explicabilidade que ajuda a entender como as *features* individuais de um modelo contribuem para as predições. Neste trabalho, foi utilizado o cálculo do *SHAP* para um modelo de regressão baseado no algoritmo *LightGBM*, associado a um conjunto de *features* extraídas de um grafo. As *features* utilizadas foram '*degree centrality*', '*closeness centrality*', '*load centrality*', '*harmonic centrality*', '*betweenness centrality*', '*average neighbor degree*', '*clustering*' e '*pagerank*'.

Inicialmente divide-se os dados em conjuntos de treinamento e teste e em seguida realiza-se os ajustes de hiperparâmetros na biblioteca *Optuna* para otimizar o modelo *LightGBM*, minimizando a métrica de erro absoluto médio (MAE). A validação cruzada

estratificada com 5 dobras foi empregada para garantir uma avaliação robusta do desempenho do modelo.

O processo de cálculo do *SHAP* foi realizado após o treinamento do modelo final com os melhores hiperparâmetros. Utilizamos o *TreeExplainer* da biblioteca *SHAP* para calcular os valores *SHAP* para os dados de teste. Em seguida, um resumo dos valores *SHAP* foi gerado e um gráfico *SHAP* foi criado para visualizar a importância das diferentes *features* na tomada de decisão do modelo.

Capítulo 5

Resultados e discussão

O presente trabalho visa a predição da popularidade de busca no Google de atos normativos brasileiros disponibilizados pelo MAPA, mais especificamente sobre segurança alimentar, através da utilização de técnicas de Aprendizado de Máquina em grafos. A coleta de dados para a construção do corpus iniciou-se em setembro de 2022 e conta com 320 normas jurídicas.

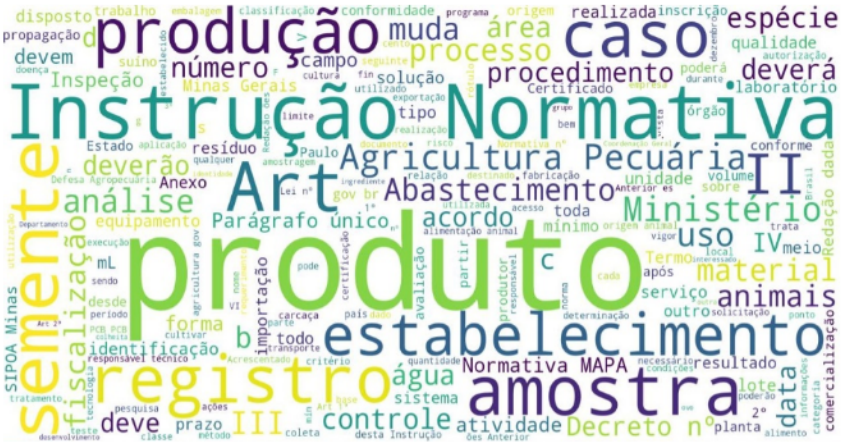
A análise dos atos normativos sobre segurança alimentar, visualizada através de uma nuvem de tags (Figura 5.1), destaca termos como "produto", "estabelecimento", "registro", "instrução normativa", "produção"e "amostra". A centralidade desses termos revela as principais preocupações das regulamentações, que buscam garantir a qualidade e segurança alimentar desde a produção até a comercialização. A presença de palavras como "produto"e "estabelecimento"sublinha a importância dos itens alimentícios e dos locais de produção, enquanto "registro"e "instrução normativa"enfatizam a necessidade de documentação e conformidade com diretrizes detalhadas.

Além disso, termos como "análise", "controle", "material"e "uso"refletem a abrangência das normas, cobrindo desde a análise de resíduos e controle de processos até a especificação de materiais permitidos. A coleta e análise de amostras são práticas fundamentais mencionadas, garantindo que os produtos atendam aos padrões de segurança e qualidade. Em suma, a nuvem de *tags* evidencia um esforço contínuo para assegurar que os alimentos consumidos pela população sejam seguros e de alta qualidade, através de regulamentações rigorosas e abrangentes.

Após a etapa de preparação do *corpus*, implementamos o modelo *BERTopic* com configurações ajustadas conforme descrito em 3,4, o que resultou na identificação de 49 tópicos distintos, com representação gráfica em barras de termos-chave para alguns desses tópicos. Esses gráficos são construídos com base nas pontuações *c-TF-IDF* associadas a cada tópico, permitindo uma análise comparativa entre os diferentes tópicos. Tal abordagem proporciona *insights* valiosos e facilita a visualização hierárquica resultante dessa análise (Figura 5.2).

Os tópicos gerados podem ser observados e organizados hierarquicamente. A fim de compreender a estrutura hierárquica potencial dos tópicos, associa-se a biblioteca *scipy.cluster.hierarchy* para formar agrupamentos e analisar suas inter-relações. Essa

Figura 5.1: Nuvem de tags representando os termos mais frequentes nos atos normativos sobre segurança alimentar. Os termos destacados, como "produto", "estabelecimento", "registro"e "instrução normativa", refletem as principais áreas de foco das regulamentações.



Fonte: Elaborado pelo autor.

abordagem facilita a seleção de um número adequado de tópicos, contribuindo para a redução da complexidade do conjunto inicial. A visualização da clusterização hierárquica é apresentada na Figura 5.3.

A análise de tópicos realizada forneceu uma forma estruturada de interpretar o conjunto de atos normativos para construção do grafo acíclico direcionado (DAG), visando explorar as relações temporais e temáticas entre documentos. Após a modelagem de tópicos, utilizou-se a similaridade de cosseno para estabelecer uma matriz de similaridade entre os pares de documentos e o limiar de 0,82 foi definido para discernir as conexões significativas e identificar quais pares de documentos compartilhavam tópicos similares. Três tópicos principais foram identificados e utilizados para a construção do grafo.

Com a matriz de similaridade estabelecida e o limiar aplicado, obteve-se pares de documentos com tópicos relevantes em comum, permitindo construir um grafo que considerasse a direcionalidade baseada no ano do documento. Este passo foi crucial para garantir que a direção das arestas refletisse a sequência temporal, apontando do documento mais antigo para o mais recente.

Para a construção do grafo, foi utilizada a biblioteca *NetworkX* em conjunto com a capacidade de visualização do *Pygraphviz*. Na Figura 5.4, é apresentado somente uma parte do grafo, com o propósito de visualizar a sua estrutura. No entanto, o grafo é consideravelmente maior e não é viável anexá-lo na íntegra ao trabalho.

Na Figura 5.4, o gráfico direcionado e acíclico visualiza a inter-relação entre diferentes atos normativos, com os vértices representando o índice de cada ato. As arestas, por sua vez, simbolizam os três tópicos mais relevantes que estabelecem a similaridade entre esses atos.

O grafo resultante foi estruturado para refletir não apenas a relação entre docu-

Figura 5.2: Gráficos de barras mostrando os termos-chave de tópicos gerados pelo *BERTopic*, com base nas pontuações *c-TF-IDF*.

Fonte: Elaborado pelo autor.

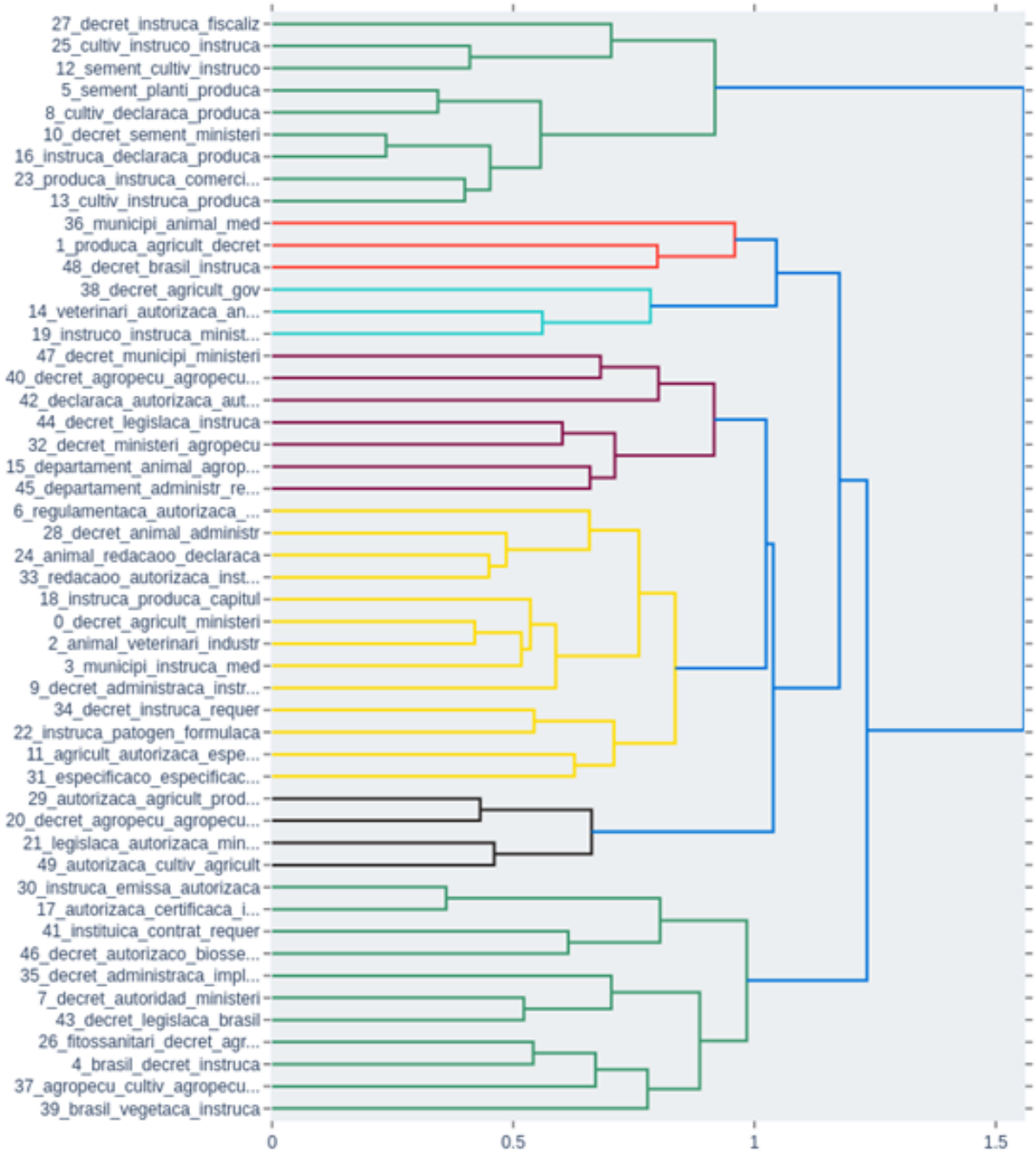
mentos, mas também para destacar os tópicos que conectam cada par de documentos. A análise das arestas e dos nós, indicou uma interconexão temática alinhada com a evolução temporal dos documentos, conforme demonstrado pelos rótulos dos tópicos. A metodologia utilizada proporcionou uma visualização clara das ligações temporais e conceituais entre os documentos, auxiliando na compreensão da progressão e influência dos tópicos ao longo do tempo.

A utilização do DAG para representar relações documentais ressalta a relevância da orientação temporal na análise de documentos. Os tópicos mais influentes podem ser rastreados em uma linha do tempo, fornecendo insights sobre como certas temáticas ganham ou perdem relevância. Embora a seleção do limiar de similaridade seja um aspecto subjetivo, o valor escolhido (0,82) provou ser eficaz na filtragem de conexões menos significativas, mantendo um grau de rigor na análise de tópicos.

Os resultados obtidos reforçam a utilidade do *BERTopic* como uma ferramenta de modelagem de tópicos robusta e do *NetworkX* como um meio de visualizar e analisar relações complexas em conjuntos de dados documentais. Essas ferramentas juntas fornecem uma poderosa capacidade de análise para pesquisadores e profissionais interessados em mineração de texto e análise de dados.

Buscou-se explorar a eficácia do *Node2Vec* e das *Graph Neural Networks* (GNNs)

Figura 5.3: Dendrograma ilustrando a clusterização hierárquica dos tópicos gerados pelo *BERTopic*. Os tópicos são agrupados com base na similaridade, facilitando a compreensão das relações entre eles.

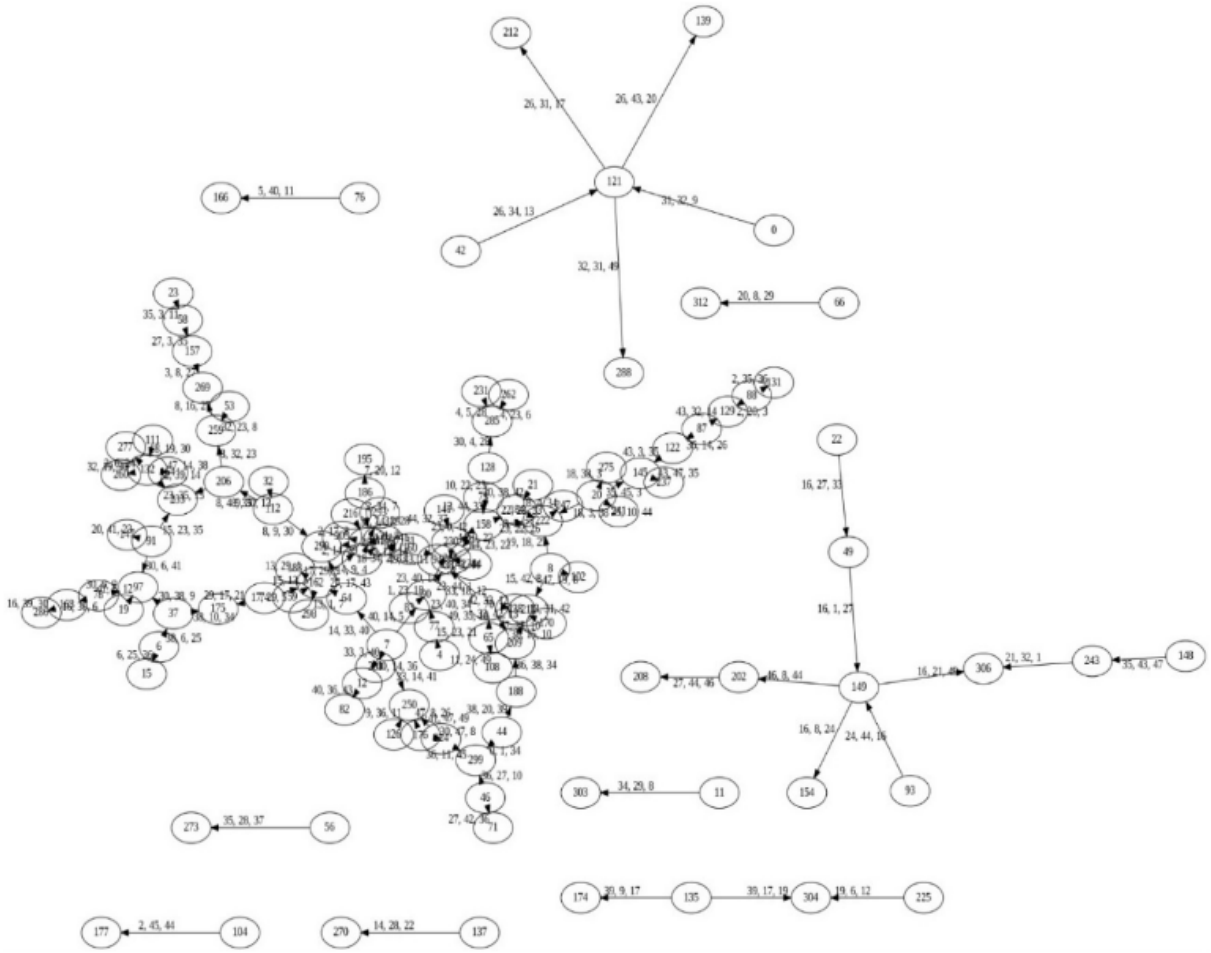


Fonte: Elaborado pelo autor.

na geração de representações vetoriais, ou *embeddings*, para nós de um grafo. Focando a análise na variação das dimensões dos *embeddings* gerados (8, 64 e 128) e em experimentos de concatenação e subtração de *embeddings* com a mesma dimensão.

A análise dos *boxplots* para os valores previstos utilizando *embeddings* de 8 dimensões para *Node2Vec* e *Graph Neural Networks* revela várias características importantes. Primeiramente, ao considerar a mediana (Q2), que é a linha central nos *boxplots*,

Figura 5.4: Parte do grafo acíclico direcionado mostrando a inter-relação entre atos normativos. Vértices representam atos normativos e arestas indicam tópicos relevantes que conectam esses atos.



Fonte: Elaborado pelo autor.

percebe-se que, para *Node2Vec*, os valores médios são consistentes entre os quartis, sugerindo uma distribuição relativamente uniforme dos valores centrais. Por outro lado, o e *Graph Neural Networks* apresenta uma mediana levemente mais alta no Q3, indicando uma leve assimetria para valores maiores neste quartil.

Em relação ao primeiro quartil (Q1), a borda inferior da caixa nos mostra que os valores iniciais dos dados são similares entre *Node2Vec* e *Graph Neural Networks*, com uma pequena variação, podendo indicar uma consistência nos valores mais baixos entre os métodos.

O terceiro quartil (Q3), que é a borda superior da caixa, sugere que o e *Graph Neural Networks* tem uma maior dispersão dos 25% dos dados superiores em comparação ao *Node2Vec*, principalmente no Q3, onde o *Graph Neural Networks* apresenta valores superiores.

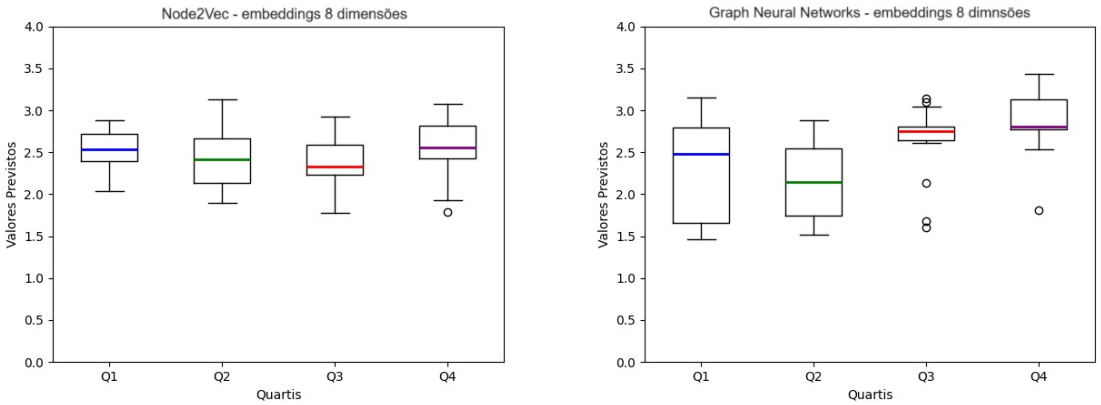
O intervalo interquartil (IQR), que mede a dispersão da metade central dos dados, parece ser mais amplo no *Graph Neural Networks*, especialmente no Q3 e Q4, indicando

uma variabilidade maior nestes quartis em comparação ao *Node2Vec*.

Os valores mínimo e máximo dentro dos limites aceitáveis mostram que o *Graph Neural Networks* possui um alcance maior de valores, como visto pelas linhas que se estendem para fora da caixa, especialmente no Q3 e Q4. Isso pode ser um indicativo de que o gnn8 captura extremos mais significativos do que o *Node2Vec*. Os *outliers*, marcados como pontos individuais fora das linhas de máximo e mínimo, são mais prevalentes no *Graph Neural Networks*, o que pode ser um indicativo de que este modelo pode estar mais sujeito a valores atípicos ou a uma maior variância nos dados preditos.

Por fim, a assimetria pode ser observada pela posição da mediana dentro da caixa e o comprimento das caudas. O *Graph Neural Networks* mostra uma assimetria positiva no Q3, sugerindo uma distribuição com uma cauda mais longa para valores mais altos. A dispersão, medida pela largura do IQR, é claramente maior no *Graph Neural Networks*, indicando maior variabilidade nos valores previstos (Figura 5.5).

Figura 5.5: *Boxplots* comparando os quartis preditos utilizando *embeddings* de 8 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos.



Fonte: Elaborado pelo autor.

Ao comparar os quartis preditos utilizando *embeddings* de 64 dimensões para *Node2Vec* e *Graph Neural Networks*, observamos diferenças notáveis e alguns padrões similares entre as duas abordagens. Para *Node2Vec*, a mediana (Q2) do primeiro e terceiro quartis (Q1 e Q3) parece ser ligeiramente mais alta do que para *Graph Neural Networks*, indicando que os valores médios previstos pelo *Node2Vec* são geralmente maiores. Isso pode sugerir que o modelo *Node2Vec* tende a prever valores centrais um pouco mais elevados, ou que a sua capacidade de captar a centralidade dos dados é diferente daquela do *Graph Neural Networks*.

O primeiro quartil (Q1) para ambos *Node2Vec* e *Graph Neural Networks* mostra uma consistência, com o *Node2Vec* apresentando uma borda inferior da caixa ligeiramente mais alta, o que implica que 25% dos dados mais baixos estão previstos com valores um pouco mais elevados do que o *Graph Neural Networks*.

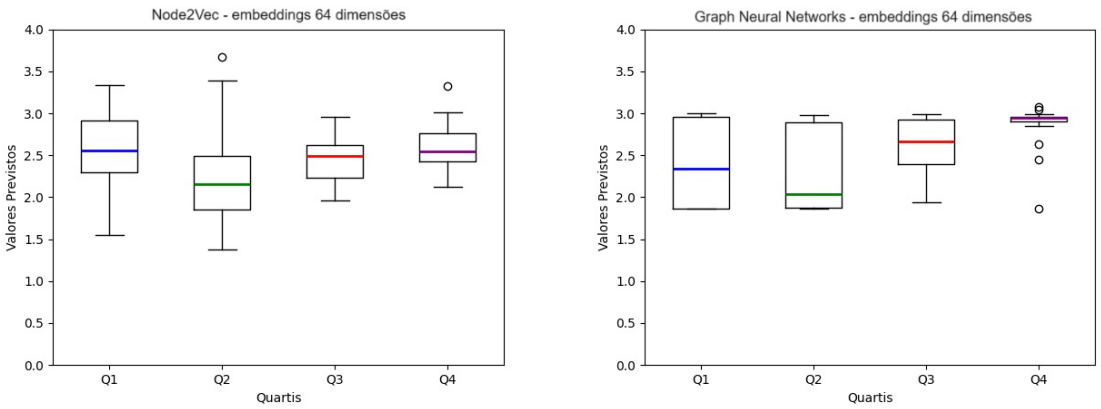
Em relação ao terceiro quartil (Q3), *Node2Vec* apresenta valores previstos mais altos para 75% dos dados, indicado pela borda superior da caixa. Isto sugere que *Node2Vec* tende a prever valores mais elevados para a maioria dos dados quando comparado com *Graph Neural Networks*.

O intervalo interquartil (IQR) para *Node2Vec* é ligeiramente mais estreito do que para *Graph Neural Networks*, especialmente nos quartis Q1 e Q3, sugerindo que *Node2Vec* tem uma dispersão menor na metade central dos dados, o que pode indicar uma consistência maior nas previsões.

Os valores mínimo e máximo (dentro de limites aceitáveis) parecem ser similares em ambos os modelos, com *Graph Neural Networks* exibindo uma ligeira tendência para valores mínimos maiores. Os *outliers*, representados por pontos individuais fora das linhas de máximo e mínimo, são visivelmente presentes em ambos os casos, mas com uma frequência ligeiramente superior no *Node2Vec*.

No que diz respeito à assimetria, ambos os modelos exibem algum grau de assimetria, como indicado pela posição da mediana dentro da caixa e pelo comprimento das caudas. A assimetria parece ser um pouco mais acentuada no *Node2Vec*, especialmente no quartil Q4. A dispersão, avaliada pela largura do IQR, como já mencionado, é menor para *Node2Vec*, indicando que as previsões deste modelo são menos variáveis (Figura 5.6).

Figura 5.6: *Boxplots* comparando os quartis preditos utilizando *embeddings* de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos.



Fonte: Elaborado pelo autor.

Ao analisar os *boxplots* dos valores previstos utilizando *embeddings* de 128 dimensões para os modelos *Node2Vec* e *Graph Neural Networks*, observamos diferenças notáveis na distribuição dos quartis, o que pode indicar diferenças na capacidade de previsão de cada modelo.

Para o modelo *Node2Vec*, percebe-se que os valores médios (Q2) apresentam uma distribuição mais homogênea entre os quartis, indicando uma certa consistência nas previsões. As medianas estão centradas nos *boxplots*, sugerindo uma simetria na distribuição

dos dados. A dispersão, indicada pelo intervalo interquartil (IQR), mostra-se relativamente estável, o que pode ser interpretado como uma variabilidade moderada nas previsões.

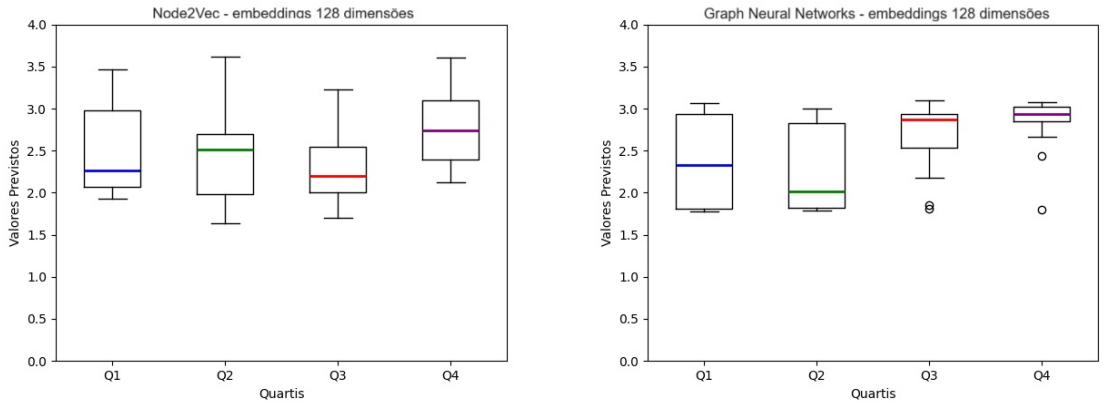
Já no modelo *Graph Neural Networks*, os *boxplots* revelam uma distribuição ligeiramente diferente. A mediana do Q1 parece mais elevada comparada ao *Node2Vec*, sugerindo que o modelo *Graph Neural Networks* pode ter uma tendência a prever valores ligeiramente superiores para o quartil inferior da distribuição. Por outro lado, o terceiro quartil (Q3) e a mediana parecem estar mais próximos, o que pode indicar uma menor dispersão dos valores mais altos, mas também uma potencial assimetria, com uma cauda inferior mais longa.

Os valores mínimos e máximos dentro dos limites aceitáveis são similares entre os dois modelos, embora o *Graph Neural Networks* pareça ter outliers mais distantes, particularmente no Q4. Esses outliers podem ser devido a peculiaridades nos dados que o modelo *Graph Neural Networks* não conseguiu captar tão bem quanto o *Node2Vec*.

A assimetria é menos evidente nos *boxplots* do *Node2Vec*, enquanto no *Graph Neural Networks*, há indicações de uma distribuição ligeiramente assimétrica, especialmente nos quartis Q1 e Q4. Isso pode ser um indicativo de que o modelo *Graph Neural Networks* tem dificuldades em prever valores que se desviam da norma.

Em termos de dispersão, ambos os modelos parecem ter uma dispersão similar, embora o *Graph Neural Networks* mostre uma dispersão ligeiramente menor no Q3 e uma maior dispersão no Q1, o que pode ser evidenciado pela presença de outliers (Figura 5.7).

Figura 5.7: *Boxplots* comparando os quartis preditos utilizando *embeddings* de 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*, destacando as diferenças de dispersão e assimetria entre os modelos.



Fonte: Elaborado pelo autor.

Os resultados obtidos a partir da análise dos *boxplots* dos valores previstos utilizando *embeddings* concatenados de 8, 64 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*, revelam informações importantes sobre a distribuição dos dados preditos em cada caso.

Para os *embeddings* concatenados de 8 dimensões para *Node2Vec* e *Graph Neural Netwoks*, percebe-se uma distribuição dos dados com mediana (Q2) estável, sugerindo uma consistência na centralidade dos valores previstos. O primeiro quartil (Q1) e terceiro quartil (Q3) demonstram uma dispersão moderada, indicada pela extensão do IQR, o que aponta para uma variabilidade significativa entre os dados mais centrais. Valores mínimos e máximos estão distribuídos de forma equilibrada, sem indícios claros de assimetria, embora alguns *outliers* sugiram pontos de atenção pontuais na análise.

A transição para os *embeddings* concatenados de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks* não apresenta mudanças drásticas na mediana, indicando que o aumento das dimensões dos *embeddings* não alterou substancialmente a centralidade dos dados. Entretanto, observa-se um leve aumento na dispersão, como mostrado pelo IQR ligeiramente mais largo. Os *outliers* são mais evidentes nesse cenário, sugerindo que a modelagem com maior dimensionalidade pode estar capturando variações mais extremas nos dados ou possíveis anomalias.

Já nos *embeddings* concatenados de 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*, a mediana mantém-se alinhada com as observações anteriores, o que pode indicar que a centralidade dos dados é robusta às mudanças nas dimensões dos *embeddings*. O IQR parece ser comparável aos *embeddings* concatenados de 64 dimensões, o que sugere uma dispersão similar. Neste caso, os outliers parecem ser menos proeminentes, o que pode indicar uma melhoria na consistência dos valores previstos ou uma filtragem mais eficaz de variações atípicas.

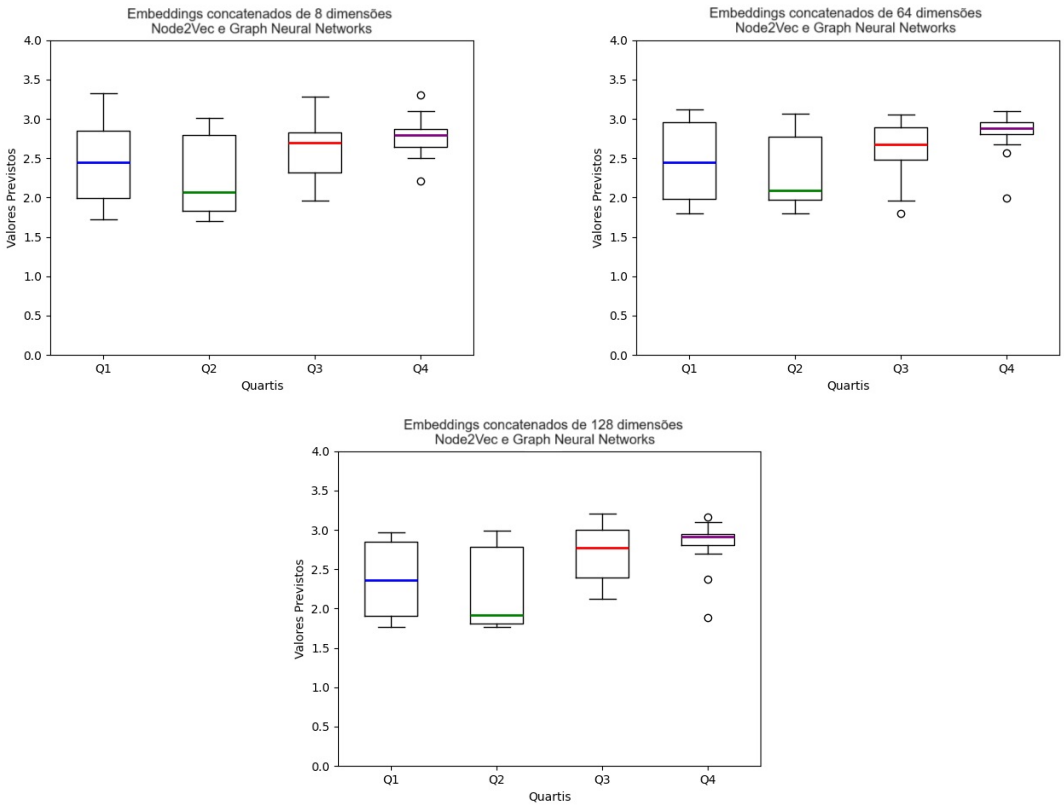
Comparando os três grupos, não se observam diferenças substanciais nas medianas, sugerindo que a centralidade das previsões é mantida independentemente da dimensão dos *embeddings*. O IQR, um indicador de dispersão, aumenta ligeiramente com o crescimento das dimensões, o que pode sugerir um aumento na variabilidade dos dados com *embeddings* mais complexos. Os outliers presentes em todas as dimensões reforçam a necessidade de uma investigação mais aprofundada sobre a natureza desses pontos extremos (Figura 5.8).

Os *boxplots* fornecidos representam a distribuição dos quartis preditos utilizando *embeddings* subtraídos de 8, 64 e 128 para *Node2Vec* e *Graph Neural Netwoks*, respectivamente. Através da análise comparativa destas visualizações, pode-se inferir várias características sobre os dados e o impacto das dimensões dos *embeddings* na predição dos valores.

Observando a mediana (Q2), que é a linha central dentro do boxplot, nota-se que ela se mantém relativamente estável entre os três *boxplots*, indicando uma consistência na tendência central dos valores preditos independentemente do número de dimensões utilizadas.

Quanto ao primeiro quartil (Q1), este representa o valor abaixo do qual 25% dos dados estão situados. Nos *embeddings* subtraídos de 8 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*, o Q1 parece ser um pouco mais elevado em comparação aos

Figura 5.8: *Boxplots* comparando os quartis preditos utilizando o somatório dos *embeddings* de 8, 64 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*.



Fonte: Elaborado pelo autor.

embeddings subtraídos de 64 dimesões para *Node2Vec* e *Graph Neural Netwoks*, sugerindo uma ligeira elevação nos valores mais baixos das predições à medida que aumentamos o número de dimensões dos *embeddings*.

O terceiro quartil (Q3), que indica o valor abaixo do qual 75% dos dados se encontram, apresenta variações mais notáveis entre os três grupos. Os *embeddings* subtraídos de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks* parecem ter um Q3 ligeiramente menor do que os outros dois, o que pode indicar uma concentração mais densa de dados em valores mais baixos.

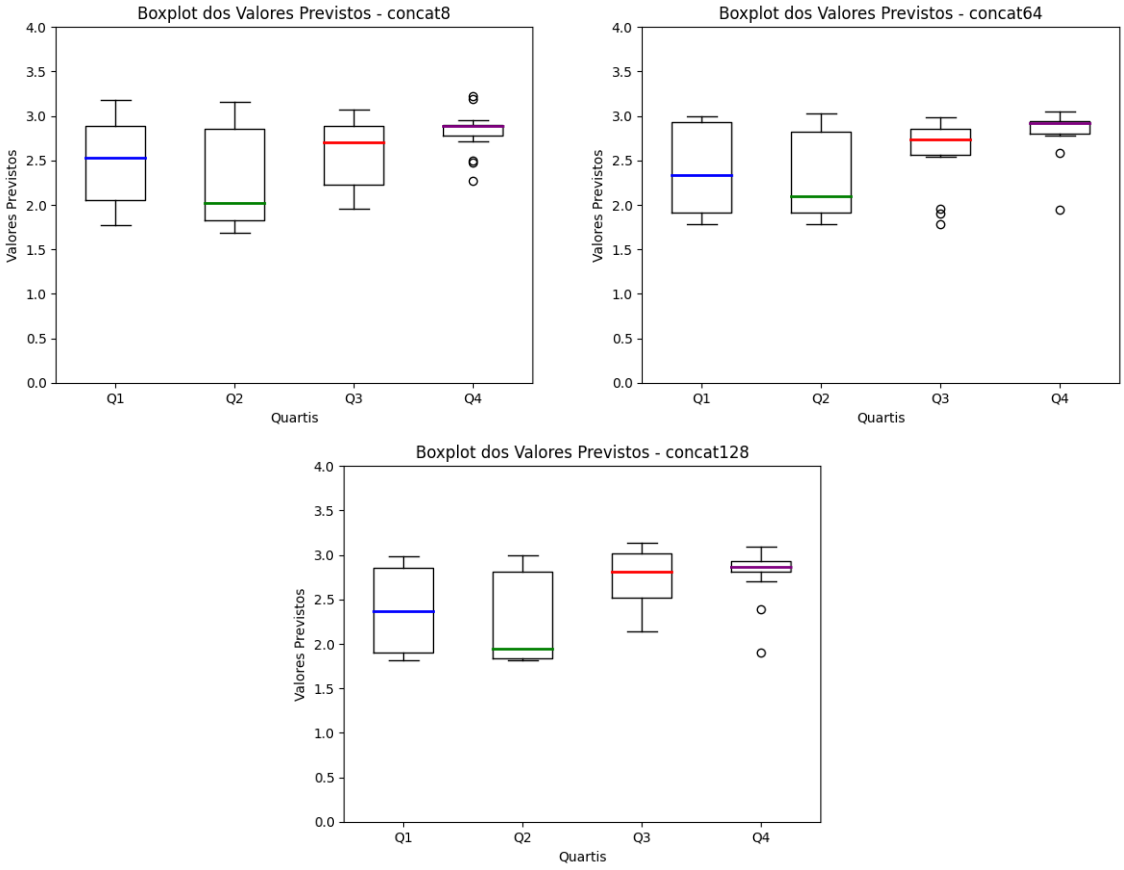
O intervalo interquartil (IQR), que é a diferença entre Q3 e Q1, é uma medida de dispersão da metade central dos dados. Através dos *boxplots*, percebe-se que o IQR varia entre os grupos, com os *embeddings* subtraídos de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks* mostrando uma dispersão um pouco maior, o que sugere uma variabilidade mais elevada nos valores preditos quando comparado aos *embeddings* subtraídos de 8 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*. Os valores mínimo e máximo, o qual são os extremos dos dados não considerados outliers, também variam entre os *boxplots*. Os *embeddings* subtraídos de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks* apresentam um valor máximo mais elevado e a presença de outliers, indicando que esse conjunto de dimensões pode estar associado a uma maior variação nos valores preditos.

Em termos de assimetria, não se observa uma assimetria significativa em nenhum dos grupos, visto que a mediana parece estar centralizada dentro da caixa neles todos, e as caudas não mostram diferenças drásticas em comprimento.

Quanto à dispersão geral, avaliada pela largura do IQR, percebe-se que ela varia entre os grupos, mas não de maneira consistente à medida que se aumenta o número de dimensões. Isso sugere que outras variáveis podem estar influenciando a dispersão além do número de dimensões dos *embeddings*.

Finalmente, ao comparar os grupos entre si, os *boxplots* facilitam a visualização das diferenças nas distribuições dos valores preditos. Enquanto os *embeddings* subtraídos de 64 dimensões para *Node2Vec* e *Graph Neural Netwoks* mostram maior variabilidade e potenciais valores extremos, os *embeddings* subtraídos de 8 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks* parecem ter distribuições mais concentradas, embora os *embeddings* subtraídos de 128 dimensões para *Node2Vec* e *Graph Neural Netwoks* apresente valores ligeiramente mais elevados no geral (Figura 5.9).

Figura 5.9: *Boxplots* comparando os quartis preditos utilizando a subtração dos *embeddings* de 8, 64 e 128 dimensões para *Node2Vec* e *Graph Neural Netwoks*.



Fonte: Elaborado pelo autor.

Ao observar o boxplot dos valores previstos utilizando as características do grafo, notamos variações significativas entre os quartis. O Q1 apresenta uma mediana próxima de 2.0, indicando que a centralidade média dos vizinhos, o *clustering* e o *pagerank* podem

ter valores baixos, sugerindo uma menor importância ou conexão desses nós dentro do grafo. Além disso, a presença de *outliers* abaixo do primeiro quartil pode indicar casos atípicos de nós com características de grafo muito diferentes da tendência geral.

O Q2, com a mediana em verde, mostra uma distribuição mais concentrada de valores previstos, com mediana levemente superior a 2.0 e uma amplitude interquartílica (distância entre o primeiro e o terceiro quartil) menor, indicando menor variabilidade nas características de centralidade como a *closeness centrality* e a *load centrality*.

O Q3, por sua vez, mostra uma mediana indicada pela linha vermelha, situada em um valor próximo de 3.0. Isso pode sugerir que nós representados neste quartil possuem uma posição mais central no grafo em termos de *betweenness centrality*, o que pode indicar um papel de intermediário importante na transferência de informações ou na conectividade do grafo.

Por fim, o Q4 mostra uma variabilidade considerável, com uma amplitude interquartílica maior e uma mediana indicada pela linha roxa, também próxima a 3.0. Isso pode ser interpretado como uma indicação de que os nós nesse quartil possuem muita centralidade em algumas das métricas, mas não em todas, o que pode refletir uma heterogeneidade nas funções ou papéis desses nós dentro da rede.

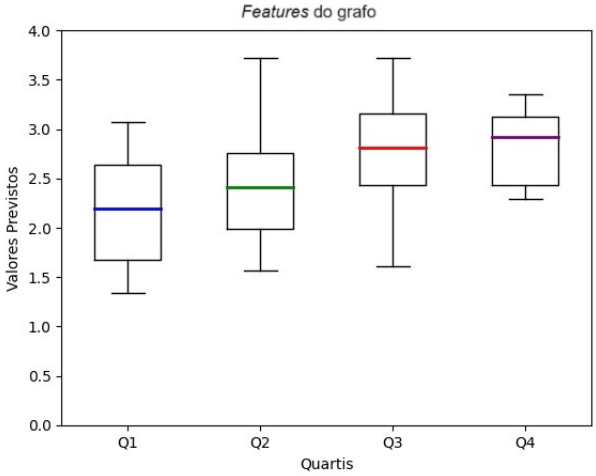
A análise dos *boxplots* sugere que há uma variação notável nas características de centralidade e importância dos nós entre os diferentes quartis. Essa variação pode ser devida à estrutura inerente da rede ou às diferentes funções que os nós desempenham dentro dela. Por exemplo, nós com alta *betweenness centrality* podem atuar como pontes entre diferentes partes do grafo, enquanto aqueles com alta *degree centrality* podem ser importantes para a coesão dentro de suas respectivas comunidades.

É importante notar que *outliers* podem tanto representar erros na coleta de dados ou na modelagem quanto indicar casos especiais que merecem atenção adicional em análises futuras. Além disso, a análise de *boxplot* por si só não é suficiente para entender completamente as dinâmicas complexas de um grafo e deve ser complementada com outras análises de rede, como a investigação das propriedades topológicas ou a visualização da estrutura da rede (Figura 5.10).

Os resultados obtidos pela análise dos valores *SHAP* para as *features* baseadas em grafos revelam *insights* significativos sobre o impacto dessas características na saída do modelo. O *SHAP* (*SHapley Additive exPlanations*) fornece uma medida do impacto de cada característica na previsão do modelo, considerando a contribuição marginal de cada uma quando combinada com outras.

A centralidade de grau (*degree centrality*) demonstra uma variação significativa nos valores *SHAP*, indicando uma influência considerável na saída do modelo. Valores positivos de *SHAP* sugerem que muita centralidade tende a aumentar a previsão do modelo, enquanto valores negativos apontam para uma diminuição. A centralidade de proximidade (*closeness centrality*) e a centralidade de carga (*load centrality*) seguem uma

Figura 5.10: *Boxplots* mostrando a distribuição dos quartis previstos com base em métricas do grafo: *degree centrality, closeness centrality, load centrality, harmonic centrality, betweenness centrality, average neighbor degree, clustering* e *pagerank*.



Fonte: Elaborado pelo autor.

tendência similar, indicando que a posição estratégica de um nó dentro do grafo é um preditor relevante do resultado.

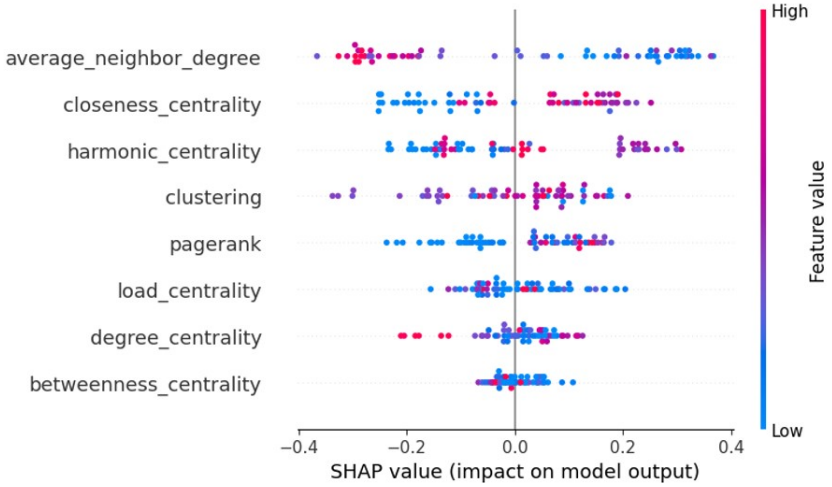
Por outro lado, a centralidade harmônica (*harmonic centrality*) mostra uma distribuição mais concentrada em torno de valores SHAP baixos, o que pode indicar uma influência menos pronunciada no resultado do modelo em comparação com outras métricas de centralidade. A centralidade de intermediação (*betweenness centrality*) apresenta uma gama ampla de valores SHAP, refletindo a sua capacidade de capturar a importância de um nó como intermediário nas comunicações entre outros pares de nós.

O grau médio dos vizinhos (*average neighbor degree*) exibe uma dispersão notável nos valores SHAP, sugerindo que a conectividade dos vizinhos de um nó tem um papel variável na previsão do modelo. Já o coeficiente de aglomeração (*clustering*) revela que agrupamentos locais dentro do grafo podem ter tanto efeitos positivos quanto negativos sobre a saída do modelo, dependendo da configuração específica da rede.

Finalmente, o *pagerank*, uma métrica clássica de importância baseada na estrutura de *links* de um grafo, mostra uma influência moderada, com valores SHAP distribuídos de maneira relativamente equilibrada entre impactos positivos e negativos.

A interação entre essas *features* indica a complexidade e a riqueza das informações codificadas na estrutura do grafo. A análise *SHAP* sugere que não apenas a importância individual de cada nó, mas também a sua conectividade e o papel dentro da rede mais ampla, são fundamentais para compreender e prever os fenômenos modelados. Estes resultados reforçam a relevância de abordagens baseadas em teoria dos grafos para análise de dados complexos e fornecem uma base sólida para pesquisas futuras nessa direção.

Figura 5.11: Gráfico de valores de *SHAP* mostrando a importância das características do grafo: *degree centrality, closeness centrality, load centrality, harmonic centrality, betweenness centrality, average neighbor degree, clustering* e *pagerank*, na previsão do modelo.



Fonte: Elaborado pelo autor.

Capítulo 6

Conclusão

O estudo demonstrou que a aplicação de técnicas de Aprendizado de Máquina em grafos pode prever a popularidade dos atos normativos brasileiros relacionados à segurança alimentar. Utilizando o *BERTopic* para identificar tópicos relevantes e técnicas como *Graph Neural Networks* (GNN) e *Node2Vec*, foi possível mapear as normas mais discutidas e aceitas. Esses resultados fornecem insights valiosos para a elaboração e implementação de políticas públicas.

Os benefícios para a sociedade incluem a possibilidade de direcionar esforços para fortalecer e melhorar as normas mais populares, promovendo um ambiente regulatório mais eficiente. A predição da popularidade dos atos normativos também pode facilitar a comunicação entre o governo e a população, garantindo a divulgação de informações relevantes e de interesse público.

Na segurança alimentar, a previsão das normas mais populares pode garantir que os regulamentos mais eficazes sejam priorizados, resultando em uma melhor fiscalização e redução de riscos de contaminação e fraudes alimentares. Para o direito, o estudo oferece uma ferramenta para análise e gestão de normas jurídicas, ajudando legisladores a identificar normas que precisam de revisão, melhorando a coerência e qualidade das leis. Contudo, algumas limitações foram observadas. A complexidade dos atos normativos e a variabilidade na qualidade das fontes de dados podem ter influenciado os resultados. Além disso, a seleção de parâmetros para os modelos de Aprendizado de Máquina e a definição de popularidade como métrica de avaliação podem ter impactado a precisão das predições.

A importância das técnicas utilizadas reside na capacidade de transformar dados normativos complexos e extensos em informações acionáveis e compreensíveis. O uso de *Graph Neural Networks* (GNN) e *Node2Vec* foi crucial para capturar a estrutura e as interrelações dos atos normativos, permitindo uma análise detalhada e precisa das conexões e influências entre diferentes normas. A consistência dessas técnicas é demonstrada pela robustez dos resultados obtidos, que foram validados por meio de experimentos rigorosos. A relevância das técnicas também é evidente na aplicação prática dos resultados, oferecendo uma ferramenta poderosa para a gestão de políticas públicas e a melhoria contínua do arcabouço regulatório. Em suma, a combinação dessas técnicas de Aprendizado de

Máquina proporciona um avanço significativo na análise e predição da popularidade dos atos normativos, contribuindo para um sistema legal mais transparente e eficaz.

Capítulo 7

Sugestões para Estudos Futuros

Para futuras pesquisas, sugere-se a expansão do conjunto de dados para incluir um espectro mais amplo de atos normativos, abrangendo diferentes áreas além da segurança alimentar. Isso permitiria uma análise mais diversificada e abrangente das tendências legislativas.

Outra recomendação seria explorar métodos alternativos de modelagem de tópicos e de representação de grafos, a fim de comparar e melhorar potencialmente a precisão das predições. Além disso, seria interessante investigar a relação entre a popularidade dos atos normativos e outros fatores, como impacto econômico ou social, para fornecer uma compreensão mais holística de sua relevância.

Por fim, seria produtivo aplicar as técnicas utilizadas neste estudo em outros contextos legislativos, tanto no Brasil quanto em outros países, para validar a eficácia das abordagens de Aprendizado de Máquina em diferentes cenários normativos e culturais.

Referências

Aloise, D. and Cruz, J. S. (2001). Teoria dos grafos e aplicações. Centro de Ciências Exatas e da Terra, Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Natal, Rio Grande do Norte.

Angelov, D. (2020). Top2vec: Distributed representations of topics. Cornell University.

Aritrasen.com (2022). Rede neural de gráfico – passagem de mensagens (gcn) – 1.1.

Bianchi, F., Terragni, S., and Hovy, D. (2021). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. Cornell University.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 127–134.

Blei, D. M. and Lafferty, J. D. (2005). A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.

Bronstein, M. M. et al. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. Cornell University.

Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer Berlin Heidelberg.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Chowdhury, G. (2005). Natural language processing. *Annual Review of Information Science and Technology*, 37(1):51–89.

da Agricultura e Pecuária, M. (2023). Home.

de Menezes Soares, F. (2019). *Elaboração Legislativa em Direito Agroalimentar*. Editora Tribo Ilha, Florianópolis.

Gal, A. (2018). The tangle: an illustrated introduction-part 3: Cumulative weights and weighted random walks. Official IOTA blog.

Godec, P. (2018). Graph embeddings — the summary. Medium.

Goldbarg, M. and Goldbarg, E. (2012). *Grafos: Conceitos, Algoritmos e Aplicações*. Elsevier, Rio de Janeiro.

Gomes, D. S. (2010). Inteligência artificial: conceitos e aplicações. *Revista Olhar Científico*, 1(2):234–246.

Grootendorst, M. (2021). Frequency in topic over time chart. GitHub.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. Cornell University.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Hamilton, W. L. (2021). *Graph Representation Learning*, volume 14 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. CoRR, abs/1609.02907.

Liddy, E. D. (2001). Natural language processing.

Luckin, R. et al. (2016). *Intelligence Unleashed An argument for AI in Education*. Pearson, Londres.

Luxburg, U. and Schölkopf, B. (2011). Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*. North-Holland.

Maia, A. L. L. M. (2023). *Uma abordagem baseada em aprendizagem de máquina e grafos para segmentação de páginas*. PhD thesis, Universidade de São Paulo, São Paulo. Tese (Doutorado em Ciências da Computação).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe. Curran Associates Inc.

Murphy, B. (2014). Latent dirichlet allocation.

Murphy, R. L. et al. (2018). Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. Cornell University.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Pham, T. et al. (2017). Column networks for collective classification. *Proceedings of the AAAI conference on artificial intelligence*, 31(1):2485–2491.

Rodrigues, G. E. P. (2022). Detecting cryptographic misuse with machine learning. Master's thesis, Universidade Estadual de Campinas, Campinas. Dissertação (Mestrado em Ciência da Computação).

Russell, S. J. (2004). *Inteligência Artificial*. Elsevier, Rio de Janeiro.

Selsam, D. et al. (2018). Learning a sat solver from single-bit supervision. Cornell University.

Silva, D. N. R., Ziviani, A., and Porto, F. (2019). Aprendizado de máquina e inferência em grafos de conhecimento. In *Tópicos em gerenciamento de dados e informações*. Sociedade Brasileira de Computação, Fortaleza.

Szwarcfiter, J. L. (1984). *Grafos e Algoritmos Computacionais*. Editora Campus, Rio de Janeiro.

Veličković, P. et al. (2017). Graph attention networks. Cornell University.

Xu, K. et al. (2018). Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR.