



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Estudo Comparativo para Classificação Automática de Viés Político avaliados com Técnicas de Explicabilidade

Gabriel Correia de Vasconcelos

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. Thiago Paulo Faleiros

Brasília  
2022



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**Estudo Comparativo para Classificação Automática  
de Viés Político avaliados com Técnicas de  
Explicabilidade**

Gabriel Correia de Vasconcelos

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Thiago Paulo Faleiros (Orientador)  
CIC/UnB

Prof.a Dr.a Roberta Barbosa Oliveira    Prof. Dr. Geraldo Pereira Rocha Filho  
CIC/UnB    CIC/UnB

Prof. Dr. João José Costa Gondim  
Coordenadora do Curso de Engenharia da Computação

Brasília, 22 de abril de 2022

# Dedicatória

Dedico este trabalho a minha família, que me apoiou de formas que nunca conseguirei retribuir. “Se eu vi mais longe, foi por estar sobre ombros de gigantes” e estes gigantes são minha mãe Andrea, meu pai Sandro e minha irmãzinha Maria Cecília, que sempre me proveram com o melhor que podiam oferecer, me incentivaram a estudar muito e trabalhar duro para alcançar meus objetivos e sempre serão uma companhia fundamental nesta longa jornada.

Sem o amor de vocês, nada disto seria possível.

# Agradecimentos

Agradeço ao Prof. Dr. Thiago Faleiros pelo tempo cedido para orientação e atencioso acompanhamento do desenvolvimento do trabalho e experimentos. Agradeço também aos professores dos departamentos de Matemática, Engenharia Elétrica e Ciência da Computação que marcaram minha trajetória no curso de Engenharia de Computação.

Aproveito para agradecer a comunidade que promove o acesso público e igualitário à informação científica e incentiva um ecossistema de livre compartilhamento de conhecimento. Sem eles, este trabalho não seria possível.

Agradeço à UnB pela oportunidade de ingresso a um Ensino Superior de qualidade e de forma pública e gratuita. Neste lugar, também pude conhecer pessoas que se tornaram grandes amigos, que me acompanharam nesta jornada e para sempre terão um lugar no meu coração.

# Resumo

O viés, seja explícito ou implícito, é um fenômeno inevitável na produção de textos. Estimar o viés político em um texto de forma manual é um processo laborioso e demorado, que necessita da ajuda de seres humanos qualificados e que também podem adicionar seus vieses. Com o auxílio da tecnologia, este trabalho se propõe a estimar o viés político de textos de forma automática, a partir de modelos de aprendizagem de máquina supervisionada. São realizados diversos experimentos que comparam um conjunto de classificadores a partir de métricas de avaliação tradicionais, que atingem um macro *F-1 score* de 0.61. Entretanto, ainda faltam trabalhos que tragam explicações interpretáveis para quais fatores influenciaram na classificação do viés. Por isto, este projeto utiliza um método de explicabilidade chamado LIME, que permite a geração de explicações locais e globais das predições de qualquer modelo. A análise destas explicações contribui para o melhor entendimento dos vieses políticos presentes em qualquer texto e do próprio modelo em si.

**Palavras-chave:** aprendizagem de máquina, processamento de linguagem natural, classificação de viés político, inteligência artificial explicável

# Abstract

Bias, explicit or implicit, is an inevitable phenomenon in the writing of texts. Identifying the political bias in a text manually is a time-consuming and laborious process that requires qualified human attention and can add its own biases as well. Powered by technology, this work proposes to estimate the political bias of texts automatically, based on supervised machine learning models. A set of experiments were designed to compare a variety of classifiers based on traditional evaluation metrics, that ultimately achieve a macro  $F-1$  score of 0.61. However, there is still a lack of studies that provide interpretable explanations for which factors influenced the classification of bias. Therefore, this project applies an explainability method called LIME, which allows the generation of local and global explanations of the predictions of any model. The analysis of these explanations contributes to a better understanding of the political biases present in any text and of the model itself.

**Keywords:** machine learning, natural language processing, political bias classification, explainable artificial intelligence

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Definição do Problema . . . . .	2
1.2	Objetivo . . . . .	2
<b>2</b>	<b>Fundamentação Teórica</b>	<b>4</b>
2.1	Aprendizagem de Máquina . . . . .	4
2.2	Classificação . . . . .	7
2.3	Processamento de Linguagem Natural . . . . .	7
2.3.1	Seleção de Atributos do Texto . . . . .	8
2.3.2	Extração de Características em Textos . . . . .	9
2.4	Modelos Estudados . . . . .	10
2.4.1	<i>Naive Bayes</i> Multinomial . . . . .	10
2.4.2	Regressão Logística . . . . .	11
2.4.3	Máquina de Vetores de Suporte . . . . .	12
2.4.4	<i>Extreme Gradient Boosting</i> . . . . .	14
2.4.5	<i>Transformers, BERT e BERTimbau</i> . . . . .	16
2.5	Avaliação e Explicabilidade de Modelos . . . . .	16
2.5.1	Métricas de Avaliação dos Modelos . . . . .	17
2.5.2	Explicabilidade . . . . .	19
<b>3</b>	<b>Revisão Bibliográfica</b>	<b>24</b>
3.1	Texto-come-Dados no Domínio Político . . . . .	24
3.2	Classificação de Viés Político e <i>Manifesto Project</i> . . . . .	25
3.3	Explicabilidade no Domínio Político . . . . .	27
<b>4</b>	<b>Desenvolvimento</b>	<b>29</b>
4.1	Extração da Base de Dados . . . . .	30
4.2	Pré-processamento dos Dados . . . . .	33
4.3	Treinamento e Validação Cruzada . . . . .	36

4.4	Avaliação e Explicabilidade . . . . .	39
4.4.1	Avaliação com dados externos a Base de Dados . . . . .	41
<b>5</b>	<b>Resultados</b>	<b>43</b>
5.1	Comparação entre os modelos . . . . .	43
5.1.1	Modelos treinados com a Base de Dados A . . . . .	44
5.1.2	Modelos treinados com a Base de Dados B . . . . .	47
5.2	Comparação com a Literatura . . . . .	49
5.3	Análise das Explicações Locais e Globais . . . . .	50
5.3.1	Modelo Vencedor treinado com a Base de Dados A . . . . .	50
5.3.2	Modelo Vencedor treinado com a Base de Dados B . . . . .	55
5.4	Aplicação do modelo em Artigos Políticos . . . . .	59
<b>6</b>	<b>Conclusão</b>	<b>64</b>
6.1	Trabalhos Futuros . . . . .	65
	<b>Referências</b>	<b>66</b>
	<b>Apêndice</b>	<b>70</b>
<b>A</b>	<b>Tabela de códigos do Manifesto Project</b>	<b>71</b>
A.1	Base de Dados A . . . . .	71
A.2	Base de Dados B . . . . .	73
<b>B</b>	<b>Explicações Globais das Classificações do BERTimbau</b>	<b>75</b>
B.1	Base de Dados A . . . . .	75
B.2	Base de Dados B . . . . .	81



# Lista de Figuras

2.1	Ilustração que representa os domínios e áreas do conhecimento ligadas a mineração de dados. . . . .	5
2.2	Exemplo de uma função aproximada ajustada, sub ajustada, normal e sobre ajustada (em azul), comparada com as amostras de treino (quadrados em laranja). . . . .	6
2.3	Exemplo de função logística e seu formato sigmoidal que divide as amostras (pontos pretos) em classes, onde o eixo Y representa as classe, sendo $Y = \{0, 1\}$ . . . . .	11
2.4	Imagem que ilustra o hiperplano separando duas classes, círculos pretos e brancos, onde vetores suporte estão circulos e indicados por $H_1$ E $H_2$ . Também está representada a margem mínima, a normal ao hiperplano $w$ e o ponto de origem $b$ . . . . .	13
2.5	Fluxograma de uma árvore de decisão para classificação binária: sim ou não. Os atributos estão representados em azul e as classes, ou folhas, estão em laranja. . . . .	14
2.6	Processo de construção da explicação de uma predição para auxiliar uma decisão humana. Neste exemplo, o modelo tenta prever a doença de um paciente a partir de seus sintomas. Sua explicação é composta pelos principais sintomas que levaram a essa classificação e é informada ao médico, profissional que tomará a decisão final de diagnóstico . . . . .	20
2.7	Exemplo da intuição presente no processo de explicação local do LIME, onde o preditor caixa preta $f$ está representado nas áreas de vermelho e azul e o modelo aproximado calculado é a linha tracejada. A cruz vermelha em negrito representa a amostra escolhida para ser explicada, e as cruces e círculos em volta, bem como seus tamanhos representam as instâncias aleatórias escolhidas e seus pesos por proximidade. . . . .	22

2.8	Exemplo do processo de escolha sub-modular entre várias amostras para explicar o modelo globalmente. As linhas representam amostras e as colunas representam os atributos dispostos na matriz $\mathcal{W}$ . Neste exemplo o atributo f2 tem a maior importância e está tracejado em azul e as amostras selecionadas foram as marcadas em vermelho, pois representam o maior número de atributos (neste caso, todos exceto o atributo f1). . . . .	23
4.1	Fluxograma do desenvolvimento do projeto proposto na monografia. . . . .	29
4.2	Lista com todas as categorias e subcategorias do <i>Manifesto Project</i> , dentro dos sete domínios políticos e seus respectivos códigos. . . . .	32
4.3	Amostra da base de sentenças codificadas. . . . .	32
5.1	Resultados da classificação de categorias e subcategorias na base A por modelo, onde o ponto representa o <i>F-1 score</i> médio e o traço que o corta representa seu intervalo de confiança. . . . .	45
5.2	Resultados da classificação de domínios na base B por modelo, onde o ponto representa o <i>F-1 score</i> médio e o traço que o corta representa seu intervalo de confiança. . . . .	45
5.3	Representação gráfica da matriz de confusão do modelo vencedor treinado com a Base de Dados A, o BERTimbau. Quanto mais intensa a cor, seguindo a escala de cores a direita, mais amostras contém a célula que relaciona classe real nas suas linhas e classe predita nas suas colunas. Os valores foram normalizados em nível dos rótulos preditos ( <i>i.e.</i> coluna a coluna) para que o desbalanceamento da distribuição de amostras por rótulo não afete intensidade da cor das células. . . . .	51
5.4	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>408.0 - “Economic Goals”</b> (em roxo escuro) e categoria predita pelo modelo <b>410.0 - “Economic Growth”</b> (em roxo claro). . . . .	51
5.5	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>408.0 - “Economic Goals”</b> (em roxo) e categoria predita pelo modelo <b>411.0 - “Technology and Infrastructure: +”</b> (em marrom). . . . .	52
5.6	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>408.0 - “Economic Goals”</b> (em roxo) e categoria predita pelo modelo 408.0. . . . .	52
5.7	Explicação global referente a predição da categoria <b>408.0 - “Economic Goals”</b> . . . . .	53

5.8	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>416.2 - “Sustainability: +”</b> (em rosa) e categoria predita pelo modelo <b>501.0 - “Environmental Protection: +”</b> (em rosa claro). . . . .	53
5.9	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>501.0 - “Environmental Protection: +”</b> (em rosa claro) e categoria predita pelo modelo <b>416.2 - “Sustainability: +”</b> (em rosa). . . . .	53
5.10	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>416.2 - “Sustainability: +”</b> (em rosa) e categoria predita pelo modelo 416.2. . . . .	54
5.11	Explicação local referente a uma sentença (caixa de texto à direita) com categoria original <b>501.0 - “Environmental Protection: +”</b> (em rosa claro) e categoria predita pelo modelo 501.0. . . . .	54
5.12	Explicação global referente a predição da categoria <b>416.2 - “Sustainability: +”</b> . . . . .	54
5.13	Explicação global referente a predição da categoria <b>501.0 - “Environmental Protection: +”</b> . . . . .	55
5.14	Representação gráfica da matriz de confusão do modelo vencedor treinado com a Base de Dados B, o BERTimbau. Quanto mais intensa a cor, seguindo a escala de cores a direita, mais amostras contém a célula que relaciona classe real nas suas linhas e classe predita nas suas colunas. Os valores foram normalizados assim como feito na Figura 5.3. . . . .	56
5.15	Explicação local referente a uma sentença (caixa de texto à direita) com domínio original <b>5 - “Welfare and Quality of Life”</b> (em roxo e código 4) e domínio predito pelo modelo <b>4 - “Economy”</b> (em vermelho e código 3). . . . .	56
5.16	Explicação local referente a uma sentença (caixa de texto à direita) com domínio original <b>4 - “Economy”</b> (em vermelho e código 3) e domínio predito pelo modelo <b>5 - “Welfare and Quality of Life”</b> (em roxo e código 4). . . . .	57
5.17	Explicação global referente a predição do domínio <b>5 - “Welfare and Quality of Life”</b> . . . . .	57
5.18	Explicação global referente a predição do domínio <b>4 - “Economy”</b> . . . . .	58
5.19	Explicação local referente a uma sentença (caixa de texto à direita) com domínio original <b>7 - “Social Groups</b> (em rosa e código 6) e domínio predito pelo modelo <b>4 - “Economy”</b> (em vermelho e código 3). . . . .	59

5.20	Explicação local referente a outra sentença (caixa de texto à direita) com domínio original <b>7 - “Social Groups”</b> (em rosa e código 6) e domínio predito pelo modelo <b>4 - “Economy”</b> (em vermelho e código 3). . . . .	59
5.21	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>506.0 - “Education Expansion”</b> (em amarelo). . . . .	60
5.22	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>701.0 - “Labour Groups”</b> (em azul claro). . . . .	60
5.23	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>701.0 - “Labour Groups”</b> (em azul claro). . . . .	61
5.24	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>414.0 - “Economic Orthodoxy”</b> (em marrom). . . . .	62
5.25	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>304.0 - “Political Corruption”</b> (em amarelo). . . . .	62
5.26	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>501.0 - “Environmental Protection”</b> (em rosa claro). . . . .	62
5.27	Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo <b>416.2 - “Sustainability”</b> (em rosa escuro). . . . .	63
B.1	Explicação global referente a predição da categoria <b>202.1 - “Democracy General: +”</b> . . . . .	75
B.2	Explicação global referente a predição da categoria <b>301.0 - “Decentralization”</b> . . . . .	75
B.3	Explicação global referente a predição da categoria <b>303.0 - “Governmental and Administrative Efficiency”</b> . . . . .	76
B.4	Explicação global referente a predição da categoria <b>304.0 - “Political Corruption”</b> . . . . .	76
B.5	Explicação global referente a predição da categoria <b>305.1 - “Political Authority: Party Competence”</b> . . . . .	76
B.6	Explicação global referente a predição da categoria <b>401.0 - “Free Market Economy”</b> . . . . .	76
B.7	Explicação global referente a predição da categoria <b>402.0 - “Incentives: +”</b> . . . . .	77
B.8	Explicação global referente a predição da categoria <b>403.0 - “Market Regulation”</b> . . . . .	77

B.9	Explicação global referente a predição da categoria <b>408.0 - “Economic Goals”</b> . . . . .	77
B.10	Explicação global referente a predição da categoria <b>410.0 - “Economic Growth: +”</b> . . . . .	77
B.11	Explicação global referente a predição da categoria <b>411.0 - “Technology and Infrastructure: +”</b> . . . . .	78
B.12	Explicação global referente a predição da categoria <b>414.0 - “Economic Orthodoxy”</b> . . . . .	78
B.13	Explicação global referente a predição da categoria <b>416.2 - “Sustainability: +”</b> . . . . .	78
B.14	Explicação global referente a predição da categoria <b>501.0 - “Environmental Protection”</b> . . . . .	78
B.15	Explicação global referente a predição da categoria <b>502.0 - “Culture: +”</b> . . . . .	79
B.16	Explicação global referente a predição da categoria <b>503.0 - “Equality: +”</b> . . . . .	79
B.17	Explicação global referente a predição da categoria <b>504.0 - “Welfare State Expansion”</b> . . . . .	79
B.18	Explicação global referente a predição da categoria <b>506.0 - “Education Expansion”</b> . . . . .	79
B.19	Explicação global referente a predição da categoria <b>605.1 - “Law and Order: +”</b> . . . . .	80
B.20	Explicação global referente a predição da categoria <b>701.0 - “Labour Groups: +”</b> . . . . .	80
B.21	Explicação global referente a predição da categoria <b>703.1 - “Agriculture and Farmers: +”</b> . . . . .	80
B.22	Explicação global referente a predição do domínio <b>1 - “External Relations”</b> . . . . .	81
B.23	Explicação global referente a predição do domínio <b>2 - “Freedom and Democracy”</b> . . . . .	81
B.24	Explicação global referente a predição do domínio <b>3 - “Political System”</b> . . . . .	81
B.25	Explicação global referente a predição do domínio <b>4 - “Economy”</b> . . . . .	82
B.26	Explicação global referente a predição do domínio <b>5 - “Welfare and Quality of Life”</b> . . . . .	82
B.27	Explicação global referente a predição do domínio <b>6 - “Fabric of Society”</b> . . . . .	82
B.28	Explicação global referente a predição do domínio <b>7 - “Social Groups”</b> . . . . .	82

# Lista de Tabelas

2.1	Exemplo de uma matriz de confusão para um problema de classificação binária. . . . .	17
2.2	Exemplo de uma matriz de confusão para um problema de classificação multi-classe, quando a classe real e predita são iguais a B. . . . .	17
4.1	Exemplo de uma sentença original, sua versão limpa e sua versão lematizada. . . . .	33
4.2	Distribuição de sentenças nas categorias e subcategorias da base A, ordenada de forma decrescente. . . . .	35
4.3	Distribuição de sentenças nas domínios da base B, ordenada de forma decrescente. . . . .	35
4.4	Hiperparâmetros experimentados na busca em grade de cada modelo classificador. . . . .	38
5.1	Resultados dos modelos classificadores para a Base de Dados A, ou seja, o rótulo que representa o código das categorias e subcategorias. . . . .	44
5.2	Resultados dos modelos classificadores para a Base de Dados B, ou seja, o rótulo que representa o código do domínio. . . . .	44
5.3	Resultados por classe do classificador BERTimbau, treinado com a base A. . . . .	47
5.4	Resultados por classe do classificador BERTimbau, treinado com a base B. . . . .	48
5.5	Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do portal do PT, com um viés positivo no apoio à política de cotas e ao acesso à educação universal. . . . .	60
5.6	Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do MDB, com um viés político ligado a posições da direita tradicional. . . . .	61
5.7	Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do PSL, com um viés político ligado a posições da direita e extrema direita. . . . .	63
A.1	1 . . . . .	71



# Capítulo 1

## Introdução

No contexto atual, a mídia hegemônica gera uma abundante quantidade de conteúdo textual que é facilmente difundida para a sociedade como um todo, graças ao amplo acesso à internet. Apesar da alegação de uma suposta neutralidade, o fato é que todos estes textos não são apenas um relato de fatos, já que até a escolha da pauta e palavras influencia em como essa notícia ou reportagem será percebida pelo público [1]. Como são inevitáveis, entender os vieses de um texto é fundamental para uma compreensão mais ampla, não só do seu significado em si, mas do contexto e com qual intenção este foi produzido.

Em vários exemplos, como em propagandas políticas ou manifestos eleitorais, estes vieses ideológicos de um texto estão claros e ligados diretamente às visões políticas do autor ou da instituição que o produziu. Entretanto, há casos em que o viés está implícito no texto de forma que nem todo o público teria a capacidade de entendê-los diretamente, bem como entender suas consequências. Em última análise, isto dá um grande poder às instituições responsáveis por formular estes textos, que podem influenciar pessoas a tomarem decisões externas a elas e de forma inconsciente. Um exemplo recente disto é o fenômeno das *fake news*, que ao disseminar informações propositalmente falsas e com um viés específico, conseguiram influenciar o resultado de eleições ao redor do mundo [2].

Nestes casos onde os vieses do texto não estão claros, a atribuição de viés pode ser possibilitada através de especialistas qualificados no campo da Ciência Social, mas o grande volume de textos existentes impossibilita a viabilidade deste processo manual em grandes escalas. Esta é uma tarefa que leva tempo, muito trabalho e que também está suscetível aos vieses do próprio avaliador.

A grande vantagem da aplicação da inteligência artificial nestes contextos é poder analisar uma quantidade de dados expressivamente maior que seres humanos, aprender com os padrões de classificação e depois poder replicá-los de forma automática e mais consistente que um ser humano.



## 1.1 Definição do Problema

O uso de algoritmos de aprendizagem de máquina, uma sub-área da inteligência artificial, aplicados no contexto político é amplo [3]. O problema abordado neste trabalho é a classificação automática de viés político em textos, área que vem sendo amplamente estudada recentemente e possibilitada pelos grandes avanços no processamento de linguagem natural e de modelos classificadores cada vez mais complexos.

Este problema é desenvolvido a partir da realização de um experimento comparativo entre um conjunto de modelos classificadores. A seleção de modelos reuniu desde os mais simples, como o *Naive Bayes*, até os mais complexos, como o BERT, modelo baseado em *Transformers* e atual estado da arte para a classificação de bases textuais. Estes modelos são treinados a partir de uma base extraída do *Manifesto Project* [4], que contém manifestos eleitorais de diversos países que foram rotulados manualmente por especialistas, sentença a sentença. O alvo dos modelos é um rótulo que representa categorias de assuntos políticos e juízos de valor associados a eles, por exemplo “Grupos Trabalhistas: Negativo” ou “Liberdade e Direitos Humanos: Positivo”.

Entretanto, como processo de aprendizado dos modelos estatísticos é feito a partir de exemplos já classificados anteriormente por seres humanos, também herda seus vieses, fenômeno chamado de viés algorítmico [5]. Para se reduzir esse efeito, é necessário esclarecer as lógicas internas do aprendizado do modelo e quais foram os parâmetros que levaram a uma determinada classificação, ou seja, é necessário que estas classificações sejam explicáveis e interpretáveis por humanos. Por este motivo, além da avaliação do modelo pelos métodos tradicionais, como acurácia e *F-1 score*, também são geradas explicações para a classificação de sentenças individuais e da classe como um todo. Isto é feito a partir de um método de explicabilidade de inteligência artificial chamado de LIME.

## 1.2 Objetivo

O objetivo do experimento é comparar os modelos, a partir de suas métricas e explicações de classificação, para entender qual traz o melhor resultado final. O modelo vencedor deve permitir uma análise conjunta dos assuntos políticos preditos por ele em cada sentença de um texto, e possibilitar um entendimento mais detalhado dos vieses políticos presentes em qualquer texto, em qualquer outro contexto.

Em complemento, o estudo também busca entender o efeito causado por variações no pré-processamento da base de dados, como o processo de lematização e o de agrupamento de classes nos resultados finais. Também são experimentados diversos conjuntos de hiperparâmetros em cada modelo, como diferentes funções de penalidade, para o melhor

entendimento de como influenciam no resultado final do experimento. Nesta parte, as explicações geradas pelo LIME contribuem bastante para a realização das análises.

Outro objetivo secundário do desenvolvimento dos modelos é facilitar o processo de codificação manual feito pelos especialistas do *Manifesto Project*, para guiar ou sugerir classificações que podem ou não necessitar de uma validação externa.

No Capítulo 2 são apresentados os fundamentos teóricos para o entendimento do experimento, enquanto o Capítulo 3 faz uma pesquisa extensiva na literatura, onde são discutidos os métodos e resultados encontrados pelos principais trabalhos relacionados ao problema abordado nesta monografia. O Capítulo 4, apresenta como foi realizado o experimento, ou seja, demonstra o processo de extração e tratamento da base, treinamento dos modelos e o processo de avaliação dos resultados. No Capítulo 5 são analisados os resultados de classificação encontrados, bem como suas explicações. Ainda, no Capítulo 6 são revisados os objetivos, os resultados encontrados e conclui-se o trabalho.

# Capítulo 2

## Fundamentação Teórica

Nesta seção serão revisados os principais fundamentos teóricos da aprendizagem de máquina, do problema de classificação e do processamento de linguagem natural seguindo os conceitos da literatura mais recente. Em seguida, será explicada brevemente a intuição dos classificadores selecionados neste experimento para o devido entendimento do desenvolvimento do trabalho. Em adição, são discutidos os principais métodos de avaliação tradicionais, complementados por uma análise em detalhe da área de estudos que aborda a explicabilidade de modelos estatísticos.

### 2.1 Aprendizagem de Máquina

O processo conhecido como Aprendizagem de Máquina é uma subárea da Inteligência Artificial e tem seu conceito diretamente ligado a outro conceito mais amplo que é o de Mineração de Dados. Segundo Frawley et al. [6] a Mineração de Dados é definida como processo de descoberta de conhecimento relevante a partir de um grande volume de dados. A definição de relevância refere-se a um conhecimento não trivial, implícito nos dados e não conhecido anteriormente, mas que pode ser útil para o objeto do estudo.

Kaufmann [7] complementa o conceito com um paralelo a ação de buscar pepitas de ouro em toneladas de matéria-prima, onde as pepitas exemplificam a sabedoria obtida e a matéria prima representa as diversas fontes de dados que atualmente são armazenadas em exponencial volume nos diversos sistemas de banco de dados espalhados pelo mundo.

Existem diferentes áreas de pesquisa na Inteligência Artificial que propõem possíveis técnicas de extração, que variam a partir do tipo de dados que se quer minerar e qual o tipo de conhecimento se quer descobrir e a Aprendizagem de Máquina é uma das áreas que possui uma grande confluência com a mineração de dados. A Figura 2.1 ilustra as várias áreas de conhecimento ligadas ao processo de mineração de dados.

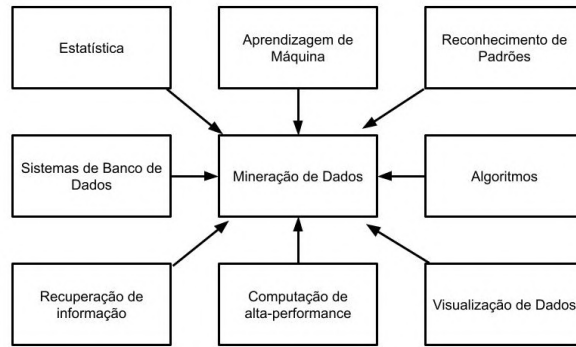


Figura 2.1: Ilustração que representa os domínios e áreas do conhecimento ligadas a mineração de dados (Fonte: [8]).

## Conceito

A intuição da Aprendizagem de Máquina é que uma máquina possa aprender de forma automática, a partir do reconhecimento de padrões complexos em um grande volume de dados e com isso, poder tomar decisões inteligentes dependendo do objetivo a ser alcançado [8]. Bonaccorso [9] explicita que o objeto de estudo é o próprio processo de aprendizagem e a intenção fim é emular, nos meios de um computador, a inteligência humana de forma com que possa reagir a estímulos e a contexto externos e relembrar de experiências passadas para que tome suas próprias decisões de forma adaptativa.

Existem diversas abordagens para os diferentes tipos de problemas que podem ser resolvidos, entretanto, em geral todas envolvem a implementação de um algoritmo que busca encontrar uma função aproximada  $f'(X_i) = y_i$  de uma função desconhecida  $f(x) = y$ , que representa o comportamento de um fenômeno do mundo real.

Como entrada, esta função aproximada recebe dados de uma amostra qualquer  $X_i$ , também chamada de observação e o conjunto de amostras forma uma base de dados  $\mathcal{X}$ , onde  $\mathcal{X} = (X_1, X_2, \dots, X_m)$  e  $m$  é o número total de amostras. Cada  $X_i$ , por sua vez, possui  $n$  atributos (ou *features*), como demonstrado na Equação 2.1 [9].

$$X_i = (x_{i1}, x_{i2}, \dots, x_{in}) \quad (2.1)$$

O retorno de  $f'$  é uma saída  $y_i$ , chamada de predição, que corresponde a um ou mais rótulos, classes ou agrupamentos, onde para cada  $X_i$  existirá um  $y_i$  resultante, tal que  $Y = (y_1, y_2, \dots, y_m)$ . Definições mais restritas tendem a considerar que a saída de  $f'(X_i) = y_i$  é na verdade uma inferência, mais do que uma predição, para evitar a presunção de que seu resultado é de alguma forma “mágico”. Portanto, este resultado é

uma extrapolação de regras gerais que são extraídas pelo modelo e que devem ser inferidas com uma alta precisão [9].

Esta função não pode ser específica demais para os dados atuais, de forma que não consiga generalizar suas previsões para entradas futuras, configurando o efeito chamado de sobreajuste. Também não pode ser generalista demais, de forma que não possua uma acurácia satisfatória para as entradas futuras e desconhecidas, efeito chamado de subajuste ou sub-aprendizado [10]. Ambos casos estão exemplificados na Figura 2.2.

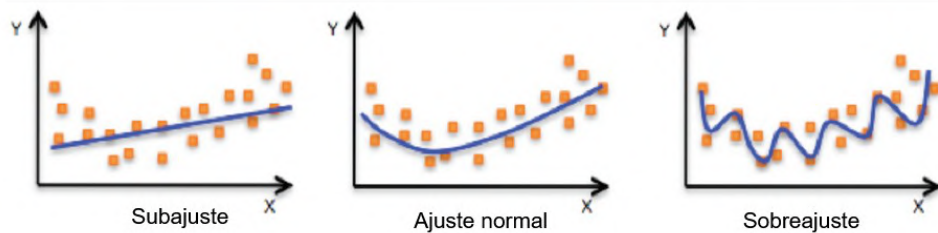


Figura 2.2: Exemplo de uma função aproximada ajustada, sub ajustada, normal e sobre ajustada (em azul), comparada com as amostras de treino (quadrados em laranja) (Fonte: [11]).

## Tipos de aprendizagem

Dependendo de características das amostras e atributos, diferentes técnicas de aprendizagem podem ser elaboradas. As mais comuns serão definidas a seguir [8] [9]:

1. aprendizagem supervisionada: onde o algoritmo que modela a função aproximada tenta encontrar padrões a partir de exemplos de dados com rótulos já conhecidos para prever seus rótulos. Por conhecer os rótulos em sua fase de treinamento, é possível que o erro do modelo seja medido com precisão e por consequência, a maioria dos algoritmos funcionam a partir da lógica da minimização de uma função de perda. Aqui há a esperança que para novas amostras não rotuladas o modelo generalize bem suas previsões.
2. aprendizagem não-supervisionada: onde não existem rótulos conhecidos para as amostras e o algoritmo aprende por meio do agrupamento de similaridades entre as amostras. A função aproximada atribui as amostras em grupos, onde busca-se maximizar a probabilidade da amostra de pertencer ao grupo correto.
3. aprendizagem ativa: é um misto da aprendizagem supervisionada e não-supervisionada com a inclusão de iterações com o usuário para ajudar criar rótulos e atribuí-los a pequenas amostras de observações. Então o modelo tenta aprender com essas escolhas e prever grandes amostras não rotuladas que sejam similares.

Este trabalho concentra seus esforços principalmente na categoria supervisionada, uma vez que os rótulos da base analisada já são conhecidos previamente. Mais especificamente, o trabalho irá buscar realizar a classificação de rótulos da base, processo que será detalhado em seguida.

## 2.2 Classificação

O problema de classificação busca determinar a classe de uma observação dentre certas classes (ou rótulos) discretas e pré-estabelecidas. Este é um processo que acontece em duas etapas: o treinamento e o teste [12].

Assumimos que as amostras são dados na forma  $(X_i, Y_i)$ ,  $X_i$  é o vetor de atributos da  $i$ -ésima amostra, e  $Y_i$  é o rótulo da  $i$ -ésima amostra. Na fase de treinamento, o modelo classificador  $H$  busca encontrar  $H(X_i) = Y_i$  se alimentando com a base em exemplos já classificados previamente e com o objetivo de ajustar seus parâmetros internos para ser capaz de realizar boas previsões de rótulo para os dados novos [13].

Para verificar se o modelo está sendo efetivo, é realizada uma etapa de avaliação dos resultados, onde são utilizadas amostras que são rotuladas, mas ainda desconhecidas para o classificador. Para sua avaliação, detalhada na seção 2.5, são calculadas métricas que tem como base os rótulos preditos comparados com os rótulos já conhecidos. Como esse segmento de dados nunca foi visitado pelo modelo na fase de treino, temos uma melhor noção de como seria sua aplicação com dados do mundo real e podemos avaliá-lo de forma menos parcial.

Por possuir mais de duas classes possíveis, o tipo de problema abordado nesse estudo se encaixa no conceito de classificação multi-classe. Segundo Mehra et al. [13], este tipo é mais delicado e desafiador, pois a maioria dos algoritmos são projetados para realizar um classificação binária, ou seja, apenas duas classes  $Y = \{1, -1\}$ . Entretanto, esta é uma área já bem desenvolvida e atualmente existem várias técnicas para sua resolução, como por exemplo, a adaptação de algoritmos de classificação binária ou divisão do problema multi-classe em múltiplos subproblemas binários.

## 2.3 Processamento de Linguagem Natural

Quando a classificação é feita utilizando produções textuais como entrada ela entra em interseção com outra sub-área da Inteligência Artificial chamada de Processamento de Linguagem Natural, ou PLN. De acordo com Goldberg [14], este campo de estudo aborda todos os problemas que tem como a linguagem humana sua entrada ou saída.

Emular o entendimento e capacidade de extração de contexto dos seres humanos é uma tarefa muito complexa justamente pela ambiguidade e alta variabilidade de nossa linguagem. A linguagem também é composicional e simbólica, de forma que o conjunto de letras formam palavras, o conjunto de palavras formam sentenças e, em diferentes contextos e disposições, podem expressar diferentes significados [14].

Liddy [15] aponta que mesmo com todos esses desafios, o volume massivo de textos que estão digitalizados e disponíveis livremente na internet facilitam o uso de métodos estatísticos para abordar problemas com intermédio da computação. Isto, aliado ao fato de que muitos destes textos estão rotulados de alguma forma, permitiu com que o uso de algoritmos de aprendizado de máquina se tornasse a abordagem padrão para a resolução deste problema.

Como computadores não entendem de fato o que são as letras e palavras, o texto precisa passar por uma etapa de pré-processamento. Esta etapa tem o fim de remover informações que não trazem valor para resolver o problema de classificação e maximizar a extração de informações [16].

### 2.3.1 Seleção de Atributos do Texto

O pré-processamento do texto também pode ser chamado de seleção de atributos. Como cada palavra pertence ao conjunto de atributos, um conjunto de textos pode chegar até 1,000<sup>30</sup> dimensões, e por isso é necessário um processo de seleção de quais são as mais importantes, com o fim de reduzir o tamanho de entradas e facilitar o treinamento do modelo de classificação [17].

O processo de *tokenização* é um dos mais básicos no processamento de linguagem natural e é definido pela divisão de uma sequência de palavras presentes em cada sentença em pedaços menores chamados *tokens*. Estes *tokens* podem ser palavras ou partes de palavras [17] e a cada um é atribuído um índice numérico, que será utilizado como a entrada das próximas etapas do modelo.

Para aprofundar a seleção dos atributos, podem ser removidos números, caracteres especiais (*i.e.* acentos e cifrões), pontuações, palavras de parada (*i.e.* conjunções, alguns tipos de pronomes, preposições, verbos de ligação) e ainda quaisquer palavras com menos de dois ou três caracteres que geralmente são pouco informativos e não agregam sentido ao texto [17]. É interessante também que o texto seja uniformizado, convertendo todos seus caracteres em letras minúsculas ou maiúsculas, para evitar que palavras como “economia” e “Economia” sejam interpretadas como palavras diferentes.

Gentzkow et al. [17] traz a importância de considerar que todo esse processo de remoção de palavras e limpeza do texto tem um *tradeoff*. Da mesma forma que traz uma grande redução de custo computacional e torna os modelos mais interpretáveis, de-

pendendo do contexto do problema, pode apagar uma potencial informação importante, como por exemplo, expressões temporais (por exemplo: “há 5 dias”, “em 22 de agosto”) e possivelmente causar perda de sentido contextual.

Para tornar o método de redução de dimensões ainda mais sofisticado, alguns pesquisadores sugerem o uso de técnicas de normalização de palavras e a estudada nesta monografia foi a lematização [18]. Com o uso de análises morfológicas e de vocabulário, palavras são reduzidas ao seu sufixo, ou seja, sua forma mais básica, chamada de *lemma* [19]. Este processo é feito previamente a *tokenização*.

Deste modo, verbos como “trabalhar” e suas inflexões “trabalhamos”, “trabalhei”, “trabalharia” são todos reduzidos ao mesmo *lemma*: “trabalhar”. Já substantivos e adjetivos são substituídos pela sua versão básica ou a sinônimos, como “maçãs” por apenas “maçã” e “melhor” por “bom” [19].

### 2.3.2 Extração de Características em Textos

Além da redução de atributos, o conteúdo textual precisa ser traduzido para uma linguagem que o modelo estatístico entenda, ou seja, é necessário transformar as palavras em valores numéricos. Existem vários algoritmos que podem dar um significado mensurável para esses valores e que caracterizam esse texto e o algoritmo usado nesse estudo é o *tf-idf* (*term frequency - inverse document frequency*).

Como o nome em inglês diz, o valor atribuído a cada palavra pesa a sua frequência na sentença (*term frequency*) e o inverso da sua frequência no conjunto de sentenças (*inverse document frequency*). Para cada *token*  $j$  em cada sentença  $i$ ,  $c_{ij}$  é a frequência de vezes que  $i$  aparece em  $j$ , que é o resultado de  $tf_{ij}$ , como demonstrado na Equação 2.2. Já o cálculo de  $idf_{ij}$  está demonstrado na Equação 2.3, onde  $n$  é o total de documentos, e represente o valor inverso da frequência de um termo no documento [17]. O valor do *token* é o produto de  $tf \times idf$ .

$$tf_{ij} = c_{ij} \tag{2.2}$$

$$d_j = \sum_i 1_{[c_{ij}>0]} \tag{2.3}$$

$$idf = \log(n/d_j)$$

Esse cálculo leva tanto os termos raros como as palavras que aparecem em vários documentos a possuírem valores menores, de forma que a presença delas pese menos para a classificação final em uma determinada categoria [17] [20]. No final do processo, temos um vetor de valores *tf-idf* de todos os termos para cada sentença, e o conjunto desses vetores que alimentará os modelos classificadores.



## 2.4 Modelos Estudados

Um dos objetivos do trabalho é entender quais dos modelos de classificação existentes melhor se adaptam para o escopo do problema abordado. Por isso, foi selecionada uma coleção de modelos que vão desde os mais tradicionais da aprendizagem supervisionada, passando pelos mais usados na prática para classificar textos, até chegar em modelos que atualmente são considerados estado da arte para a classificação de textos. Os próximos parágrafos descrevem brevemente como funcionam esses modelos, bem como suas potenciais vantagens e desvantagens.

### 2.4.1 *Naive Bayes* Multinomial

O *Naive Bayes* é um dos modelos mais básicos de classificação probabilística. Como seu nome em inglês já diz, é um modelo ingênuo, pois não considera em seu algoritmo a correlação entre os seus atributos ao realizar a seu treinamento, ou seja, assume que todos os atributos são independentes.

Neste modelo, o teorema de Bayes define como é calculada a probabilidade de uma observação de  $x$  ser da classe  $y$  e está demonstrada na Equação 2.4 [21]. A suposição da independência de todos os atributos é descrita na Equação 2.5.

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (2.4)$$

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y) \quad (2.5)$$

Portanto, o classificador *Naive Bayes* pode ser descrito a partir da Equação 2.6.

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y) \quad (2.6)$$

O modelo estudado ainda é parametrizado por um vetor peso  $\theta_y = (\theta_{y_1}, \dots, \theta_{y_n})$  para cada classe  $y$ , e por isso é chamado de multinomial. Esses parâmetros estimados  $\hat{\theta}_y$  são calculados a partir da equação de máxima verossimilhança, suavizada por uma constante  $\alpha$  e seu cálculo está demonstrado na Equação 2.7

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (2.7)$$

Como cada classificação nesse modelo é independente, é comum ser utilizado quando os atributos que descrevem as instâncias analisadas são independentes entre si. Além disso, por essa mesma independência, não precisa de um grande número de dados de teste para concluir classificações com uma relativa boa precisão. Por essas características e sua

simplicidade computacional, seus resultados são interessantes para serem considerados como base de referência para qualquer experimento de classificação [22].

Todavia, como apontado por Rennie et al. [23], mesmo com a aplicação de técnicas de aprimoramento dos resultados, como o uso do tf-idf citado na subseção 2.3.2, o modelo possui vários erros sistêmicos. Estes erros levam o algoritmo a tomar decisões erradas de classificação, ao favorecer uma classe sobre a outra de forma inapropriada e por isso o torna um modelo limitado a análises mais simples [23].

## 2.4.2 Regressão Logística

Neste modelo, uma função logística é usada para modelar as probabilidades das possíveis classificações de uma observação dependendo dos valores de suas variáveis independentes, que podem ser categóricas ou contínuas [24]. Isto é, este modelo representa uma forma de atribuir significado estatístico para cada atributo a respeito da sua capacidade de classificar uma observação e a partir de uma função linear [25].

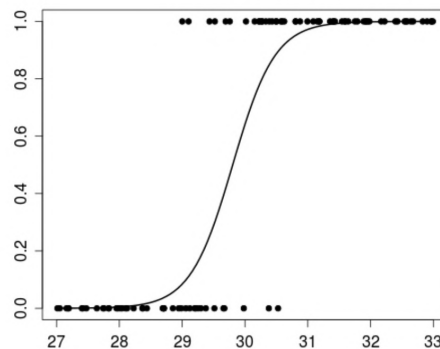


Figura 2.3: Exemplo de função logística e seu formato sigmoide que divide as amostras (pontos pretos) em classes, onde o eixo Y representa as classe, sendo  $Y = \{0, 1\}$  (Fonte: [24]).

Assumindo um problema de classificação binária de classes  $Y = \{1, 0\}$ , as variáveis independentes  $X_i = (x_{i0}, x_{i1}, \dots, x_{im})$  representam os atributos,  $\beta = (\beta_0, \beta_1, \dots, \beta_m)$  são os pesos de cada atributo, a função linear que modela o problema está representada na Equação 2.8.

$$Y = \beta_0 + \beta_1 x_{i0} + \dots + \beta_{m+1} x_{im} \quad (2.8)$$

Na regressão logística, essa função tem formato sigmoide e está representada na Equação 2.9 e ilustrada na Figura 2.3, onde  $P(y = 1)$  é a probabilidade de uma observação ter sua classe igual a 1. O classificador é treinado para escolher os parâmetros  $\beta$  que

maximizem a verossimilhança mostrada na Equação 2.10, onde  $n$  é o número de atributos [24].

$$P(y = 1) = \frac{1}{1 + e^{-(\beta^T X)}} \quad (2.9)$$

$$\prod_{m=1}^n p(y_m | X_m) \quad (2.10)$$

Aly [26] descreve que para problemas multi-classe, com  $j$  classes distintas e  $Y = y_0, y_1, \dots, y_j$ , uma técnica possível é conhecida como *One-vs-Rest*, do inglês um contra e resto. Esta divide o problema em várias classificações binárias, onde para cada classe  $y_j$  é calculada uma regressão quem tem como rótulo possíveis: ser da classe ou não ser da classe (*i.e.*  $Y = \{y_j, \neg y_j\}$ ).

A vantagem deste modelo é que não requer grande poder computacional e é simples de ser implementado. Seus resultados são interpretáveis e muito intuitivos, justamente pelos pesos que são definidos para cada atributos ajudarem a entender como estes influenciam a classificação final [25]. Aggarwal et al. [27] mostra ainda que, na classificação textual, quando as observações da base de dados pertencem ao mesmo domínio de conhecimento, algumas palavras podem ter uma importância para a classificação maior que outras e a aplicação da regressão logística se encaixa perfeitamente.

Ao mesmo tempo, essa característica pode se tornar uma desvantagem, ao deixar o modelo muito suscetível a um sobreajuste, ou seja, incorretamente atribuindo um peso muito alto em palavras das amostras presentes na fase de treinamento do modelo que nem sempre podem estar presentes nas amostras futuras [28].

### 2.4.3 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte, ou do inglês, SVM (Support Vector Machine), é um método com uma teoria robusta e com excelentes resultados empíricos na classificação de textos [29]. Ele consiste na definição de um hiperplano, ou conjunto que hiperplanos, que separam as amostras em diferentes classes, como demonstrado na Figura 2.4.

O hiperplano ótimo escolhido é o que representa a maior margem de separação entre as classes, ou mais especificamente, o que possua o menor erro de generalização. Um SVM é treinado a partir do problema de otimização representado na Equação 2.11 e limitado pelas restrições da Equação 2.12 [31], onde  $w$  representa a normal ao hiperplano e  $b$  um ponto no hiperplano.

Os dados aqui precisam ser linearmente separáveis e com existência de ruídos nos dados ou *outliers* é inevitável, nem sempre é possível encontrar o hiperplano ótimo. Por esse

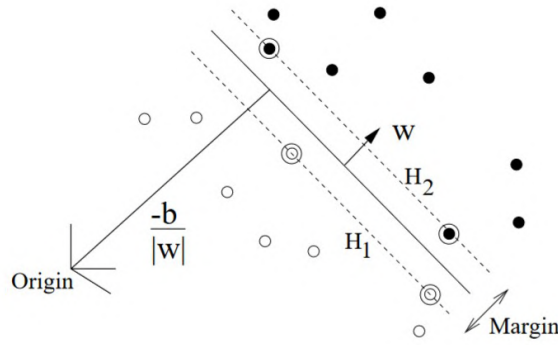


Figura 2.4: Imagem que ilustra o hiperplano separando duas classes, círculos pretos e brancos, onde vetores suporte estão circundados e indicados por  $H_1$  e  $H_2$ . Também está representada a margem mínima, a normal ao hiperplano  $w$  e o ponto de origem  $b$  (Fonte: [30]).

motivo, existem variações do SVM que adicionam uma margem suave para lidar com esse tipo de problema [32]. Isso é representado nas variáveis de folga  $\xi_i$  que são adicionadas as restrições representadas na Equação 2.12 e no parâmetro  $C$  definido pelo usuário, onde quanto maior, maior a penalidade a erros.

$$\operatorname{argmin}_w \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (2.11)$$

$$y_i (X_i \cdot w + b) \geq 1 - \xi_i \forall i \quad (2.12)$$

Quando ainda não é possível se separar as classes com os hiperplanos, a SVM ainda pode ser estendida para sua versão não-linear, mas que pelos experimentos realizados por Rennie e Rifkin [31], não apresenta resultados melhores que a linear para classificação textual e por isso sua aplicação não foi aprofundada neste estudo.

Conceitualmente, SVMs e a Regressão Logística são muito similares e por isso possuem vantagens parecidas, como sua facilidade de implementá-las computacionalmente. Sua diferença principal está nos detalhes de sua implementação e otimização nas fases de treino. A SVM apresenta também uma dificuldade maior de se interpretar seus resultados [32]. Ademais, este é um modelo muito sensível aos seus parâmetros de otimização e uma escolha ruim deles pode trazer resultados muito ruins quando comparados a outros modelos.

Outro ponto importante é que a formulação original da SVM resolve apenas problemas de classificação binários. Existem dois métodos principais para se resolver os problemas multi-classe, mas um deles, que realiza a modificação do algoritmo original, leva a um aumento da complexidade computacional e por isso não é muito utilizado. O outro,

funciona em um processo similar a como é feito na regressão logística mencionada na seção 2.4.2 e realiza a subdivisão do problema em subproblemas binários [33].

#### 2.4.4 *Extreme Gradient Boosting*

O classificador de *Gradient Boosting* é um poderoso algoritmo, altamente adaptável às necessidades de sua aplicação e com grande sucesso na resolução de problemas de aprendizado de máquina [34]. Ele se baseia no algoritmo da árvore de decisão, um modelo mais simples que consiste num fluxograma (como na Figura 2.5) em estrutura de árvore, onde cada nó representa um atributo e cada ramificação um processo de decisão que por fim leve a uma folha, diga-se, a escolha de uma classe final [8].

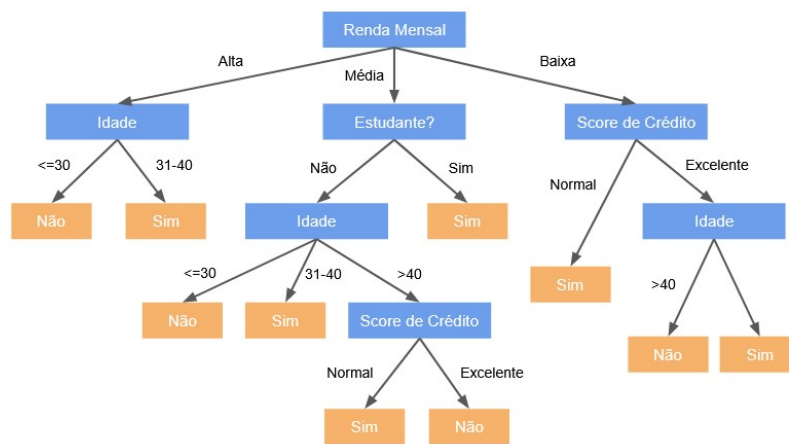


Figura 2.5: Fluxograma de uma árvore de decisão para classificação binária: sim ou não. Os atributos estão representados em azul e as classes, ou folhas, estão em laranja (Fonte: [35]).

O *Gradient Boosting* utiliza um método chamado de *ensemble*, que consiste na combinação de diferentes árvores de decisão que votam e decidem um rótulo para a classificação. Além disso, são atribuídos pesos aos votos de forma com que cada instância de treinamento corrija os erros de classificação de suas predecessoras, processo chamado de *boosting* [8].

A intuição de seu treinamento é encontrar uma função  $H'(X) = y$  em que seu valor seja minimizado por uma função de perda  $\psi(y, H(x))$ , onde  $X$  é o conjunto de atributos e  $y$  é o rótulo, como demonstrado na Equação 2.13 [34].

$$H'(x) = \arg \min_{H(x)} \psi(y, H(x)) \quad (2.13)$$

O treino do classificador é um processo iterativo onde, a cada iteração, os pesos  $\hat{\theta}_i$  da árvore de decisão atual são estimados e aplica-se o método do gradiente (ou *gradient descent*) [34], algoritmo de otimização responsável por escolher os melhores pesos que minimizem a função de perda empírica  $J(\theta)$ . A Equação 2.14 descreve esse processo estimação dos pesos.

$$\begin{aligned}
 J(\theta) &= \sum_{i=1}^N \psi(y_i, h(x_i, \hat{\theta})) \\
 \nabla J(\theta) &= \{\nabla J(\theta_i)\} = \left[ \frac{\partial J(\theta)}{\partial J(\theta_i)} \right]_{\theta=\hat{\theta}^t} \\
 \hat{\theta}_t &\leftarrow -\nabla J(\theta)
 \end{aligned}
 \tag{2.14}$$

A escolha de  $\psi(y, H(x))$  varia de acordo com o tipo do rótulo que se quer classificar, com a característica mais comum entre elas sendo a sua resistência a *outliers*, ou seja, a valores extremos. Para rótulos contínuos, exemplos são a função Gaussiana ou de Laplace. Já para problemas categóricos, como os presentes nesta monografia, podem ser escolhidas funções de perda como a Binomial ou Adaboost [34].

Para adaptar melhor o modelo a dados do mundo real é feita uma regularização a partir de uma penalidade. Também é introduzida uma aleatoriedade a cada iteração ao longo do treinamento, num processo similar ao *bagging* utilizado em florestas randômicas [36]. Diferentes amostras dos dados de treino são distribuídas para diferentes árvores de decisão que estão sendo treinadas com o objetivo de generalizar melhor seus resultados finais e diminuir o perigo do sobreajuste [36].

Apesar de ser um modelo bem complexo, seu uso com as configurações padrões já atinge resultados muito expressivos, sendo amplamente usado na classificação de conteúdo textual e com notoriedade por ser o campeão em muitas competições neste sentido [37].

Essas vantagens vêm com um custo: um grande consumo de memória e armazenamento durante seu processo de treino [34]. Para diminuir os gastos computacionais do *Gradient Boosting* e ao mesmo tempo entregar resultados mais acurados, Chen et al. [37] propôs um método chamado *XGBoost* ou (*Extreme Gradient Boosting*).

Ele diminui os custos de memória ao paralelizar e distribuir o treinamento dos modelos, superando o entrave de seu treinamento sequencial. Ao empregar gradientes de segunda ordem e um processo de regularização mais avançado, como a regressão *Ridge*, os resultados alcançados com esse modelo tem uma maior capacidade de generalização e por isso esta variação foi escolhida para ser estudada no trabalho.

### 2.4.5 *Transformers, BERT e BERTimbau*

Os modelos usados até então fazem parte de um conjunto de modelos tradicionais para a resolução do problema de classificação em linguagem natural. Na literatura mais recente, o modelo *BERT* [38] (*Bidirectional Encoder Representations from Transformers*) vem estabelecendo o novo estado da arte para a resolução desta [39] e de outras tarefas de *NLP*, como por exemplo a tradução e sumarização de textos, e por isso foi escolhido para o experimento.

A arquitetura deste modelo é baseada na implementação de redes neurais com um mecanismo de atenção, chamada *Transformer* [40]. Intuitivamente, este classificador procura extrair conhecimento não só a partir da frequência das palavras, mas também o contexto e significado entre elas. Isto é uma grande vantagem, pois enriquece a capacidade do modelo de classificação de reconhecer padrões e o torna mais próximo do processo humano.

Para exemplificar sua atuação: dada uma palavra que pode ter mais de um sentido, como “banco”, um modelo tradicional teria dificuldade de diferenciar se ela quer dizer o móvel para sentar ou a agência bancária. Já modelos baseados em *Transformers* conseguem captar o contexto em que a palavra “banco” aparece, para diferenciar seus significados.

O *BERT* aplica essa lógica de forma bidirecional ao analisar o texto, ou seja, da esquerda para direita e da direita para esquerda, para realizar uma análise de contexto que não dependa da ordem das palavras no texto. Além disso, é um modelo pré-treinado em toda a base do *Wikipedia*, com mais 2500 milhões de palavras, o que traz ao modelo um profundo aprendizado sobre como a própria linguagem humana funciona. Devido a essa fase de pré-treino, não é necessária uma base de dados muito grande na etapa de treino em si pois o aprendizado é transferido da etapa anterior. Após essa etapa de treino, há um processo de ajuste fino (*fine-tuning*) dos parâmetros, que adiciona algumas camadas na rede neural para que ela atinja o objetivo final, que no caso é o de classificação multi-classe.

Ainda nesse experimento, é utilizada uma variação do *BERT*, o *BERTimbau* [41], pré-treinado utilizando a base da *Wikipedia* em português e o *brWaC*, um corpus com 2.6 bilhões de *tokens* em português. Mesmo que o *BERT* tenha uma abordagem multi linguística, ou seja, que generalize seu aprendizado para vários idiomas, um modelo treinado especificamente na língua portuguesa pode trazer melhores resultados e por isso também foi incluído na análise comparativa.

## 2.5 Avaliação e Explicabilidade de Modelos

Os modelos devem ter uma confiabilidade alta na sua classificação e alta capacidade de generalização para garantir a aplicabilidade de seus resultados no mundo. Por isso são

Tabela 2.1: Exemplo de uma matriz de confusão para um problema de classificação binária.

Classe Real	Classe Preditada	
	Positiva	Negativa
Positiva	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Negativa	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Tabela 2.2: Exemplo de uma matriz de confusão para um problema de classificação multi-classe, quando a classe real e predita são iguais a B (Fonte: [42]).

Classe Real	Classe Preditada			
	A	B	C	D
A	VN	FP	VN	VN
B	FN	VP	FN	FN
C	VN	FP	VN	VN
D	VN	FP	VN	VN

usadas métricas e técnicas durante o treinamento para avaliar suas saídas e possibilitar o estudo comparativo dos resultados alcançados pelos modelos estudados.

### 2.5.1 Métricas de Avaliação dos Modelos

A partir do momento que uma classe é atribuída, a cada rótulo é possível montar a matriz de confusão dessa classificação para se avaliar a performance do modelo. Essa matriz, representada nas Tabelas 2.1 a 2.2, é o material base para o cálculo da maioria das métricas de avaliação preditiva [42]. Por convenção, suas colunas representam todas as classes preditas e suas linhas representam as classes verdadeiras das amostras.

Por consequência dessa organização, as amostras classificadas corretamente se encontram na diagonal principal, que vai do canto superior esquerdo até o canto inferior direito e são chamadas de Verdadeiros Positivos (VP), onde positivo significa o pertencimento a uma das classes  $Y = -1, 1$  e negativo o não-pertencimento. Tudo que está fora dessa diagonal principal é chamado de Verdadeiro Negativos (VN), Falso Positivo (FP) ou Falso Negativo (FN), de acordo como é indicado na Tabela 2.2.

#### Precisão e Revocação

A precisão é a métrica que mede, das classificações positivas, quais eram de fato verdadeiras e está demonstrada na Equação 2.15. Ela quantifica o quanto podemos confiar em modelo quando este prediz que uma amostra é positiva [42]. Já a métrica revocação calcula dentre todas amostras positivas, quais foram classificadas corretamente. Em outras palavras, ela nos indica a habilidade do modelo em encontrar todas as amostras positivas, Sua lógica está demonstrada na Equação 2.16.



$$\text{Precisão} = \frac{VP}{VP + FP} \quad (2.15)$$

$$\text{Revocação} = \frac{VP}{VP + FN} \quad (2.16)$$

## Acurácia

A acurácia é a métrica mais utilizada na prática para problemas de classificação [43] e sua grande vantagem é a facilidade de implementação e interpretação. Ela indica a performance de classificação do modelo, ao indicar a probabilidade da classe predita de uma amostra estar certa. Seu cálculo, demonstrado na Equação 2.17, é a razão da soma das classificações verdadeiras, os Verdadeiros Positivos e Verdadeiros Negativos, pelo total de classificações.

$$\text{Acurácia} = \frac{VP + VN}{\text{Total}} \quad (2.17)$$

Entretanto, é importante ter atenção nas características do problema de classificação ao se usar a acurácia. Quando existem múltiplas classes seu uso é pouco informativo, desfavorece classes com poucas amostras e favorece as com muitas amostras [43]. Uma técnica para remediar isso é fazer a soma ponderada das classes, onde os pesos significam a representatividade de amostras em uma determinada classe, de tal modo que quanto mais amostras, maior seu peso [42].

## F-1 Score

Uma outra métrica baseada na matriz de confusão, é *F-1 score*. Ela é formada pela média ponderada de outras duas métricas na tentativa de aproveitar as vantagens de cada uma delas. Mais especificamente é formada pela média harmônica da precisão e revocação, como demonstrado na fórmula Equação 2.18. Esta métrica tem um melhor poder discriminatório que a acurácia e por isso é mais indicada para problemas de classificação [43].

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \quad (2.18)$$

Para problemas multi-classe o cálculo dessa métrica envolve o cálculo de score para cada classe e por isso existem duas formas de se calculá-la: o macro *F-1 score* e *micro F-1 score* [42]. Intuitivamente, no macro a média é feita de forma que todas as classes têm o mesmo peso não importando seu número de representantes, ou seja, uma média aritmética simples. Já no micro F-1 score, o cálculo é essencialmente a própria acurácia, contando a

soma dos Verdadeiros Positivos de todas as classes e dividindo por todas as classificações e, conseqüentemente, pertence às mesmas vantagens e desvantagens da acurácia.

## 2.5.2 Explicabilidade

A maioria dos modelos apresentados neste estudo funcionam como caixas pretas, onde são necessários apenas provê-los de uma entrada de amostras de dados que o modelo retorna como saída um vetor de probabilidades ou um voto de maioria indicando a classificação escolhida. Avaliá-los por meio de métodos estatísticos é importante e necessário, mas ainda sim, é um processo que se mostra incompleto quando o domínio do problema a ser resolvido é complexo o suficiente [44].

Em problemas relacionados ao domínio médico, judicial ou qualquer outro que envolva um impacto direto na vida humana, modelos de aprendizagem de máquina, por mais complexos e generalizáveis que sejam, não consegue codificar todo o contexto e aprendizado suficiente para garantir seu uso no mundo real com confiança [44]. Estes modelos são treinados e otimizados para reconhecer padrões e prever classificações a partir de correlações dos dados. Os seres humanos, por outro lado, precisam descobrir relações causais a partir de seu julgamento próprio e aí sim, tomar uma decisão.

Como o uso de modelos estatísticos nestes domínios não é completamente aceito, a explicabilidade tem função de complementar seus resultados, dando explicações tanto do funcionamento do modelo em si, como de seu processo preditivo. O objetivo final é aumentar a confiança geral no uso destes modelos e trazer uma garantia maior na precisão de seus resultados quando aplicados no mundo real.

Os modelos de explicabilidade dispõem da habilidade de abrir esta caixa preta para entender quais foram os mecanismos internos do modelo que levaram a uma classificação e poder explicar, a partir dos atributos das amostras dadas, quais foram os que mais influenciaram na decisão de predição do modelo. Mais especificamente na classificação textual, a explicação da predição de uma amostra é dada se informando ao usuário quais foram as palavras que mais pesaram na predição de rótulo. Esse processo é ilustrado na Figura 2.6, onde é exemplificado o uso de um método de explicabilidade no contexto médico, mais especificamente, a construção da conclusão de um diagnóstico.

### Definição do problema e Tipos de Explicabilidade

Existem diversas formas de prover essa explicabilidade, mas antes é necessário entender que diferentes modelos exigem diferentes técnicas distintas. Alguns modelos são naturalmente explicáveis, como por exemplo as árvores de decisão e modelos lineares, definidos nas subseções 2.4.4, 2.4.2 e 2.4.3. Outros modelos, como os baseados em aprendizagem

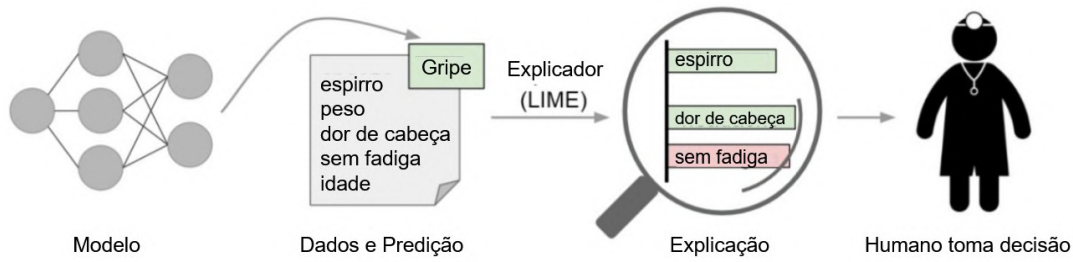


Figura 2.6: Processo de construção da explicação de uma predição para auxiliar uma decisão humana. Neste exemplo, o modelo tenta prever a doença de um paciente a partir de seus sintomas. Sua explicação é composta pelos principais sintomas que levaram a essa classificação e é informada ao médico, profissional que tomará a decisão final de diagnóstico (Fonte: [45]).

profunda, tem uma arquitetura que não é interpretável por si só e nesses casos é necessária a construção de um modelo de aproximação (ou *Surrogate Model*), que pode abordar o problema de forma global ou local. Os “globais” tentam gerar suas explicações com base em todos as amostras preditas enquanto os “locais” conseguem explicar apenas uma predição do modelo e por isso, suas explicações são válidas apenas localmente [44].

Formalmente, uma função explicável está na forma apresentada na Equação 2.19, que recebe um modelo de aprendizagem supervisionada e um conjunto de dados como entrada [44] e gera uma explicação  $e \in \mathcal{E}$ , onde  $\mathcal{E}$  é o conjunto de todas as explicações possíveis.

$$e : (\mathcal{X} \rightarrow Y) \times (\mathcal{X} \times Y) \rightarrow \mathcal{E} \quad (2.19)$$

Daí derivam os dois problemas de geração de explicação:

1. global: tenta extrair uma explicação global  $e(H, \mathcal{X})$  de um conjunto de dados  $\mathcal{X}$  e um classificador  $H$ .
2. local: tenta extrair uma explicação  $e(H, (X, y))$  para apenas uma amostra  $x$  e sua classe predita correspondente  $y$ .

Independente do processo, um explicador, ou *explainer*, deve sempre ser gerado com o objetivo de derivar explicações de um modelo ou predição de forma compreensível a qualquer ser humano. Estas explicações podem ser na forma de estatísticas dos atributos, grau de importância dos atributos ou até visualizações gráficas

## LIME

O LIME (*Local Interpretable Model-agnostic Explanations*) [45], é uma técnica de explicabilidade de modelos que, como seu nome indica, procura gerar explicações que sejam

interpretáveis ao seu contexto, portanto, uma representação entendível a humanos. Outra característica sua é ser agnóstico, ou seja, pode explicar qualquer classificador, ao sempre tratá-lo como uma caixa preta. A partir dessa premissa, esta técnica cria um modelo aproximado (ou *surrogate model*) ao classificador que seja localmente fiel a suas predições. Aqui é importante esclarecer que sua fidelidade local não implica necessariamente no entendimento global do classificador e só faz sentido nas proximidades da amostra selecionada durante o treino.

Formalmente, a explicação produzida pelo LIME é demonstrada na Equação 2.20, onde procura-se minimizar  $\mathcal{L}$ , função que mede o quão infiel o modelo aproximado  $g$  é do modelo caixa preta  $f$  na localidade definida pela função  $\pi_x$ . Para garantir que essa explicação pode ser minimamente interpretada por humanos, é adicionada uma função  $\Omega$  que mede a complexidade da explicação e que varia de modelo para modelo. Por exemplo, em uma árvore de decisão, essa complexidade pode ser denotada pela profundidade da árvore. Ribeiro et al. [45] complementa mostrando que para classificações textuais, a função  $\Omega$  deve ser limitada por um número máximo de palavras no vocabulário.

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (2.20)$$

Para alimentar a Equação 2.20, é construída uma base de dados  $\mathcal{Z}$  a partir de amostras  $z'$  aleatoriamente selecionadas ao redor de uma amostra central  $x'$  da base de dados original  $\mathcal{X}$ . Essas amostras são ponderadas por  $\pi_x(z)$  de forma que quanto mais próximas de  $x'$ , maior seu peso. Esta função de medição de proximidade garante uma confiança local do modelo  $g$ , ao mesmo tempo que é robusta ao ruído recorrente desta amostragem aleatória.

Todo esse processo está ilustrado na Figura 2.7 que mostra intuitivamente como funciona a escolha de uma amostra a ser explicada, sua localidade próxima e por fim a aproximação de uma função  $g$  em  $f$ .

No LIME, a função  $g$  pertence a classe de funções lineares esparsas,  $\mathcal{L}$  é uma função de erro quadrático com pesos locais, e  $\pi_x(z)$  é uma função de kernel exponencial que é definida a partir de um função de distância  $D$ , como a distância de cossenos no caso de classificação textual.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2 \quad (2.21)$$

$$\pi_x(z) = \exp\left(-D(x, z)^2 / \sigma^2\right) \quad (2.22)$$

Como explicado anteriormente, apesar de explicações locais serem importantes para se entender a predição de uma amostra elas não representam como o modelo se comporta

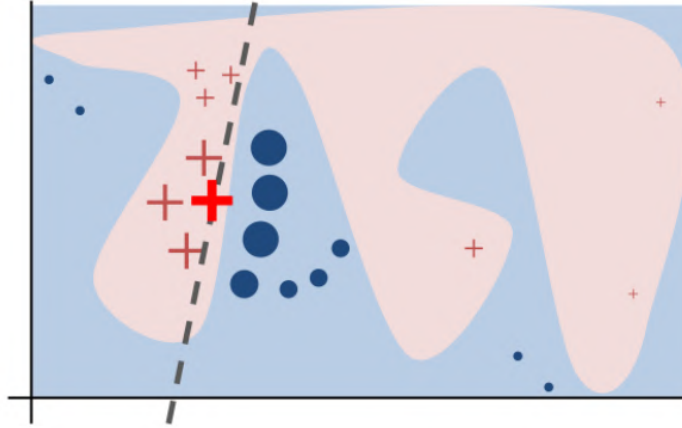


Figura 2.7: Exemplo da intuição presente no processo de explicação local do LIME, onde o preditor caixa preta  $f$  está representado nas áreas de vermelho e azul e o modelo aproximado calculado é a linha tracejada. A cruz vermelha em negrito representa a amostra escolhida para ser explicada, e as cruzes e círculos em volta, bem como seus tamanhos representam as instâncias aleatórias escolhidas e seus pesos por proximidade (Fonte: [45]).

globalmente. Com a proposta de também resolver esse problema, o LIME propõe uma técnica que complementa suas explicações locais e funciona da seguinte forma: ela escolhe uma diversidade de pares de predições de amostras e explicações que não sejam redundantes e na tentativa de englobar todo o contexto do modelo para explicá-lo globalmente [45], método chamado SP-LIME, de escolha sub-modular (ou *sub-modular pick*).

Para definir quais atributos têm relevância na explicação global do modelo, o método cria uma matriz  $\mathcal{W}_{ij}$  de explicações, onde as linhas  $i$  representam as amostras escolhidas e as colunas  $j$  representam os atributos da amostra. São atribuídos escores de importância para cada atributo a partir de uma função  $I_j$ . No contexto de classificações textuais,  $I_j$  tem o formato mostrado na Equação 2.23

$$I_j = \sqrt{\sum_{i=1}^n \mathcal{W}_{ij}} \quad (2.23)$$

A escolha (ou *pick*) das melhores amostras  $v \in V$ , é feita de forma a se evitar amostras com explicações parecidas, de tal sorte que todos os atributos sejam representados na seleção. Este problema está definido na Equação 2.8 e limitado por  $B$ , o número máximo de amostras a serem analisadas.  $c$  é a função que passa por toda a matriz  $\mathcal{W}$  e calcula as importâncias dos atributos (dado por  $I$ ) das amostras que aparecem ao menos uma vez em  $V$ . Por este problema ser implementado em um algoritmo guloso, é garantida uma aproximação da solução ótima em tempo polinomial [46].

$$\text{Pick}(\mathcal{W}, I) = \underset{V, |V| \leq B}{\text{argmax}} c(V, \mathcal{W}, I) \quad (2.24)$$

A Figura 2.8 representa de forma ilustrada o conjunto de amostras e atributos que formam a matriz  $\mathcal{W}$ , o processo de atribuição de importância das amostras e também o processo de escolha das amostras que melhor representam o modelo.

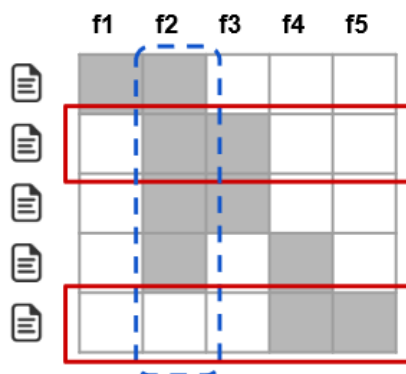


Figura 2.8: Exemplo do processo de escolha sub-modular entre várias amostras para explicar o modelo globalmente. As linhas representam amostras e as colunas representam os atributos dispostos na matriz  $\mathcal{W}$ . Neste exemplo o atributo f2 tem a maior importância e está tracejado em azul e as amostras selecionadas foram as marcadas em vermelho, pois representam o maior número de atributos (neste caso, todos exceto o atributo f1) (Fonte: [45]).

### Avaliação da Explicação

O processo de medir e avaliar a qualidade de uma explicação ainda é uma tarefa muito complexa principalmente devido ao componente da subjetividade na análise da explicação. De acordo com o estudo psicológico apresentado por Miller [47], uma “boa” explicação deve criar um contraste relevante em resposta a um evento contra-factual. Entretanto, sua avaliação é muito influenciada pelo contexto social em que é apresentada e sua seleção é sempre feita de forma enviesada e que por isso varia muito entre diferentes indivíduos. O autor exemplifica, mostrando casos onde apresentar causas únicas e simples, que não necessariamente são verdade, são mais confiáveis do que se referir a probabilidades e correlações estatísticas como as feitas nessa técnica.

Adadi et al. [48] complementa demonstrando que modelos de aprendizado de máquina são comumente tão complexos que, dois modelos idênticos treinados com o mesmo conjunto de dados duas vezes no mesmo modelo podem seguir diferentes caminhos de execução em seus algoritmos. Este fato, conseqüentemente, leva a geração de diferentes explicações em casos que ela deveria ser a mesma.

# Capítulo 3

## Revisão Bibliográfica

Neste capítulo, será sintetizado o conteúdo de alguns artigos que foram utilizados como referência neste projeto. Esse processo de pesquisa foi dividido em três partes, com a intenção de cobrir os principais trabalhos de pesquisa ligados aos assuntos que esta monografia aborda: o uso de textos políticos junto ao aprendizado de máquina, a classificação de viés político e uso do *Manifesto Project* e por fim, o uso da explicabilidade de inteligência artificial no contexto político.

### 3.1 Texto-come-Dados no Domínio Político

Grimer e Stewart [3] desenvolveram o trabalho que é a maior referência na automatização da análise de textos que pertencem ao contexto da ciência política. Nele, os autores organizam um guia muito completo que mostra quais os principais objetivos que podem ser explorados na área, dentre eles a descoberta de tópicos em comum entre os textos, a classificação entre categorias conhecidas ou até a localização de atores políticos em um espaço ideológico.

O artigo também apresenta: 1) os processos padrões de aquisição dos dados textuais, 2) as etapas de tratamento de texto como dados (nomeando *Text-as-Data*), que são necessárias para se reduzir a complexidade do texto, 3) e por fim a etapa de processamento, que pode ser feita utilizando modelo de classificação supervisionada, não supervisionada ou em métodos baseados em dicionários. Ainda, o trabalho explicita os cuidados que devem ser tomados na análise dos resultados apresentados por cada método e apontam suas principais falhas e pontos de atenção.

Entretanto, a principal contribuição do estudo se dá na ênfase que estes modelos e métodos estatísticos são uma aproximação imperfeita da linguagem humana e nunca substituirão uma análise detalhada feita por um pesquisador qualificado. Na realidade, estes têm o papel de assistir e amplificar as habilidades humanas.

Wilkerson e Casas [49] fazem uma coletânea de diversos trabalhos que abordam os diferentes objetivos que podem ser almejados na área, e em especial no objeto de estudo desta monografia. Os autores também denotam que, apesar do amplo uso de modelos não-supervisionados, estes podem trazer resultados muito instáveis e dificilmente replicáveis no mundo real. Em contraste, os pesquisadores descobriram que modelos supervisionados são muito mais robustos nesse sentido e por possuírem padrões-ouro para comparação, seu processo de validação se torna mais confiável.

## 3.2 Classificação de Viés Político e *Manifesto Project*

Existe uma variedade de trabalhos anteriores que utilizam manifestos eleitorais, discursos e debates transcritos para treinar modelos que tentam encaixar atores políticos dentro de um espectro ideológico ou viés político, sendo esta uma das primeiras aplicações da análise automatizada de textos na literatura [49]. Estes atores podem ser desde os próprios políticos e seus partidos, como a imprensa e suas notícias e reportagens.

Na literatura isso normalmente é feito por três técnicas principais. A primeira resume o viés político em apenas duas classes: “esquerda” e “direita”. Apesar de muito simples e não trazer muitos detalhes sobre os alinhamentos políticos do discurso, é suficiente para dar uma noção geral sobre o texto e facilita o treinamento dos modelos, por ser um rótulo binário. A segunda usa o partido político atribuído ao texto como rótulo de classificação, ou seja, é um rótulo *proxy* (ou de aproximação) que sugere uma posição no espectro político correspondente a praticada dentro do partido. Como a maioria dos textos (*i.e.* discursos e manifestos eleitorais), estão naturalmente atribuídos ao partido da pessoa que os proferiu, o processo de atribuição de rótulo a um texto é muito facilitado, e por isto é o método mais comum aplicado nos trabalhos. A terceira técnica tenta classificar os textos a partir de bases rotuladas manualmente por especialistas políticos. Estes rótulos normalmente englobam assuntos políticos mais detalhados, como “Ambientalismo”, “Economia” e “Seguridade Social”, onde a combinação deles dá mais detalhes sobre a posição política de um texto. Por esse processo ser manual, existem vieses intrínsecos a estas classificações e que podem afetar a predição final e por isso, apesar de seus rótulos serem mais informativos para uma análise detalhada, não apresentam resultados tão expressivos[49]. Exemplos de ambas técnicas aplicadas serão dados nos próximos parágrafos.

Molinari [50] constrói um modelo *Naive Bayes* para prever o alinhamento à direita ou esquerda de discursos políticos proferidos por deputados brasileiros e assim identificar a polaridade na Câmara. Esse alinhamento à direita ou à esquerda é inferido a partir da classificação do partido político. Ao treinar o modelo com discursos proferidos ao longo



de 16 anos, o estudo identifica as intensificações de polarização em momentos sensíveis da política brasileira, como durante o processo de *impeachment* da presidenta Dilma Rousseff. O estudo é um exemplo que, mesmo por meio de um classificador mais simples e que atinge resultados modestos (acurácia aproximada de 60%), ainda é possível extrair informações valiosas que podem auxiliar pesquisadores da área em suas análises. Seus resultados ainda são complementados por uma análise do vocabulário das classes, que explica melhor os resultados por classe do modelo.

Cavalcanti [51] faz um processo similar em uma base de dados de notícias oriundas dos portais dos partidos políticos PSDB, PMDB, PT, PSOL. Todavia, utiliza um modelo mais complexo de redes neurais, chamado de WiSARD, que originalmente era usado para reconhecimento de imagens, mas que mostrou resultados melhores no experimento do estudo, quando comparados aos dos modelos de *Naive Bayes*, SVM e *Gradient Boosting*. Outra vantagem do modelo proposto é a possibilidade de inclusão de novos elementos para um retreinamento e aprimoramento dos resultados, isso sem a necessidade do processamento de toda a base de dados.

Já Temporão et al. [52], tem um objetivo parecido ao de Molinari e Cavalcanti, mas utiliza um método baseado em dicionários. Ele utiliza um algoritmo chamado *Wordfish*, que constrói um dicionário dinâmico de termos que pertencem a uma ideologia e os utiliza para categorizar seus textos. Ele treina este modelo para textos pequenos, publicados em redes sociais como o *Twitter*, e constata que o uso de seu modelo é eficiente para identificar possíveis intenções de voto, quando esta análise é feita durante períodos eleitorais. Por utilizar uma abordagem diferente, sua avaliação é feita pela comparação dos parâmetros estimados e por isso não podem ser comparados os *F-1 scores* dos modelos utilizados no trabalho.

Biessmann [53] é mais ambicioso ao desenvolver um modelo para prever o viés político que funcione para qualquer texto genérico e para realizar esta tarefa seu artigo desenvolve uma combinação de várias técnicas e bases de dados. Utilizando um modelo de *Gradient Boosting*, o autor propõe três problemas de classificação a partir de duas bases em diferentes níveis de granularidade. A primeira base é composta por discursos transcritos do parlamento alemão onde procura-se classificar o alinhamento do texto: com partidos políticos do sistema político alemão e pela sua adesão às políticas do governo. A segunda é uma base chamada de *Manifesto Project*, composta por pequenas sentenças oriundas de manifestos políticos eleitorais e rotuladas manualmente por especialistas, sendo como objetivo classificar suas respectivas visões políticas por meio de tópicos gerais.

Apesar do estudo não encontrar resultados expressivos (*F-1 score* de 0.46) que permitam seu uso para classificação no mundo real, eles ainda são muito relevantes para ajudar especialistas em suas análises e possivelmente em modelos de aprendizagem ativa.

Além disso, ao analisar diferentes legislaturas no parlamento, o autor percebe que a acurácia do modelo para classificação de um partido pode mudar drasticamente ao longo do tempo e isto pode estar diretamente relacionado a mudanças de visões políticas internas ao partido. Complementarmente, foi realizada uma análise de correlação das palavras no texto com suas classes preditas e notou-se que determinadas palavras têm forte poder discriminatório entre as classes.

Zirn et al. [54] utiliza um modelo SVM e explora ainda mais o uso da base do *Manifesto Project* ao reduzir o problema de classificação de mais de 50 tópicos políticos para apenas 7 grupos de tópicos. Os autores também introduzem a ideia de se comparar a similaridade entre sentenças e verificar se a predição das sentenças adjacentes é igual à predição da sentença atual. Estes parâmetros são introduzidos ao modelo final para contribuir na classificação do manifesto como um todo. A junção destes métodos aumenta significativamente os resultados encontrados chegando a um *F-1 score* de 0.775.

Ao utilizar modelos baseados em redes neurais, como a CNN, LSTM e GRU, Reddy et al. [55] tenta emular o comportamento humano na leitura de notícias para detectar seu viés político. Os autores argumentam que, tanto os leitores, como jornalistas, tendem a prestar mais atenção nas manchetes das notícias e por isso, é adicionada ao modelo mais uma camada, chamada de camada de atenção (ou *attention layer*). Ela funciona de forma a adicionar um peso maior a partes específicas do texto da notícia, como por exemplo sua manchete. O trabalho então classifica o alinhamento de notícias de um jornal local a partidos políticos do sistema político indiano e encontra resultados impressionantes que chegam a alcançar um *F-1 score* de 0.89, quase 30% maiores que os resultados do *Naive Bayes* e SVM também presentes no experimento deste estudo.

### 3.3 Explicabilidade no Domínio Político

De acordo com Chatsiou e Mikhaylov [5], apesar de ser atraente a ideia de que algoritmos de aprendizado de máquina, quando aplicados à ciência política, podem tomar decisões ausentes de influência humana e sem nenhum viés, ela é uma ideia equivocada. Por ser treinado com dados produzidos e acompanhados por humanos, algoritmos tendem a perceber esses padrões de comportamento enviesados e conseqüentemente, perpetuá-los e reforçá-los, no que é chamado de viés algorítmico.

Portanto, já que não é possível simplesmente remover esse viés, uma análise clara dos resultados só é possível quando estes vieses estão esclarecidos e facilmente visíveis. Isso se faz ainda mais crítico em domínios sensíveis e que tem um impacto imensurável em vidas humanas, como o domínio legal [56] ou de saúde, e por isso cada vez mais a literatura começou a incorporar modelos de explicabilidade para complementam os resultados dos

modelos de classificação. Em um exemplo relevante e recente, a explicabilidade foi usada para aprender mais sobre a COVID-19 [57] [58].

Entretanto, o uso da explicabilidade de modelos no domínio político ainda não foi tão explorada [59]. A maior parte dos trabalhos nesta área se concentra na geração automática de explicações para um maior entendimento de modelos classificadores de notícias falsas. Um exemplo é a pesquisa de Kurasinski e Mihailescu [59], que treina dois modelos baseados em redes neurais, o LSTM bi-direcional e o BERT, com uma base de dados de mais de 9 milhões de amostras de notícias com rótulos “falsa” e “real”.

Assim como na técnica apresentada pelo supracitado Reddy et al., este tipo de modelo atribui em seu treinamento uma maior atenção a certas palavras mais que outras para contribuir para sua predição. A contribuição da pesquisa é utilizar este valor de atenção para criar uma demonstração visual que indica em vermelho, amarelo e verde palavras que influenciam negativamente, moderadamente e positivamente na classificação, respectivamente. Ademais, a intensidade da cor é utilizada para explicitar os níveis de atenção para cada palavra. Por meio desta técnica os autores constroem uma explicação interpretável de “como” os modelos distribuem sua atenção nas palavras do texto e que contribuem para o entendimento dos seus resultados.

A pesquisa de dos Santos [60] é outro exemplo de um trabalho que tenta extrair explicabilidade de redes neurais para a classificação de notícias falsas. Em complemento ao uso do modelo naturalmente explicável *Sequential S3*, também utiliza um método de modelo de aproximação chamado de LRP para gerar suas explicações para uma rede neural simples. O autor conclui que a análise das explicações geradas contribui para a identificação de diversos vieses nos modelos treinados.

# Capítulo 4

## Desenvolvimento

Com a finalidade de definir a melhor abordagem para classificação de viés político foi realizada uma análise comparativa entre diferentes modelos preditivos, variando sua complexidade e interpretabilidade. Para o treinamento destes modelos foi utilizada uma base de sentenças de manifestos políticos já categorizadas em dois rótulos que indicam sua visão política, e se diferenciam em seu nível de granularidade. Para aumentar a capacidade de generalização dos modelos e maximizar seus resultados, foi realizada a técnica de validação cruzada na fase de treinamento e a busca de hiperparâmetros ótimos em grade. Os resultados alcançados foram então avaliados a partir de uma métrica de desempenho estatístico, o  $F-1$  score e a partir de um método de explicabilidade, o LIME.

O desenvolvimento do experimento que será descrito a seguir está ilustrado na Figura 4.1.

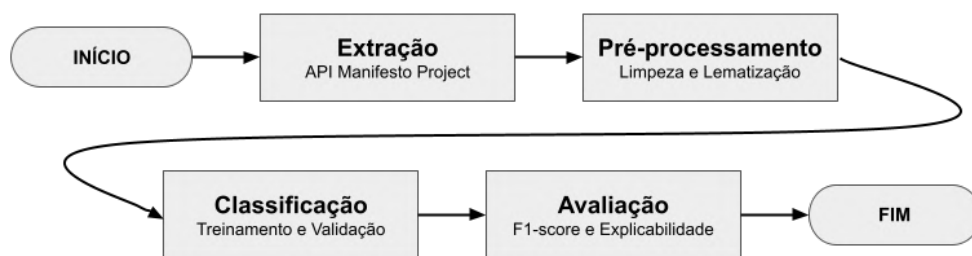


Figura 4.1: Fluxograma do desenvolvimento do projeto proposto na monografia.

## 4.1 Extração da Base de Dados

Por motivos legais, na maioria dos países democráticos, os discursos e debates proferidos em seus parlamentos são transcritos, digitalizados e disponibilizados livremente na *internet*. Como consequência, existe uma grande quantidade de dados, em diversas línguas e que representam uma variedade de posições e opiniões políticas.

A base selecionada para treinar os modelos foi uma coleção de manifestos políticos chamada *Manifesto Corpus*. A base faz parte do *Manifesto Project*, foi desenvolvida por Volkens et al. [4] e está disponibilizada gratuitamente em um banco de dados online dividida em duas partes, uma que engloba países pertencentes a OCDE (Organização para a Cooperação e Desenvolvimento Econômico) e outra exclusiva de países da América Latina. Os manifestos presentes na base representam os programas eleitorais de candidatos a eleições de mais de 50 países diferentes e em quase 40 línguas, compreendendo um total de mais de 4.500 documentos. Devido a limitação de escopo e objetivo do projeto foram selecionados apenas os manifestos em língua portuguesa, do Brasil e de Portugal, relativos apenas a eleições presidenciais que ocorreram nestes países.

A escolha desta base foi motivada justamente pelo grande volume de dados já categorizado em suas visões políticas, trabalho feito por especialistas qualificados na ciência política. Outro ponto é seu conteúdo textual que, por se tratar de manifestos eleitorais, está muito ligado a um contexto político e onde pressupõe-se uma comunicação clara dos vieses políticos de um determinado candidato ou partido.

Cada manifesto é analisado por um conjunto de especialistas que inicialmente dividem os manifestos em quasi-sentenças e após isto definem qual tópico ou assunto político ela pertence, a partir de um código numérico. As quasi-sentenças correspondem a uma unidade básica de texto, que contém exatamente uma afirmação ou frase. Este processo segue um manual de codificação que guia os cientistas políticos a seguirem os mesmos critérios de codificação, ao definir regras e conceitos comuns.

O código atribuído às quasi-sentenças pode variar entre 56 diferentes categorias de assuntos políticos. Algumas destas categorias ainda podem ser subdivididas em subcategorias, seja para assuntos mais específicos dentro de uma categoria ou ainda para definir juízos de valor a respeito de um tópico, por exemplo, os sinais “+” e “-” representam subcategorias que abordam o assunto de forma positiva ou negativa, respectivamente. A soma de todas as subdivisões de categorias totaliza 89 possibilidades de classificação diferentes para cada sentença, incluindo a possibilidade de não possuir código. Essa última classe chamada “Sem Código” se aplica a cabeçalhos, títulos curtos, e sentenças que não possuem significado dentro de um manifesto.

Os códigos dos assuntos políticos possuem uma característica hierárquica associada a eles. Todos são compostos por três algarismos, e se possuem alguma subcategoria, estas

são representadas por decimais desses números, ou seja, um número a mais após o ponto. As centenas, ou seja, o primeiro algarismo do código, representam um agrupamento de assuntos políticos, chamado de domínio político. Existem sete domínios políticos principais e as sentenças sem código são agrupadas no chamado “Sem Domínio”. Este agrupamento de categorias e subcategorias em domínios será muito útil na experimentação pois diminui o número de classificações possíveis no treinamento dos modelos, ao mesmo tempo em que ainda são diferentes entre si. Se essa diferença é suficientemente separável, isto pode fazer com que os modelos tenham uma probabilidade maior de encontrar bons resultados de classificação [61].

A Figura 4.2 representa todas as categorias e subcategorias dentro dos sete domínios políticos <sup>1</sup>. Por exemplo, a categoria com código **202.0 - “Democracia”**, possui quatro subcategorias: **202.1 - “Geral: +”**, **202.2 - “Geral: -”**, **202.3 - “Democracia Representativa: +”** e **202.4 - “Democracia Direta: +”**. Estes códigos estão todos agrupados dentro do domínio **2 - “Liberdade e Democracia”**.

Para a construção da base de dados final, inicialmente foi necessária a extração dos metadados dos manifestos, como por exemplo seu identificador único, o nome do candidato, nome do partido e data criação. Esses dados foram então utilizados para extração via API das sentenças e seus respectivos metadados para a construção de uma base de sentenças. Dos 26 manifestos brasileiros existentes, que abarcam todo o período após a redemocratização, ou seja, as eleições diretas desde 1989 até 2018, apenas 24 possuíam sentenças codificadas válidas. Dos 111 manifestos portugueses que vão de 1975 até 2019, apenas 28 continham sentenças válidas. A base final totaliza 92.041 sentenças codificadas entre 86 das 89 categorias possíveis, faltando amostras apenas para os rótulos **602 - “Estilo de Vida Nacionalista: Negativo”**, **305.4 - “Transição: Elites pré-democráticas: Positivo”** e **608.3 - “Multiculturalismo: Direitos Indígenas: Negativo”** (em tradução livre).

Todos estes dados estão estruturados segundo as variáveis descritas abaixo e uma amostra aleatória de exemplo dos dados está demonstrada na Figura 4.3, onde é possível observar as quasi-sentenças, códigos e identificadores dos manifestos.

- **manifest\_id**: a identificação de manifesto de origem.
- **text**: conteúdo textual das quasi-sentenças.
- **code**: o código que relaciona uma sentença a um dos 89 categorias e subcategorias de assuntos políticos.

---

<sup>1</sup>As descrições de cada domínio, categoria e subcategoria estão disponíveis na 5ª edição do Manual de Codificação de Manifestos, disponível no endereço eletrônico [https://manifesto-project.wzb.eu/download/papers/handbook\\_2014\\_version\\_5.pdf](https://manifesto-project.wzb.eu/download/papers/handbook_2014_version_5.pdf).

Table 1: Categories and Subcategories in Seven Policy Domains

<b>Domain 1: External Relations</b>	412 Controlled Economy: Positive
101 Foreign Special Relationships: Positive	413 Nationalisation: Positive
102 Foreign Special Relationships: Negative	414 Economic Orthodoxy: Positive
103 Anti-Imperialism: Positive	415 Marxist Analysis: Positive
103.1 State Centred Anti-Imperialism	416 Anti-Growth Economy: Positive
103.2 Foreign Financial Influence	416.1 Anti-Growth Economy: Positive
104 Military: Positive	416.2 Sustainability: Positive
105 Military: Negative	<b>Domain 5: Welfare and Quality of Life</b>
106 Peace: Positive	501 Environmental Protection: Positive
107 Internationalism: Positive	502 Culture: Positive
108 European/LA Integration: Positive	503 Equality: Positive
109 Internationalism: Negative	504 Welfare State Expansion
110 European/LA Integration: Negative	505 Welfare State Limitation
<b>Domain 2: Freedom and Democracy</b>	506 Education Expansion
201 Freedom and Human Rights: Positive	507 Education Limitation
201.1 Freedom	<b>Domain 6: Fabric of Society</b>
201.2 Human Rights	601 National Way of Life: Positive
202 Democracy	601.1 General
202.1 General: Positive	601.2 Immigration: Negative
202.2 General: Negative	602 National Way of Life: Negative
202.3 Representative Democracy: Positive	602.1 General
202.4 Direct Democracy: Positive	602.2 Immigration: Positive
203 Constitutionalism: Positive	603 Traditional Morality: Positive
204 Constitutionalism: Negative	604 Traditional Morality: Negative
<b>Domain 3: Political System</b>	605 Law and Order
301 Decentralisation: Positive	605.1 Law and Order: Positive
302 Centralisation: Positive	605.2 Law and Order: Negative
303 Governmental and Administrative Efficiency:	606 Civic Mindedness: Positive
Positive	606.1 General
304 Political Corruption: Negative	606.2 Bottom-Up Activism
305 Political Authority: Positive	607 Multiculturalism: Positive
305.1 Political Authority: Party Competence	607.1 General
305.2 Political Authority: Personal Competence	607.2 Immigrant Integration: Diversity
305.3 Political Authority: Strong government	607.3 Indigenous rights: Positive
305.4 Pre-Democratic Elites: Positive	608 Multiculturalism: Negative
305.5 Pre-Democratic Elites: Negative	608.1 General
305.6 Rehabilitation and Compensation	608.2 Immigrant Integration: Assimilation
<b>Domain 4: Economy</b>	608.3 Indigenous rights: Negative
401 Free-Market Economy: Positive	<b>Domain 7: Social Groups</b>
402 Incentives: Positive	701 Labour Groups: Positive
403 Market Regulation: Positive	702 Labour Groups: Negative
404 Economic Planning: Positive	703 Agriculture and Farmers
405 Corporatism: Positive	703.1 Agriculture and Farmers: Positive
406 Protectionism: Positive	703.2 Agriculture and Farmers: Negative
407 Protectionism: Negative	704 Middle Class and Professional Groups: Positive
408 Economic Goals	705 Minority Groups: Positive
409 Keynesian Demand Management: Positive	706 Non-Economic Demographic Groups: Positive
410 Economic Growth	
411 Technology and Infrastructure: Positive	000 No meaningful category applies

Figura 4.2: Lista com todas as categorias e subcategorias do *Manifesto Project*, dentro dos sete domínios políticos e seus respectivos códigos (Fonte: [4]).

- **domain\_code:** o código que relaciona uma sentença a um dos 8 domínios políticos.

	MANIFEST_ID	TEXT	CODE	DOMAIN_CODE
<b>36783</b>	180310_201010	construir presídios, de modo a reduzir o atual...	605.1	6
<b>77428</b>	35313_199910	consideramos, ao inves, que os dois unicos par...	305	3
<b>63978</b>	35220_200203	empenha-se na construcao de aliancas com outro...	107	1
<b>67680</b>	35311_200502	* agilizacao dos processos tributarios, os qua...	303	3
<b>14087</b>	180240_200210	relacao reservas cambiais/divida externa (0,17),	408	4

Figura 4.3: Amostra da base de sentenças codificadas.

Tabela 4.1: Exemplo de uma sentença original, sua versão limpa e sua versão lematizada.

Variável	Sentença
<i>text</i>	e o que abordaremos neste capítulo, reservando para o capítulo econômico as medidas de estímulo de competitividade e apoio à família.
<i>text_clean</i>	abordaremos capitulo reservando capitulo economico medidas estimulo competitividade apoio familia
<i>text_lemma</i>	abordar capitular reservar capitular economico medir estimular competitividade apoio familia

## 4.2 Pré-processamento dos Dados

Assim como descrito na seção 2.3.1, é preciso reduzir o tamanho das entradas com o fim de facilitar o treinamento do modelo e aumentar sua capacidade de generalizar seus resultados. Esse processamento todo é feito utilizando a linguagem *Python* em sua versão 3.8 e com o auxílio dos módulos *pandas*, para as realizar as operações com dados, *re* para a utilização de expressões regulares e *unidecode* para limpeza do texto.

Nesta etapa, primeiro é realizada uma padronização da base de sentenças, ou seja, os textos são todos convertidos para letras minúsculas. Também são removidos acentos, caracteres especiais, palavras de parada e qualquer palavra com menos de três caracteres. Ainda é feita a lematização, por meio de uma biblioteca chamada *spacy*, que possui um modelo lematizador pré-treinado em notícias e reportagens em português e que possui uma acurácia de 77% <sup>2</sup>.

No fim deste processo, são adicionadas duas novas colunas na base original, a **text\_clean** que representa apenas o texto limpo, e a **text\_lemma** que representa o texto limpo e lematizado. Com esse processo de limpeza, 753 sentenças ficam vazias e por isso são removidas. A Tabela 4.1 demonstra um exemplo de sentença, sua versão após a limpeza e sua versão após a lematização.

Também é feita uma análise estatística descritiva nas sentenças, para a identificação de sentenças com poucas palavras ou muitas palavras e assim remover quaisquer *outliers*. O percentil-99 destas sentenças é 31 e por isso são removidas 582 sentenças com tamanho maior que esse valor.

Além da limpeza do texto das sentenças, para que um classificador performe bem ele precisa de amostras suficientes em cada uma das classes. A capacidade de generalização dos aprendizados destes modelos está diretamente ligada a ele ser exposto em sua fase de treinamento a vários exemplos de sentenças que falem sobre o mesmo assunto político. A consequência de uma quantidade baixa de amostras é uma baixa confiabilidade no

<sup>2</sup>O modelo é o *pt\_core\_news\_sm* e detalhes podem ser encontrados no endereço [https://spacy.io/models/pt#pt\\_core\\_news\\_sm](https://spacy.io/models/pt#pt_core_news_sm).



resultado destes modelos no mundo real, devido a alta chance de sobre-ajuste do seu treinamento, fenômeno mencionado na subseção 2.1.

Por isso, neste pré-processamento, 64 classes que possuíam menos de 1000 sentenças foram removidas. O número limite de sentenças foi decidido empiricamente a partir de vários testes com os modelos.

Outra remoção necessária foi a das sentenças atribuídas à classe “Sem Código”. Como os atributos dos modelos são justamente as palavras em cada rótulo, os modelos na sua fase de treinamento buscam palavras que separem bem as classes. Como as sentenças deste rótulo não seguem um padrão específico e são muito gerais, poderiam acabar confundindo palavras que estão nesta classe em outras, e conseqüentemente afetar negativamente a capacidade de classificação dos modelos.

Todas as transformações realizadas no pré-processamento resultaram numa redução do tamanho total da base original. No final, é formada a base de dados A, que possui como rótulo o código de categorias e subcategorias e que contém 63.919 sentenças válidas, categorizadas entre 21 códigos possíveis. Estas sentenças possuem um tamanho médio de 11 palavras, com desvio padrão igual a 6 e um vocabulário de 35.235 palavras diferentes para o *text\_clean* e 20.671 palavras para o *text\_lemma*. As palavras mais frequentes são “nacional”, “política”, “social”, “desenvolvimento” e “saúde”. A Tabela 4.2 mostra a distribuição de sentenças nas 21 categorias e subcategorias presentes na base A.

Além da base de dados A, foi criada uma outra base a parte, chamada de B e que possui o código do domínio como rótulo e 81.619 sentenças válidas classificadas dentre 7 domínios possíveis. Nesta base, acontece o mesmo pré-processamento citado para a base A, exceto pela etapa onde removem-se as categorias com menos de 1000 sentenças. Graças ao agrupamento das categorias em domínios, não é necessário removê-las pois todos os 7 domínios possuem um número de sentenças acima de 1000, assim como demonstrado na Tabela 4.3

A tabela com a distribuição de sentenças nos códigos da base original, ou seja, antes do pré-processamento foi omitida por possuir tamanho extenso e pode ser encontrada nas Tabelas A.1 e A.2 presentes no apêndice A.

Tabela 4.2: Distribuição de sentenças nas categorias e subcategorias da base A, ordenada de forma decrescente.

<b>Código</b>	<b>Descrição do código</b>	<b>Número de Sentenças</b>
504.0	Welfare State Expansion	7787
411.0	Technology and Infrastructure: +	7087
506.0	Education Expansion	5199
503.0	Equality: +	5015
303.0	Governmental and Administrative Efficiency	4644
501.0	Environmental Protection	3840
502.0	Culture: +	3013
202.1	Democracy General: +	2848
416.2	Sustainability: +	2770
701.0	Labour Groups: +	2694
410.0	Economic Growth: +	2539
402.0	Incentives: +	2273
605.1	Law and Order: +	2213
703.1	Agriculture and Farmers: +	1999
403.0	Market Regulation	1981
414.0	Economic Orthodoxy	1740
301.0	Decentralization	1726
401.0	Free Market Economy	1219
304.0	Political Corruption	1161
408.0	Economic Goals	1099
305.1	Political Authority: Party Competence	1072

Tabela 4.3: Distribuição de sentenças nas domínios da base B, ordenada de forma decrescente.

<b>Código do Domínio</b>	<b>Domínio</b>	<b>Número de Sentenças</b>
5	Welfare and Quality of Life	25709
4	Economy	24386
3	Political System	9718
7	Social Groups	6915
6	Fabric of Society	5527
2	Freedom and Democracy	4989
1	External Relations	4375

## 4.3 Treinamento e Validação Cruzada

Para a execução do experimento foi escolhida a plataforma *Google Colabs* por ela fornecer um ambiente isolável e reprodutível. Outra vantagem é ela disponibilizar de forma gratuita uma GPU, um tipo de processador especializado em processamento paralelo e ideal para rodar vários experimentos ao mesmo tempo.

Antes de todo o processo de treino, a base, já tratada e balanceada, é dividida em dois conjuntos de dados, um conjunto de treinamento, composto por 80% da base e um conjunto de teste, composto pelos 20% restantes. A seleção de amostras é feita de forma randomizada e estratificada, ou seja, onde as proporções de amostras nas classes possíveis seja aproximadamente preservada em ambos conjuntos [62].

O processo de treinamento do modelo é composto por algumas etapas e por isso é implementado por meio da funcionalidade Pipeline do conhecido módulo de aprendizagem de máquina, o *sklearn*. Cada modelo classificador roda o seguinte fluxo de treinamento:

1. o tokenizador `CountVectorizer`, separa os tokens dos textos e cria uma matriz esparsa da contagem desses tokens por amostra da base de treino.
2. o `TfidfTransformer`, aplica o *tf-idf* e normaliza essas frequências de tokens em valores de ponto flutuante para cada amostra para preparar a entrada do modelo.
3. o modelo classificador, é finalmente treinado pelo método da busca em grade.

Os passos 1. e 2. são necessários para traduzir o conteúdo textual das sentenças para uma linguagem que o modelo estatístico entenda, assim como explicado na seção 2.3.2. Os modelos classificadores citados no passo 3. que foram selecionados para o experimento são: o *Naive Bayes*, Regressão Logística, SVM, *Gradient Boosting*, também implementados pelo *sklearn*, nas classes `MultinomialNB`, `LogisticRegression`, `SGDClassifier`, `XGBClassifier`, respectivamente.

Em complemento, os modelos de aprendizagem profunda BERT e BERTimbau, foram implementados pelo módulo *simpletransformers*, que importa a implementação destes modelos, *bert-base-uncased* e *liaadsrl-pt\_bertimbau-base*, diretamente do repositório de modelos chamado *Hugging Faces*. A intuição e funcionamento de cada um é estudada na Seção 2.4. Este tipo de modelo específico não segue o mesmo fluxo de treinamento citado anteriormente e só necessita de uma entrada textual e das classes únicas possíveis.

Cada modelo será treinado com duas entradas diferentes: a **text\_clean** e a **text\_lemma**, para que se verifique o efeito da lematização do texto nos resultados de classificação. Experimentos foram realizados com as duas bases de dados desenvolvidas: a A, que traz mais detalhes em relação ao assunto político da sentença; e a B, que é mais geral, pois representa o domínio político da sentença. A combinação de todos os fatores

testados leva a realização de vários experimentos, de modo que o processo de treinamento descrito nesta seção foi realizado 4 vezes para cada modelo.

Para cada modelo, excluindo os baseados em *Transformers*, serão realizados diversos treinamentos internos com diferentes hiperparâmetros, em uma técnica conhecida como busca em grade, ou *grid search*. O objetivo é que a comparação entre modelos seja feita com a melhor versão de cada modelo classificador, ou seja, que cada um esteja configurado com a combinação de hiperparâmetros que tragam a maior métrica de avaliação. A métrica de avaliação da classificação escolhida foi o macro *F-1 score*, estudado na Seção 2.5.1. Esta técnica de busca em grade foi implementada com o auxílio da função *GridSearchCV* da biblioteca *scikit-learn*.

O algoritmo *GridSearchCV* também divide os dados de treino em duas partes: uma para fazer o treino de fato e a outra para fazer a validação do treino, com o objetivo de verificar se o modelo está melhorando ao longo do treinamento. Esta divisão dos dados é necessária para evitar que o algoritmo fique viciado nos dados de treino e para garantir que as métricas representam bem o resultado obtido pelo modelo. O método de divisão foi feito utilizando a técnica da validação cruzada, ou *cross-validation* [63], que foi configurada para dividir o conjunto de treinamento em cinco lotes, ou *folds*.

Inicialmente é feita uma análise combinatória de todos os hiperparâmetros escolhidos. Por exemplo, se existe um deles com 3 valores possíveis e outro com 2 valores, no total teríamos 6 combinações a serem treinadas. Para cada combinação, segue o seguinte processo:

1. um dos cinco lotes é escolhido para validação do modelo, enquanto o restante é utilizado para o treinamento do modelo.
2. após o treino, as previsões são feitas para os dados do lote de validação e é calculado o macro *F-1 score* destes resultados.
3. repete-se o passo 1) até que todos os lotes tenham sido utilizados como validação.

No final, é calculada a média dos cinco macro *F-1 scores* associados a uma combinação, e a combinação de hiperparâmetros com maior *F-1 score* médio é escolhida como o classificador vencedor. A Tabela 4.4 mostra quais foram os parâmetros experimentados em cada modelo.

No *Naive Bayes*, o hiper-parâmetro experimentado foi o *alpha*, também conhecido como taxa de crescimento. Este parâmetro também é usado no algoritmo de *Gradient Descent*, que em cada iteração varia o parâmetro *alpha*, de maneira decrescente, com um objetivo chegar ao menor ponto possível da função, chamada função de custo. Um dos pontos cruciais para uma boa regressão usando esse parâmetro é definir bons valores para *alpha*, para que a regressão não seja muito lenta ou chegue a divergir.

Tabela 4.4: Hiperparâmetros experimentados na busca em grade de cada modelo classificador.

Sistema	Fator	Níveis
Naive Bayes	alpha	[1, 0.1, 0.01, 0.001]
Regressão Logística	penalty C	['l1', 'l2'] [0.01, 0.1, 1, 10, 100]
SVM Linear	alpha penalty	[0.0001, 0.00001, 0.000001] ['l2', 'elasticnet']
CountVectorizer	max_df	[0.75, 0.85, 1.0]

Para a regressão logística, foram selecionados dois hiperparâmetros para a análise: a penalidade e C. O hiperparâmetro penalidade implementa uma técnica bem conhecida de regularização, comumente utilizada para corrigir o *over-fitting*. A diferença entre os parâmetros l1 e l2 está no termo de penalização utilizado por cada uma, pois os métodos de regressão utilizados por elas são diferentes. Para l1, a técnica de regressão utilizada é a *Lasso Regression* [64], que adiciona um valor absoluto de magnitude do coeficiente como termo de penalidade da função de perda. Para l2, a técnica de regressão utilizada é a *Ridge Regression* [65], que adiciona uma magnitude quadrática do coeficiente como termo de penalidade da função de perda.

O parâmetro C [66] é comumente utilizado em regressões para analisar se os dados foram classificados corretamente, adicionando uma penalidade para cada dado que é mal classificado. Além disso, a penalidade não é a mesma para todos os exemplos mal classificados, pois seu algoritmo define o valor da penalidade de acordo com a distância em relação ao limite de decisão. Se o C assume um valor baixo, a penalidade para dados mal classificados é baixa, de modo que o limite de decisão com uma grande margem acaba sendo escolhido. Se C é grande, busca-se minimizar o número de exemplos mal classificados em razão da alta penalidade, resultando em um limite de decisão com uma menor margem.

Para o *SVM Linear*, utilizamos os mesmos hiperparâmetros alpha e penalidade, já explicados acima, mas nesse caso os valores variam entre l2 e *elasticnet*. O *Elastic Net* [67] é utilizado em razão do fato da *Lasso Regression* ter a possibilidade de definir variáveis de seleção muito dependentes dos dados analisados, o que cria uma possível instabilidade em decorrência do uso desse algoritmo. Assim, combina linearmente as penalidades da *Lasso Regression* e da *Ridge Regression* para aproveitar o melhor de cada um.

Devido à maior complexidade do algoritmo e conseqüente maior tempo de treino, o modelo *Gradient Boosting* foi treinado com a configuração padrão. O único valor padrão de parâmetro alterado foi o *max\_depth* de 6 para 4, que controla o tamanho máximo de profundidade das árvores e foi alterado para um valor menor na expectativa de se diminuir

a chance de sobre-ajuste.

O procedimento de escolha de parâmetros e validação cruzada não é utilizado nos modelos BERT e BERTimbau por dois motivos principais. O primeiro é que devido a fase de pré-treino, o modelo já tem agregado a ele um contexto muito amplo e as pequenas variações da base de treino que seriam realizadas em uma validação cruzada na etapa de ajuste fino tem pouco impacto no resultado final. O segundo motivo é relacionado a um maior tempo gasto na fase de treino devido a complexidade do modelo, que acaba por inviabilizar a testagem de um número elevado de parâmetros. Por esse mesmo motivo os modelos baseados no BERT foram configurados para serem treinados apenas com duas épocas, ou seja, duas fases completas de treinamento, e com os parâmetros fixos: *learning\_rate* igual a  $10^{-5}$  e com o formato meia precisão de ponto flutuante, ou *fp16*, desligado.

Além dos hiperparâmetros dos modelos, também é testada na busca em grade variações no tokenizador. O parâmetro *max\_df* é testado em todos os modelos com os valores 0.75, 0.85 e 1.0, onde 1.0 representa a seleção de 100% do vocabulário, 0.85 representa 85% e assim por diante.

## 4.4 Avaliação e Explicabilidade

Os modelos devem ter uma confiabilidade alta para serem capazes de determinar com precisão a qual assunto político uma sentença pertence. Para comparar os diferentes modelos será utilizada a experimentação como técnica de avaliação e a métrica de desempenho macro *F-1 score*. Dessa forma, podemos avaliar com mais precisão o desempenho de cada modelo em relação à mesma carga de trabalho. Os resultados em si, obtidos na comparação dos modelos serão discutidos na próxima seção.

O macro *F-1 score* foi escolhido como métrica principal pois, mesmo após todo o pré-processamento realizado para ambas bases de dados, as classes continuam desbalanceadas em relação ao número de sentenças por classe. Por exemplo, para a base de dados A, o código **504.0 - “welfare: +”** possui mais de 7 vezes mais sentenças que o código **305.1 - “political authority: party competence”**. Uma das vantagens do macro *F-1 score* é justamente sua simplicidade, que não considera em seu cálculo essas diferenças de quantidade de amostras entre classes e indica um resultado por classe mais realista.

Devido aos diversos fatores que são variados entre os experimentos, como diferentes bases de dados, rótulos e configurações de hiperparâmetros, a análise comparativa é feita tanto entre os modelos como dentro dos modelos. O objetivo é entender quais são as melhores entradas e saídas do modelo, bem como o melhor modelo para classificar o viés ideológico de uma sentença.

Em complemento a esta análise, também serão geradas explicações locais e globais do modelo vencedor a fim de aprofundar o entendimento dos seus resultados e de seu processo de classificação. Assim como descrito na seção 2.5.2, a geração destas explicações foi feita a partir das funções *LimeTextExplainer* e *SubmodularPick*. Estas foram implementadas pelo módulo *lime*, construído e disponibilizado na internet por Ribeiro et al. [45].

Para alimentar o gerador de explicações do LIME, são passadas como parâmetro o texto da sentença e as probabilidades de predição em todas as classes relativas ao texto, retornadas pelo modelo já treinado. Apesar de ser agnóstico a modelos, o LIME espera que as probabilidades de predição estejam entre os valores 0 e 1 e por isso é necessário normalizar as saídas dos modelos e garantir este intervalo de valores. Isso é feito no experimento por meio de uma função exponencial normalizada, ou *softmax*.

Estas explicações indicarão os pesos que as palavras da sentença tem na classificação de uma classe, onde pesos positivos indicam uma influência positiva e pesos negativos indicam uma influência negativa na probabilidade de predição em uma classe  $x$ . Como o problema abordado é do tipo multi-classe, a análise segue o processo *one-versus-rest*, ou seja, para uma classe  $x$ , são comparados os pesos entre  $x$  e  $\neg x$ , assim como definido na Subseção 2.4.3.

A interpretação dos resultados pode ser realizada a partir da verificação dos pesos que as palavras tem na classificação de uma sentença, sejam eles positivos ou negativos. Se uma sentença tem uma probabilidade de predição 0.89 na classe  $x$  e a palavra “saúde” tem peso 0.32, podemos interpretar que remover essa palavra do texto, diminuiria essa probabilidade para 0.57. A mesma lógica invertida funciona para palavras com peso negativo, que influenciam na “não-classificação” da sentença na classe  $x$ .

A explicação global é gerada para cada classe do modelo a partir das 10 palavras que mais influenciam a decisão de classificação em uma determinada classe. São amostradas explicações de 100 sentenças por classe, que foram escolhidas aleatoriamente para compor esta explicação global, já que utilizar toda a base é um processo extremamente custoso computacionalmente.

É importante ressaltar que a análise de várias explicações não necessariamente implica como uma nova sentença será classificada [44]. Assim como explorado por Holzinger et al [68], é importante se diferenciar explicabilidade de causalidade, pois enquanto uma parte de um sistema, a outra parte de pessoas. Um julgador humano é ainda o único que tem o poder e conhecimento de ditar e medir a qualidade destas explicações e é isto que será realizado no Capítulo 5.

#### 4.4.1 Avaliação com dados externos a Base de Dados

Para analisar a aplicabilidade do modelo em um contexto real, o classificador ganhador foi aplicado a artigos publicados em portais de notícias para uma avaliação qualitativa de suas classificações. Apesar de não trazer os melhores resultados, foi selecionado o BERTimbau treinado com a base A, pois seu rótulo de classificação que compreende as categorias e subcategorias traz mais detalhes para a análise do texto como um todo. Os artigos foram extraídos dos portais de grandes partidos políticos brasileiros e a expectativa é que a identificação de viés político neste tipo de publicação seja mais fácil. Os artigos selecionados serão detalhados a seguir.

O primeiro texto analisado foi extraído do portal de notícias do Partido dos Trabalhadores (PT), escrito pela jornalista Dandara Maria Barbosa <sup>3</sup>. O partido é o segundo maior do Brasil, muito associado à centro-esquerda e está ligado à luta por direitos trabalhistas, combate à fome e questões sociais em geral. O artigo apresenta a seguinte manchete “Cotas Sim: A importância do acesso da população negra à universidade” e é um texto que aborda positivamente assuntos ligados a políticas de ações afirmativas e a igualdade no acesso à educação.

Já o segundo artigo foi extraído do portal de notícias do Movimento Democrático Brasileiro (MDB), escrito pelo ex-presidente Michel Temer <sup>4</sup>. Este partido é atualmente o maior do Brasil em número de filiados e é considerado um partido “guarda-chuvas”, por agrupar ideologias diversas, mas que em sua maioria estão ligadas à centro-direita ou direita. O artigo tem o título “Reforma trabalhista é injustamente atacada”, onde rebate críticas à reforma trabalhista realizada no governo de Temer e aborda os efeitos da reforma na economia e na vida dos trabalhadores, bem como sua legitimidade de acordo com a legislação.

Por último, ainda é analisado um último artigo, extraído do portal de notícias do Partido Social Liberal (PSL), sem autor definido <sup>5</sup>. O partido elegeu o atual presidente do Brasil e está ligado a posições de direita e extrema direita. O artigo, de manchete “PL de José Medeiros libera extração mineral em qualquer área do País em momentos de crise”, anuncia a tramitação de um projeto de lei que libera a extração mineral em qualquer área do país, inclusive em áreas hoje protegidas pelo código ambiental. A justificativa é suprir a deficiência de insumos e contribuir com um aumento no crescimento econômico do país.

---

<sup>3</sup>O artigo foi publicado no dia 20/04/2022 e seu conteúdo pode ser acessado na íntegra pelo site <https://pt.org.br/cotas-a-importancia-do-acesso-da-populacao-negra-a-universidade/>.

<sup>4</sup>O artigo foi publicado no dia 10/01/2022 e seu conteúdo pode ser acessado na íntegra pelo site <https://www.mdb.org.br/reforma-trabalhista-e-injustamente-atacada/>.

<sup>5</sup>O artigo foi publicado no dia 14/03/2022 e seu conteúdo pode ser acessado na íntegra pelo site [http://pl22.com.br/Noticias\\_Liberaais\\_2022/noticias\\_2022\\_0868.html](http://pl22.com.br/Noticias_Liberaais_2022/noticias_2022_0868.html).



A análise qualitativa será realizada no texto como um todo, ou seja, no conjunto de classificações relativas às sentenças que nele estão contidas. O processo será complementado pela geração de explicações locais de algumas sentenças para possibilitar o entendimento das classificações por uma pessoa não qualificada.

O código que realiza experimento completo se encontra disponibilizado publicamente no repositório [https://github.com/gcvasconcelos/unb-classification\\_political\\_bias](https://github.com/gcvasconcelos/unb-classification_political_bias), para garantir a transparência do método e a reprodutibilidade do mesmo.

# Capítulo 5

## Resultados

Neste capítulo, serão apresentados os resultados de cada modelo que foram coletados durante o experimento. Primeiro, será realizada uma análise da eficiência, comparando os melhores resultados de cada modelo para realizar a tarefa de classificação em ambos rótulos desenvolvidos. Em seguida, os resultados atingidos são avaliados em relação aos da literatura. Para complementar a análise dos resultados, são apresentadas explicações locais e globais relativas às classificações do modelo vencedor. Ainda o modelo treinado é aplicado a textos de artigos fora da base original e seus resultados são avaliados.

### 5.1 Comparação entre os modelos

Após a realização dos experimentos com as variações de bases de entrada, rótulos de classificação e classificadores, os resultados foram armazenados para sua devida análise comparativa. O critério de escolha do melhor modelo foi o macro *F-1 score*, porém métricas como o micro *F-1 score* e tempo de processamento também foram consideradas na análise. As Tabelas 5.1 a 5.2 analisadas a seguir contarão com todas essas informações de resultados e a análise foi dividida em duas partes: a dos modelos treinados com a base A e a dos modelos treinados com a base B.

Nas tabelas, a coluna de **macro *F-1 score*** representa o macro *F-1 score* médio e desvio padrão do conjunto de resultados gerado a partir da média aritmética dos cinco lotes do processo da validação cruzada. Como as bases são apenas uma amostra do mundo real, foi calculado um intervalo de confiança de 95% para esta métrica, que estão na coluna **IC 95%**. Esse intervalo de confiança tem o objetivo de complementar a análise dos resultados e possibilitar uma comparação entre modelos de forma mais realista, em relação a sua aplicabilidade em um contexto real. Como o micro *F-1 score* não foi o critério de seleção da validação cruzada, o valor demonstrado na coluna **micro *F-1 score*** representa apenas o do modelo com os melhores parâmetros. A coluna **Tempo de Treino** representa o

Tabela 5.1: Resultados dos modelos classificadores para a Base de Dados A, ou seja, o rótulo que representa o código das categorias e subcategorias.

Modelo	macro F1-score	IC 95%	micro F1-score	Tempo de Treino
Naive Bayes	$0.419 \pm 0.004$	[0.414 0.425]	0.500	1.2 s
Regressão Logística	$0.467 \pm 0.003$	[0.463 0.472]	0.560	11.7 s
SVM linear	$0.473 \pm 0.005$	[0.467 0.481]	0.560	5.3 s
XGBoost	$0.426 \pm 0.006$	[0.418 0.434]	0.480	5.8 min
BERT	0.456	-	0.542	22 min
BERTimbau	<b>0.525</b>	-	<b>0.590</b>	21 min

Tabela 5.2: Resultados dos modelos classificadores para a Base de Dados B, ou seja, o rótulo que representa o código do domínio.

Modelo	macro F1-score	IC 95%	micro F1-score	Tempo de Treino (em minutos)
Naive Bayes	$0.480 \pm 0.006$	[0.473 0.488]	0.500	1.2 s
Regressão Logística	$0.537 \pm 0.006$	[0.529 0.546]	0.560	6.1 s
SVM linear	$0.555 \pm 0.006$	[0.547 0.564]	0.560	3.4 s
XGBoost	$0.437 \pm 0.003$	[0.432 0.442]	0.480	2.5 min
BERT	0.571	-	0.631	29 min
BERTimbau	<b>0.610</b>	-	<b>0.662</b>	28 min

tempo médio do processo de treinamento de todas as combinações de hiperparâmetros, dentre os cinco lotes de treinamento. No caso dos modelos BERT e BERTimbau, esta coluna representa o tempo total do treinamento de uma época.

Em todos os experimentos, nota-se que os modelos baseados no BERT não possuem desvio padrão nem intervalo de confiança calculados, pois não foi realizada a validação cruzada neste tipo de modelo e ele possui apenas um valor final.

As Figuras 5.1 a 5.2 representam graficamente a comparação entre métricas de desempenho de cada modelo, já considerando o intervalo de confiança desta métrica.

### 5.1.1 Modelos treinados com a Base de Dados A

Devido ao processo de busca em grade, cada modelo possui uma combinação de hiperparâmetros que possui o melhor resultado. Para o *Naive Bayes*, o parâmetro *alpha* de valor 0.01 foi o vencedor, por ser o valor que mais aproxima a função de custo do seu valor mínimo. Com relação a Regressão Linear, os melhores parâmetros para a classificação foram o l2, para a penalidade, e 1, para o parâmetro C, combinação que adiciona uma penalidade maior para palavras que classificam mal o texto ao mesmo tempo que tenta minimizar o sobre-ajuste com uma regularização média. Para o SVM Linear, os melhores parâmetros para a classificação foram alpha, com o valor de 0.00001, e para a penalidade,

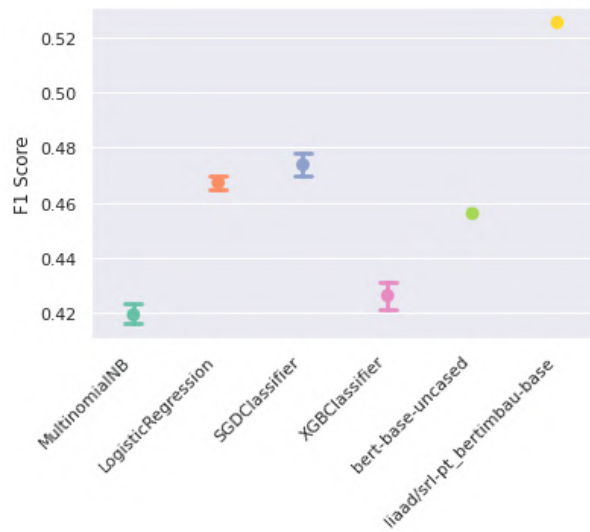


Figura 5.1: Resultados da classificação de categorias e subcategorias na base A por modelo, onde o ponto representa o  $F-1$  score médio e o traço que o corta representa seu intervalo de confiança.

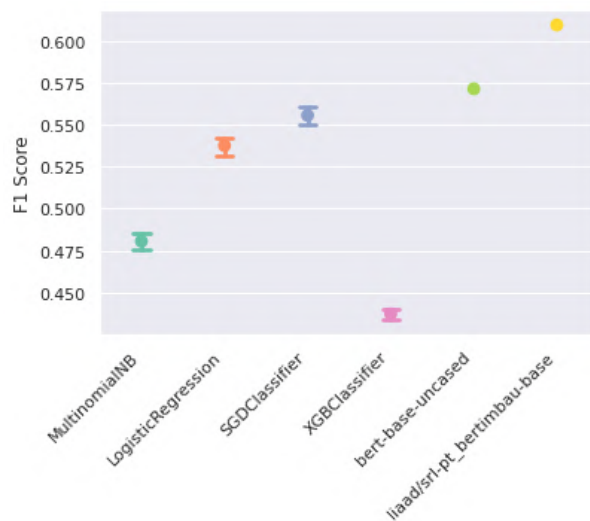


Figura 5.2: Resultados da classificação de domínios na base B por modelo, onde o ponto representa o  $F-1$  score médio e o traço que o corta representa seu intervalo de confiança.

a melhor foi a *elasticnet*. Como o *Elastic Net* faz uma combinação das regressões de l1 e l2, acaba tendo um bom desempenho para modelos que não estão em um extremos de *overfit* e *underfit*. A mesma análise do parâmetro  $\alpha$  que foi feita para o modelo *Naive Bayes* pode ser realizada aqui. Os modelos *XGBoost*, BERT e BERTimbau foram treinados apenas com uma combinação de parâmetros.

Como é possível verificar ao analisar os dados da Tabela 5.1 e Figura 5.1, vemos que

o modelo que teve a melhor performance para a Base de Dados A foi BERTimbau, com, macro  $F-1$  score de 0.525. Como estudado no Capítulo 2 é verificado na literatura recente, discutida no Capítulo 3, este resultado confirma a hipótese de que o uso de modelos de aprendizagem profunda baseados em *Transformers*, quando comparado a modelos tradicionais, apresentam os melhores resultados na tarefa de classificação do viés ideológico de textos.

É importante notar, que este fenômeno só aconteceu de fato para o modelo pré-treinado especificamente com textos em português, o BERTimbau, enquanto que o BERT demonstrou resultados piores que alguns modelos tradicionais, como a Regressão Logística e o SVM linear. Isto também demonstra que a qualidade e especialização da base pode trazer melhores resultados que a volume de palavras na fase de pré-treinamento em modelos de aprendizagem profunda.

Dentre os modelos tradicionais, o que performou melhor foi o SVM linear e aqui pode-se verificar a importância de continuar incluindo estes modelos mais simples nos experimentos de classificação de texto. O tempo de treinamento destes modelos são muito inferiores, quando comparados aos modelos de aprendizagem profunda e trazem resultados próximos, de 3 a 5 pontos percentuais de diferença.

Outro fato interessante encontrado neste experimento foi a baixa performance do *XG-Boost*. Apesar de ser um modelo que geralmente traz excelentes resultados em competições de classificação, mesmo em sua configuração padrão, e ainda possuir o maior tempo de processamento entre os modelos tradicionais, é o que traz um dos piores resultados. Como este tipo de modelo é composto por árvores de decisão baseadas em regras, provavelmente o alto número de classes dentro do rótulo (21 categorias e subcategorias distintas) aliado ao fato da base possuir poucas amostras por classe, o modelo parece ter dificuldades de criar regras boas o suficiente para classificar bem as sentenças, quando comparado aos outros.

Olhando para o micro  $F-1$  score de todos os modelos, vemos que os valores são significativamente maiores. Isto indica que os scores nas classes com mais sentenças são maiores do que as classes com menos sentenças, e isso leva esse número para cima. Em outras palavras, o modelo tem uma melhor performance em uma classe quando é exposto a mais amostras de sentenças desta classe em seu treinamento. Isto corrobora a hipótese de que a base A não possui amostras suficientes na maioria de suas classes, fato que influencia negativamente o resultado de todos os modelos treinados com a base A, e caso possuísse, poderia ter resultados mais próximos aos dos indicados pelo micro  $F-1$  score.

A Tabela 5.3 demonstra os resultados de  $F-1$  score por classe, vindos da classificação do modelo BERTimbau e aplicados a base de testes. Estes scores foram calculados pelo método *one-versus-rest*, explicado na Subseção 2.4.3. A análise desta tabela contribui

Tabela 5.3: Resultados por classe do classificador BERTimbau, treinado com a base A.

Códigos	F1-score	Número de sentenças
506.0	0.73	1038
502.0	0.68	568
605.1	0.65	433
504.0	0.65	1570
501.0	0.61	752
703.1	0.61	402
503.0	0.58	1044
701.0	0.57	544
411.0	0.60	1434
202.1	0.52	610
414.0	0.52	363
301.0	0.51	353
304.0	0.49	226
303.0	0.45	878
416.2	0.43	545
402.0	0.42	460
403.0	0.40	391
410.0	0.39	509
401.0	0.34	233
305.1	0.30	220
408.0	0.07	211

para a validação da hipótese apresentada acima acerca do balanceamento das amostras. É possível verificar que os códigos **506.0 - “Education Expansion”**, **504.0 - “Welfare State Expansion”** e **503.0 - “Equality: +”** estão dentre os que possuem mais amostras e são os que tem os melhores resultados, enquanto os códigos **408.0 - “Economic Goals”** e **305.1 - “Political Authority: Party Competence”** que possuem menos amostras, são os que apresentam os piores resultados.

Outra conclusão interessante que o experimento mostrou é que a utilização da lematização nos textos de entrada trouxe pouquíssimos ganhos ( $< 1\%$  de aumento no *F-1 score*) em todos os modelos. Como converter os *tokens* em *lemmas* é um processo custoso computacionalmente e o ganho foi baixo, o experimento concluiu que a realização desse processo não vale o *tradeoff*, comentado na Subseção 2.3.1. Por este motivo, os resultados encontrados no experimento foram omitidos.

### 5.1.2 Modelos treinados com a Base de Dados B

Para não se repetir o processo de escolha de parâmetros em grade, os modelos foram configurados com os mesmos valores de hiperparâmetros dos modelos vencedores treinados anteriormente com a Base de Dados A.

Tabela 5.4: Resultados por classe do classificador BERTimbau, treinado com a base B.

Domínios	F1-score	Número de sentenças
5	0.73	5174
4	0.72	4870
1	0.66	899
7	0.55	1343
2	0.54	1022
3	0.54	1947
6	0.54	1069

Os resultados demonstrados na Tabela 5.2, contribuem ainda mais para confirmar a hipótese sobre o desbalanceamento das amostras. O agrupamento de categorias em domínios realizado na Base de Dados B, leva à redução do número total de classes distintas e, conseqüentemente, o aumento de amostras por classe. Isto tem efeitos positivos na performance do modelo, pois nesta base há um aumento de quase 9 pontos percentuais no macro *F-1 score* e uma maior proximidade do macro e micro *F-1 score*, indicando que o número de amostras não está mais afetando tanto o cálculo da acurácia. O modelo BERTimbau continua trazendo os melhores resultados, apresentando um macro *F-1 score* de 0.61.

A Tabela 5.4 demonstra o *F-1 score* por domínio do classificador BERTimbau aplicados ao segmento de testes da base B. Apesar de reduzido, ainda sim verifica-se o problema de desbalanceamento que causa uma performance baixa do modelo nas classes com menos amostras.

O aumento de desempenho em relação a base A não é tão vertiginoso pois agregar as categorias e subcategorias também é uma decisão que tem um *tradeoff*. Apesar de seus efeitos positivos no resultado final, também causa um efeito negativo, que é diminuir a separabilidade das classes. Em outras palavras, as classes agrupadas em domínios passam a ser mais genéricas, pois englobam muitas categorias que até estão relacionadas entre si, mas podem possuir vocabulários muito diferentes. Isto tem o efeito de criar uma nova dificuldade para criar bons classificadores, pois como os modelos são alimentados por frequências de palavras nas sentenças e nas classes, estes não podem ser genéricos demais e acabar por confundir as classes e prejudicar suas predições.

Por exemplo, o domínio **6 - “Fabric of Society”**, agrupa sentenças das categorias **605.1 - “Law and Order: +”** e **601.1 - “National Way of Life General: +”**, que apesar de serem categorias relacionados ao comportamento de relações da sociedade, possuem um vocabulário e palavras chaves distintas em suas sentenças. Um ponto que ajuda a confirmar esta hipótese a análise das Tabelas 5.3 a 5.4, onde pode-se verificar que o desempenho dos classificadores para este código de domínio 6 é um dos piores (0.54) na

base B, enquanto o desempenho da classificação específica da categoria 605.1 é uma das melhores (0.65) na base A.

## 5.2 Comparação com a Literatura

Quando as métricas de desempenho encontradas neste experimento são comparados com resultados de outros encontrados na literatura relacionada ao problema da classificação de viés político, observa-se que eles estão bem distantes do estado da arte, que chegam a encontrar micro *F-1 scores* de 0.89, como comentado na Subseção 3.2.

Entretanto, a maioria dos experimentos que atingem estes resultados acabam por utilizar classes muito simples: seja reduzindo toda uma gama de visões políticas em “esquerda” e “direita”, seja utilizando classes *proxy* (ou aproximadas), como a classificação de pertencimento a partidos políticos. Estas técnicas acabam tendo uma aplicabilidade muito restrita a sua região e linguagem e são muito sensíveis a variações no tempo, já que a composição política dentro do partido pode se alterar a cada nova legislatura.

Este estudo utiliza rótulos mais complexos e descritivos e que, conseqüentemente, levam a um processo de aprendizagem mais difícil. Para permitir uma comparação mais realista, os resultados encontrados nesta monografia são comparados a trabalhos que usam rótulos similares em sua complexidade, mais especificamente a trabalhos que também treinam seus modelos a partir das quasi-sentenças do *Manifesto Project*.

A remoção de classes com poucas sentenças e de sentenças sem código na base A, aliado ao uso de modelos mais complexos se mostrou uma técnica eficiente para trazer um aumento de performance do modelo, quando comparado ao modelo de Biessmann [53], que atingiu um macro *F-1 scores* de 0.46 usando manifestos do parlamento alemão.

Já os resultados da base B, que agruparam os códigos em domínios assim como no trabalho de Zirn et al. [54] não se aproximou dos resultados de macro *F-1 score* igual a 0.775, encontrado em seu experimento. A abordagem que analisa a predição das sentenças adjacentes para complementar a classificação final utilizada neste trabalho se mostrou muito eficiente para trazer um score maior, mas até os resultados do modelo SVM utilizado como referência encontrou melhores resultados (0.728) que os apresentados nesta monografia.

De todo modo, quando comparamos os resultados com a classificação de uma classe escolhida pelo acaso, ou seja,  $\frac{1}{21} = 0.047$  na base A e  $\frac{1}{7} = 0.142$  na base B, vemos que os resultados são significativamente melhores do que essa linha de referência. Por isso, apesar de não serem confiáveis o suficiente para uma aplicação direta, sem assistência humana, o modelo desenvolvido neste trabalho pode ser utilizado para auxiliar especialistas a codificarem com mais facilidade novas sentenças no processo de codificação manual. Por



receber como entrada textos em geral, o uso do modelo não precisa se limitar a manifestos e sua performance pode ser avaliada em outros contextos, como em discursos políticos, notícias ou até em textos de redes sociais, como o Twitter [50][55][52].

## 5.3 Análise das Explicações Locais e Globais

Foram geradas explicações locais e globais, a partir do método LIME e sp-LIME, para complementar a análise dos motivos que um modelo performou melhor ou pior para determinada sentença ou classe de sentenças. Estas explicações foram analisadas para entender porque os resultados encontrados pelo melhor modelo do experimento não foram suficientes para superar os da literatura analisada.

As sentenças escolhidas para a geração de explicações locais foram selecionadas com base na análise da matriz de confusão gerada pelo melhor modelo das duas bases de dados, e que se encontra nas Figura 5.3 e Figura 5.14, processo explicado na Subseção 2.5.1.

As Figuras 5.4 a 5.19 mencionadas nas análises a seguir representam as explicações locais e globais da classificação do modelo para um determinado texto, onde as locais são compostas por três partes. A primeira demonstra as 5 maiores probabilidades de predição (à esquerda) em um gráfico em barras, que indicam a confiança do modelo na predição em uma classe. A segunda parte é composta por um ou mais gráficos que indicam os pesos positivos e negativos que as 5 palavras mais importantes têm na classificação do modelo. Por último, temos a sentença completa, onde as palavras coloridas no texto indicam seu peso na classificação de uma classe de mesma cor, tal que quanto mais intensa a cor, maior seu peso em módulo. As explicações globais possuem apenas o gráfico em barras com os pesos das palavras principais.

### 5.3.1 Modelo Vencedor treinado com a Base de Dados A

Para a classificação de categorias e subcategorias, a informação que chama mais atenção na Figura 5.3 são os resultados ruins de predição do código **408.0 - “Economic Goals”**, que mais confunde seus resultados com sentenças de código real **410.0 - “Economic Growth”** e **411.0 - “Technology and Infrastructure: +”**. Além de pertencerem ao mesmo domínio, estas são classes que realmente possuem uma intersecção de significados, já que uma meta econômica pode incluir um plano de crescimento econômico de um país ou de investimento de infraestrutura.

As explicações demonstradas nas Figuras 5.4 a 5.5 representam duas sentenças selecionadas aleatoriamente que pertenciam ao código 408.0, mas foram classificados erroneamente como 408.0 e 411.0, respectivamente. Para a explicação da Figura 5.4, palavras como “crescimento” e “produtividade” tem um peso maior para a classe 410.0, enquanto

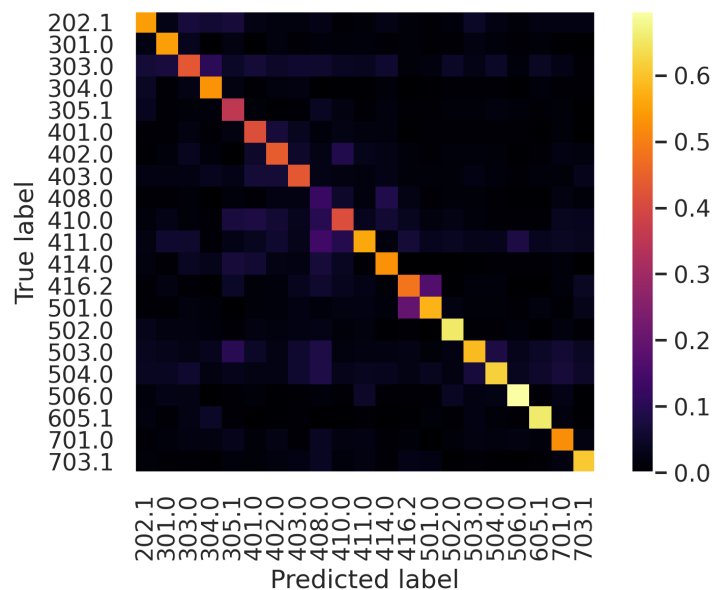


Figura 5.3: Representação gráfica da matriz de confusão do modelo vencedor treinado com a Base de Dados A, o BERTimbau. Quanto mais intensa a cor, seguindo a escala de cores a direita, mais amostras contém a célula que relaciona classe real nas suas linhas e classe predita nas suas colunas. Os valores foram normalizados em nível dos rótulos preditos (*i.e.* coluna a coluna) para que o desbalanceamento da distribuição de amostras por rótulo não afete intensidade da cor das células.

os verbos de ação “centrar” e “aumentar” tiveram pouca relevância para modelo, apesar de indicarem essa ideia de “Meta” para um avaliador humano. Já na explicação da Figura 5.5, o grande influenciador foi a palavra “tecnologias”, que por ter muito peso dentro do código 411.0 acaba enviesando o modelo a uma classificação errada.

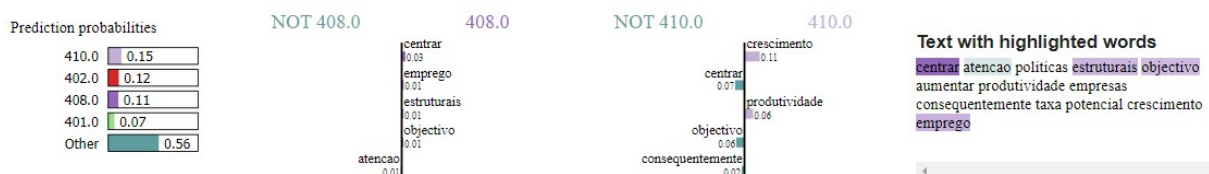


Figura 5.4: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original 408.0 - “Economic Goals” (em roxo escuro) e categoria predita pelo modelo 410.0 - “Economic Growth” (em roxo claro).

Um fator comum em ambos exemplos é falta de palavras com pesos fortes na classe verdadeira do modelo, aliado a uma baixíssima probabilidade de predição na classe predita, o que indica que os atributos usados para decidir a predição são fracos e o modelo chuta a classe “menos pior”. Isto também é notado na análise das explicações das sentenças que

foram preditas corretamente pelo modelo e um exemplo está ilustrado na Figura 5.6, que possui uma probabilidade de apenas 0.20.

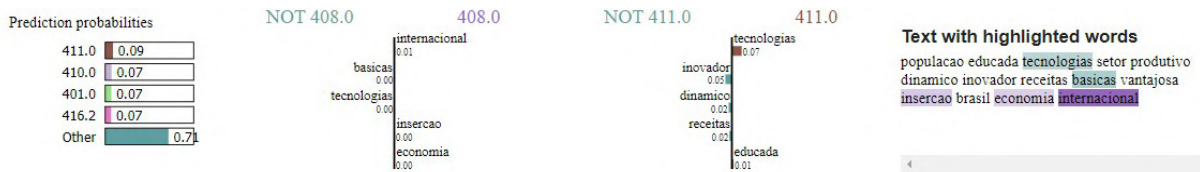


Figura 5.5: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original **408.0 - “Economic Goals”** (em roxo) e categoria predita pelo modelo **411.0 - “Technology and Infrastructure: +”** (em marrom).

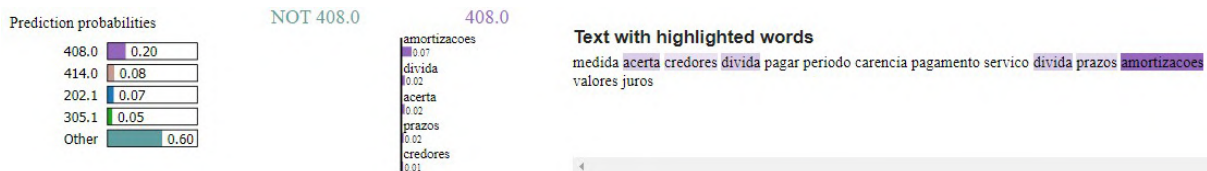


Figura 5.6: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original **408.0 - “Economic Goals”** (em roxo) e categoria predita pelo modelo **408.0**.

A explicação global das predições da classe **408.0 - “Economic Goals”** como um todo pode ser encontrada Figura 5.7. Os três principais grupos de atributos nos confirmam a hipótese da falta da capacidade do modelo de encontrar palavras relevantes para a classificação. A sua análise nos mostra que a maioria das palavras são muito comuns a outras classes e não seguem um padrão muito sólido. O contexto que uma sentença desta classe está dentro do manifesto parece ter um grande papel na decisão de classificação, pois é muito comum manifestos políticos a presença uma seção exclusiva para expor as metas econômicas de um projeto de governo. Este foi o fator que provavelmente teve um grande peso nos melhores resultados encontrados por Zirn et al. [54].

Voltando a Figura 5.3, percebe-se também que as categorias **416.2 - “Sustainability: +”** e **501.0 - “Environmental Protection: +”** se confundem frequentemente. Apesar de não pertencerem ao mesmo domínio, é possível que possuam uma intersecção de significados, já que políticas de proteção ambiental e o desenvolvimento econômico sustentável são temas que andam lado a lado e compartilham vocabulários parecidos.

Para verificar esta hipótese são analisadas as explicações de suas predições corretas, nas Figuras 5.8 a 5.9 e incorretas, nas Figuras 5.10 a 5.11, em cada classe. Nas classificações incorretas, nota-se que as sentenças possuem palavras com pesos grandes para ambas categorias e, conseqüentemente, há quase um empate entre as duas maiores probabilidades

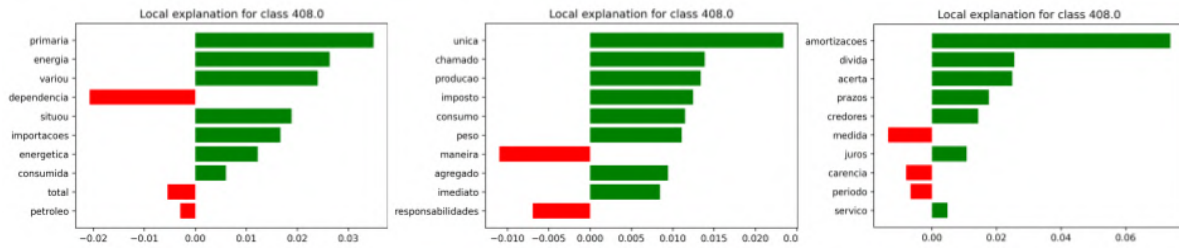


Figura 5.7: Explicação global referente a predição da categoria 408.0 - “Economic Goals”.

de predição final. É importante citar que até para um humano qualificado, distinguir entre estas classes seria um processo mais complicado.

Já nas classificações corretas, é evidente a presença de palavras diretamente ligadas a categoria, como “sustentabilidade” e “pegada ecológica” para a 416.2 - “Sustainability: +”, e “flora”, “fauna”, “natureza” e “protegida” para a 501.0 - “Environmental Protection: +”. Estas palavras levam a uma alta probabilidade de predição final para a classe correta, o que mostra que o modelo conseguiu aprender e extrair bons classificadores em sua fase de treinamento.

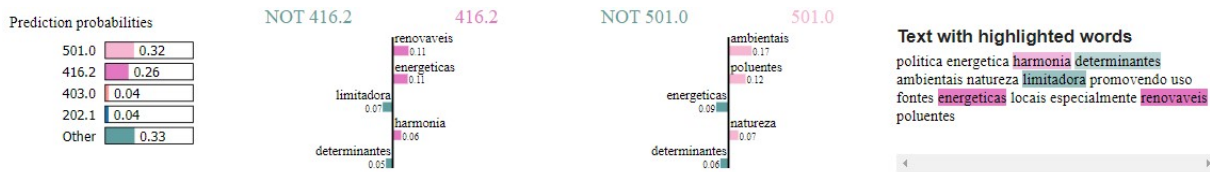


Figura 5.8: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original 416.2 - “Sustainability: +” (em rosa) e categoria predita pelo modelo 501.0 - “Environmental Protection: +” (em rosa claro).

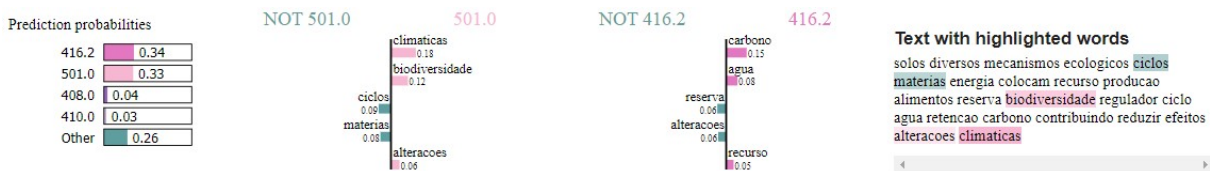


Figura 5.9: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original 501.0 - “Environmental Protection: +” (em rosa claro) e categoria predita pelo modelo 416.2 - “Sustainability: +” (em rosa).

Quando analisamos as explicações globais das duas classes, demonstradas nas Figuras 5.12 a 5.13, vemos que os grupos de atributos principais para cada classe de fato fazem



Figura 5.10: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original **416.2** - “Sustainability: +” (em rosa) e categoria predita pelo modelo 416.2.



Figura 5.11: Explicação local referente a uma sentença (caixa de texto à direita) com categoria original **501.0** - “Environmental Protection: +” (em rosa claro) e categoria predita pelo modelo 501.0.

sentido com a categoria em que estão. As palavras “sustentável”, “produção”, “energética” e até “carbono”, provavelmente se referenciando a créditos de carbono, fazem sentido como pesos positivos para uma segmentação econômica do tema sustentabilidade que o código 416.2 representa. No código 501.0, “especies”, “biomas” e “climáticas” se encaixam muito bem no tema de proteção ambiental, englobando a vida animal e vegetal.

A análise destas explicações contribui para entender como a separabilidade das classes também influencia no resultado final e que classes que agrupam sentenças de significados e vocabulário similares tendem a possuir resultados menores. É importante se levar em conta que o processo de rotulagem manual também não tem resultados 100% confiáveis e possuem erros e vieses humanos. Portanto, o modelo acaba importando estes vieses no seu processo de aprendizagem e os propaga em suas próprias previsões.

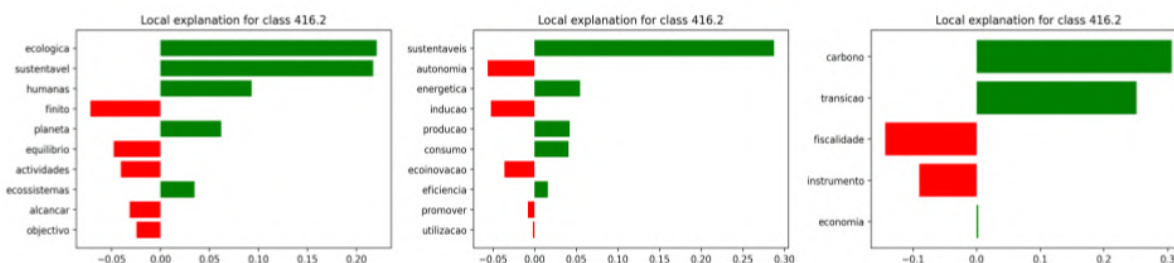


Figura 5.12: Explicação global referente a predição da categoria **416.2** - “Sustainability: +”.

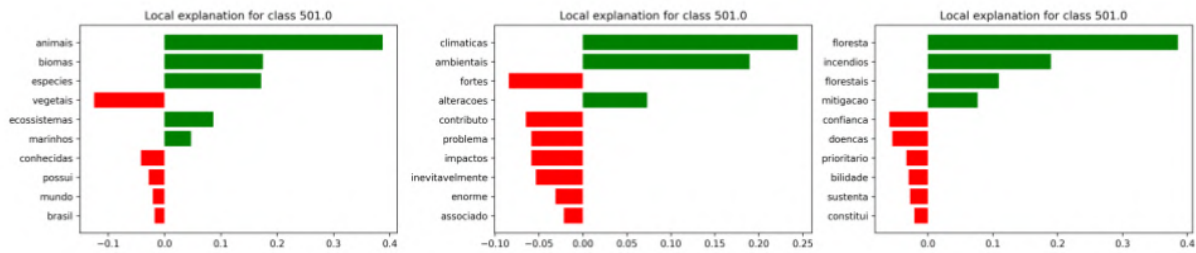


Figura 5.13: Explicação global referente a predição da categoria 501.0 - “Environmental Protection: +”.

### 5.3.2 Modelo Vencedor treinado com a Base de Dados B

Uma ressalva relacionada às explicações nesta Subseção é que nesta base de dados, por limitações técnicas, foi necessário codificar o rótulo original no treinamento do modelo. O código de todas as classes foi subtraído 1 unidade, ou seja, o domínio 5 é representado pelo código 4, o domínio 1 pelo código 0 e assim por diante. Anteriormente esses resultados eram decodificados pelo próprio modelo, mas como o LIME faz um modelo de aproximação para gerar suas explicações, perde-se essa informação da decodificação. A consequência disto é que na visualização das explicações a seguir, todas as classes estão ainda codificadas.

Para a classificação de domínios, a análise da matriz de confusão, presente na Figura 5.14, permite nos concluir que os domínios com maior acerto nas predições e maior número de amostras, 4 - “Economy” e 5 - “Welfare and Quality of Life”, são também os que se confundem entre si. Podemos notar também uma linha horizontal com um intensidade levemente maior para quando a classe verdadeira é 5 - “Welfare and Quality of Life”, indicando um índice maior de falso negativos para esta classe.

A Figura 5.15 representa a explicação de uma sentença originalmente rotulada como 5 - “Welfare and Quality of Life”, mas que o modelo classificou como 4 - “Economy”. A sentença contém palavras muito relacionadas à economia, como “ciência”, “tecnologia” e “pequisa” e que levam a uma alta probabilidade de predição no domínio 4. Apesar destas palavras também poderem fazer sentido no domínio 5, provavelmente não estavam tão presentes no vocabulário desta classe. Como o modelo se baseia muito na frequência das palavras e nos documentos, ao utilizar o *tf-idf*, estas palavras acabaram tendo um peso negativo associado a elas.

Na Figura 5.16 encontra-se a situação oposta, ou seja, uma sentença com classe verdadeira “4 - Economy”, mas classificada como 5 - “Welfare and Quality of Life”. Neste exemplo, a presença de palavras muito ligadas a políticas ambientais como “desma-

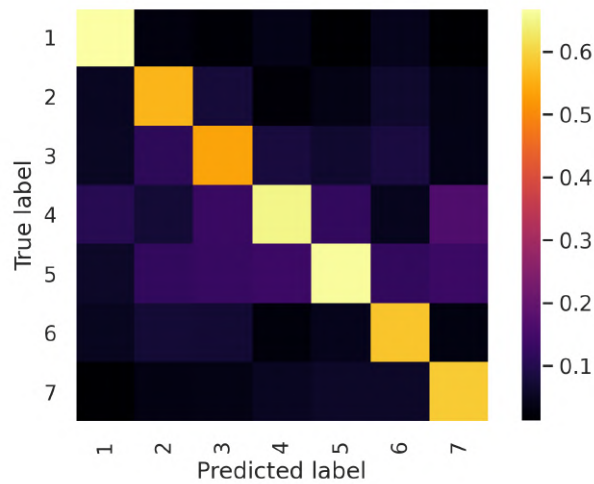


Figura 5.14: Representação gráfica da matriz de confusão do modelo vencedor treinado com a Base de Dados B, o BERTimbau. Quanto mais intensa a cor, seguindo a escala de cores a direita, mais amostras contém a célula que relaciona classe real nas suas linhas e classe predita nas suas colunas. Os valores foram normalizados assim como feito na Figura 5.3.

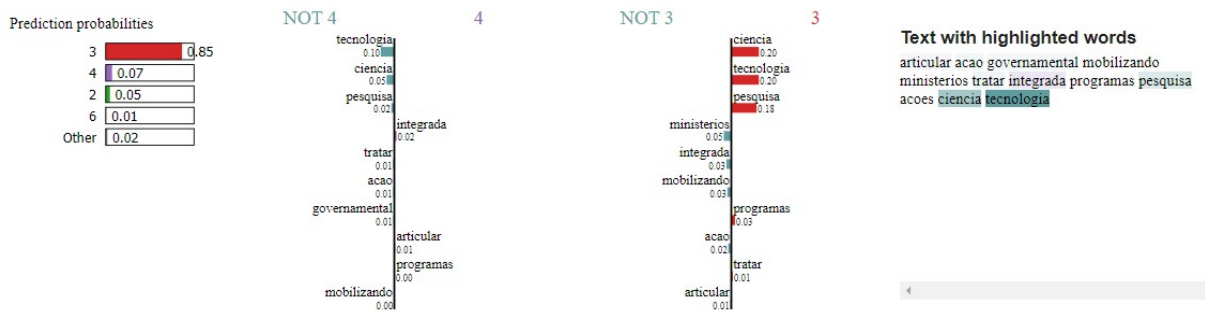


Figura 5.15: Explicação local referente a uma sentença (caixa de texto à direita) com domínio original **5** - “Welfare and Quality of Life” (em roxo e código 4) e domínio predito pelo modelo **4** - “Economy” (em vermelho e código 3).

tamento”, “conservação” e “biodiversidade” como uma política econômica em si e acaba fugindo do discurso usual onde o crescimento econômico usualmente está ligado a processos de destruição do meio ambiente e exploração de recursos. A hipótese é que o modelo parece de fato ter internalizado este viés do discurso hegemônico, que acaba por mencionar menos estas palavras no vocabulário do domínio 4 e por isso atribui pesos negativos a elas. Por estes motivos, o modelo acaba classificando a sentença erroneamente e com altíssima probabilidade de predição.

A análise da explicação global do domínio **5** - “Welfare and Quality of Life”, na Figura 5.17, é possível observar que as palavras “cultura”, “patrimônio”, “moradas po-



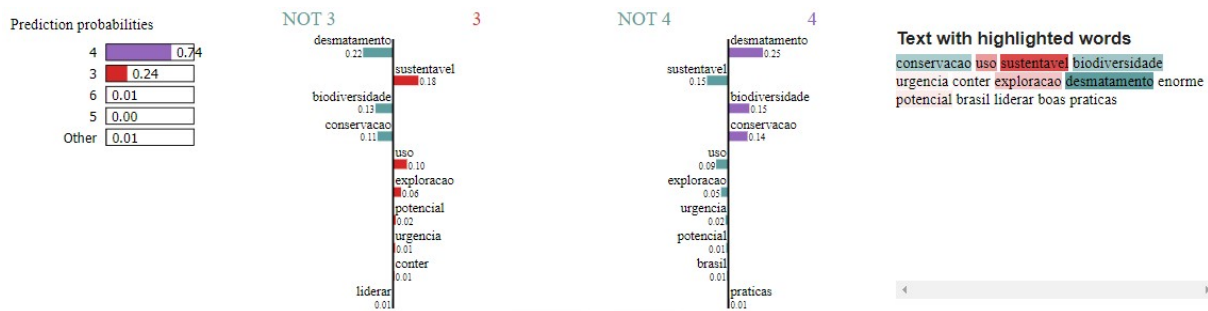


Figura 5.16: Explicação local referente a uma sentença (caixa de texto à direita) com domínio original 4 - “**Economy**” (em vermelho e código 3) e domínio previsto pelo modelo 5 - “**Welfare and Quality of Life**” (em roxo e código 4).

pulares” são ótimos pesos para predizer corretamente esta classe, enquanto “burocracia”, “setoriais” e “administrações” também são bons pesos para se diminuir a possibilidade de classificação nesta classe. Aqui nota-se também um exemplo de atributo ruim, a palavra “nordeste”. Por se referenciar a uma região geográfica específica do brasil e possuir alto peso negativo, diminui a capacidade do modelo de generalizar seus resultados em outros contextos. Por isto, idealmente este tipo de atributo de localidade deveria ser removido da base antes do treinamento.

A explicação global do domínio 4 - “**Economy**”, pode ser vista na Figura 5.18. As palavras que estão muito ligadas ao vocabulário econômico se mostram bons classificadores, como “rendimento”, “consumo” e “macroeconômico”. Nestas explicações não vemos confirmação da hipótese citada anteriormente, relacionada ao peso negativo em palavras relacionadas ao meio ambiente. Como estas explicações globais recebem apenas uma amostra de sentenças da classe, não é possível confirmar se a hipótese é válida ou não.

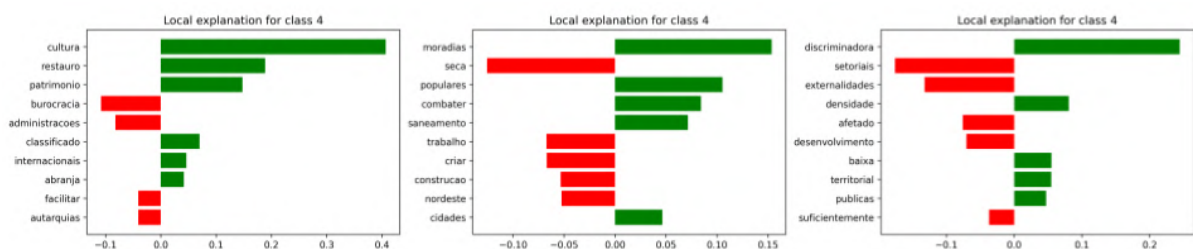


Figura 5.17: Explicação global referente a predição do domínio 5 - “**Welfare and Quality of Life**”.

Outro fenômeno interessante que pode ser observado na matriz de confusão da Figura 5.14 é a concentração de predições erradas no domínio 7 - “**Social Groups**”, quando a classe verdadeira era a 4 - “**Economy**”.



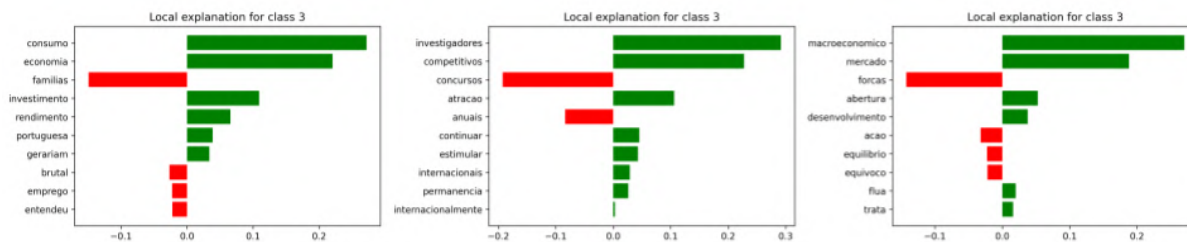


Figura 5.18: Explicação global referente a predição do domínio 4 - “Economy”.

As Figuras 5.19 a 5.20 demonstram duas explicações para quando acontece esse erro. A análise dos resultados das duas sentenças nos mostra palavras como “profissional”, “industrial”, “salário mínimo” que são muito ligadas a grupos sociais ligados à questão trabalhista. Entretanto sofre o mesmo fenômeno demonstrado na Figura 5.15, ou seja, como estas palavras são utilizadas mais frequentemente no vocabulário do domínio 4, acabam pesando a classificação para este domínio.

Isto também ajuda a exemplificar o problema do rótulo de domínios citado anteriormente na Subseção 5.1.2. Estas sentenças de exemplo originalmente pertenciam a categoria **701.0 - “Labour Groups”** e possuem palavras muito relevantes, frequentemente utilizadas em todas as sentenças deste código. Ao analisar o *F-1 score* desta categoria para os modelos da base A na Tabela 5.3, ele é relativamente alto, valendo 0.63.

Porém, quando as categorias 701.0, 703.1 (com *F-1 score* de 0.65) e outras com menos amostras como 706.0, 703.0, 705.0 ..., são agrupadas no domínio **7 - “Social Groups”**, a classe se torna generalista demais e o modelo tem mais dificuldade de encontrar atributos relevantes. As palavras que antes tinham grande peso na predição agora são misturadas com outras não necessariamente muito relevantes para a categoria original da sentença. Consequentemente, o domínio traz um *F-1 score* de 0.55, pior que suas categorias individuais, como indicado na Tabela 5.4.

Estas explicações demonstradas são apenas amostras selecionadas a mão por terem sido consideradas relevantes para o experimento. Contudo, explicações globais de todas as classes, podem ser encontradas nas Figuras B.22 a B.28 para os domínios e Figuras B.1 a B.21 para as categorias, todos localizados no Apêndice B.

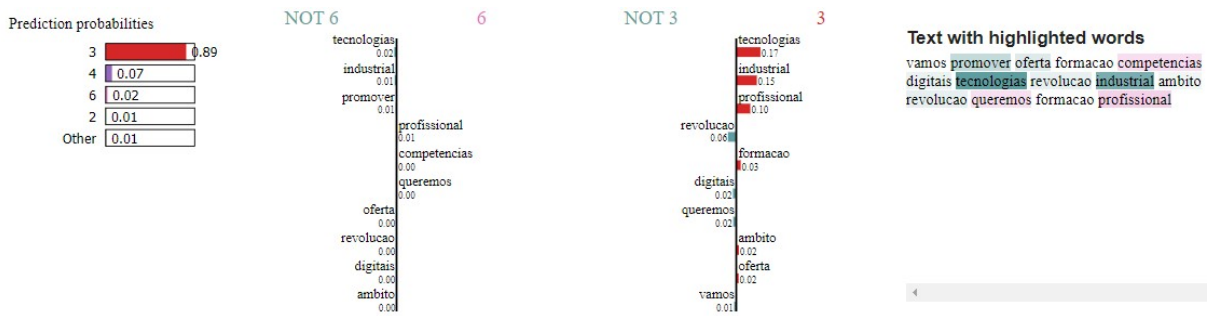


Figura 5.19: Explicação local referente a uma sentença (caixa de texto à direita) com domínio original 7 - “**Social Groups** (em rosa e código 6) e domínio predito pelo modelo 4 - “**Economy**” (em vermelho e código 3).



Figura 5.20: Explicação local referente a outra sentença (caixa de texto à direita) com domínio original 7 - “**Social Groups**” (em rosa e código 6) e domínio predito pelo modelo 4 - “**Economy**” (em vermelho e código 3).

## 5.4 Aplicação do modelo em Artigos Políticos

Para o artigo do PT, relacionado ao apoio à política de cotas e ao acesso à educação universal, o resultado do modelo BERTimbau pode ser encontrado na Tabela 5.5. O modelo teve um excelente resultado para esta amostra e como se esperava, a maioria das sentenças foram classificadas com códigos relacionados ao tema do texto e com alta probabilidade de predição, como o 503.0 - “**Equality: +**” e 506.0 - “**Education Expansion**”. Os temas com menor probabilidade média, como o 202.1 - “**Democracy General**” e 504.0 - “**Welfare State Expansion**”, também fazem sentido dentro do contexto do texto, ou seja, mesmo com uma confiança baixa o modelo identificou corretamente os vieses do texto.

A Figura 5.21 demonstra a explicação para a sentença com maior probabilidade de predição na classe 506.0. Nota-se um peso muito alto para a palavra “educação”, aliada de outras relevantes ao tema como “oportunidade”. Um peso tão grande da palavra “educação” é um alerta, pois pode indicar um sobreajuste do modelo nesta classe, visto que frases como “falta de educação” teria o mesmo peso, mas a frase indicaria um significado

Tabela 5.5: Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do portal do PT, com um viés positivo no apoio à política de cotas e ao acesso à educação universal.

Código	Categoria	Nºde sentenças	% de sentenças	Probabilidade Média
503.0	Equality: +	20	49%	0.853
506.0	Education Expansion	16	39%	0.762
504.0	Welfare State Expansion	2	5%	0.366
202.1	Democracy General: +	1	2%	0.279
303.0	Governmental and Administrative Efficiency	1	2%	0.185
701.0	Labour Groups: +	1	2%	0.666

oposto.



Figura 5.21: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo **506.0** - “**Education Expansion**” (em amarelo).

Outro fato interessante neste artigo é a presença de uma sentença com código **701.0** - “**Labour Groups**” que provavelmente está diretamente ligada ao viés do partido em si. A visualização da explicação relacionada a esta sentença, presente na Figura 5.22, mostra que palavras relacionados à questão trabalhista como “empregada”, “doméstica” e “trabalhar” pesam muito na classificação desta categoria.



Figura 5.22: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo **701.0** - “**Labour Groups**” (em azul claro).

A Tabela 5.6 demonstra os resultados do modelo para o texto do artigo do ex-presidente Temer, que aborda temas da reforma trabalhista, sua legitimidade e seus efeitos na economia e na sociedade. O modelo parece ter performado bem neste exemplo também. Apesar de apenas possuir uma categoria com alta confiança, a **701.0** - “**Labour Groups**”, as predições com baixa confiança também fazem sentido dentro do contexto do texto.

Tabela 5.6: Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do MDB, com um viés político ligado a posições da direita tradicional.

Código	Categoria	Nº de sentenças	% de sentenças	Probabilidade Média
701.0	Labour Groups: +	17	50%	0.817
305.1	Political Authority: Party Competence	5	15%	0.295
202.1	Democracy General: +	4	12%	0.302
504.0	Welfare State Expansion	3	9%	0.264
414.0	Economic Orthodoxy	2	6%	0.467
303.0	Governmental and Administrative Efficiency	1	3%	0.151
304.0	Political Corruption	1	3%	0.266
410.0	Economic Growth: +	1	3%	0.246

Além do tema principal envolvendo a questão trabalhista, temas como **414.0 - “Economic Orthodoxy”** e **304.0 - “Political Corruption”** estão dentro da retórica do partido associada a um discurso da direita política e, por isso, ajudam a validar os resultados do modelo. Em complemento, a presença dos códigos **305.1 - “Political Authority: Party Competence”** e **303.0 - “Governmental and Administrative Efficiency”** fazem muito sentido, já que o texto foi escrito pelo ex-presidente Temer, personalidade muito ligada à política tradicional e ao discurso de maximização da eficiência da máquina pública.

As Figuras 5.23 a 5.25 apresentam exemplos de explicações para algumas sentenças presentes no artigo e ajudam a entender quais foram os critérios usados pelo modelo. Uma inspeção geral mostra que as palavras com maior peso fazem sentido dentro de suas classes e foram fundamentais para um bom resultado do modelo, mesmo que com pesos mais baixos. Um ponto interessante na explicação da imagem Figura 5.24 é como o modelo pesa as palavras dentro de uma categoria muito específica da economia, a ortodoxa, e de fato levam a uma classificação correta.

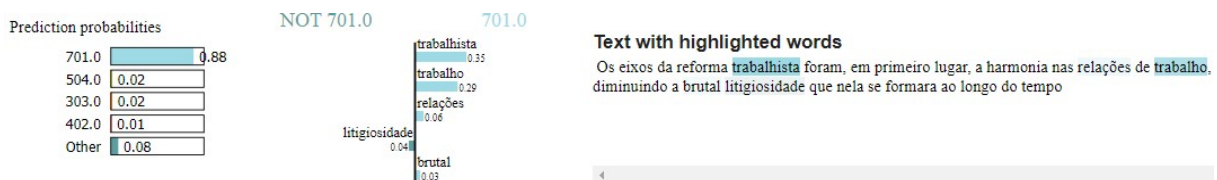


Figura 5.23: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo **701.0 - “Labour Groups”** (em azul claro).

Por fim, a Tabela 5.7 mostra os resultados do último artigo analisado, o do PSL. Aqui os resultados que o modelo encontra são muito divergentes do significado verdadeiro do

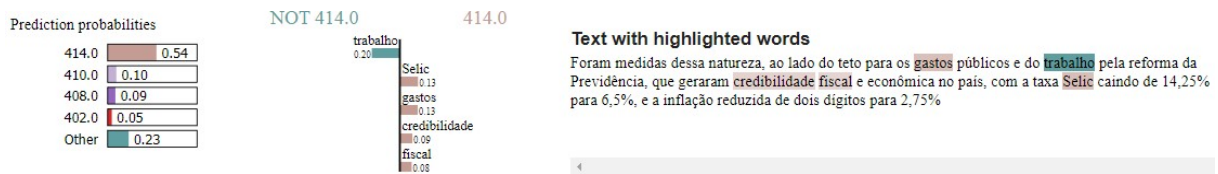


Figura 5.24: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo 414.0 - “Economic Orthodoxy” (em marrom).

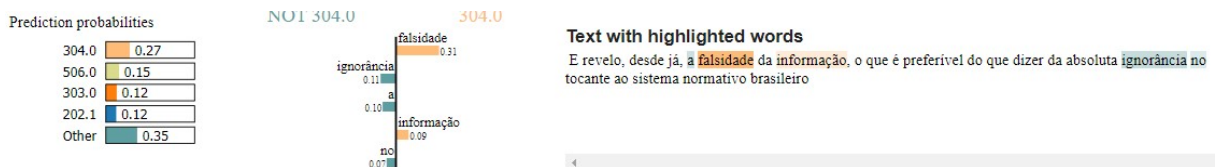


Figura 5.25: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo 304.0 - “Political Corruption” (em amarelo).

texto. Todas suas probabilidades de predição média são baixas e o modelo atribui categorias como 501.0 - “Environmental Protection” e 416.2 - “Sustainability” a um texto que defende posições de aumento de extração de recursos naturais, que inevitavelmente prejudicam o meio ambiente e não são práticas sustentáveis.

Explicações da classificação de sentenças que pertencem a estas classes estão presentes nas Figuras 5.26 a 5.27. A análise destas explicações mostra que o uso de um vocabulário similar ao encontrado nestes rótulos, como as palavras “ambiental”, “desenvolvimento” e “sustentável”, acabam confundindo o modelo a escolher esta classificação, mesmo que o seu significado seja justamente o oposto. Este fato ajuda a confirmar a hipótese citada acima relacionada a possibilidade de sobreajuste em algumas classes. Por este motivo, classificações com baixa acurácia devem sempre ser validadas por seres humanos qualificados ou apenas desconsideradas.

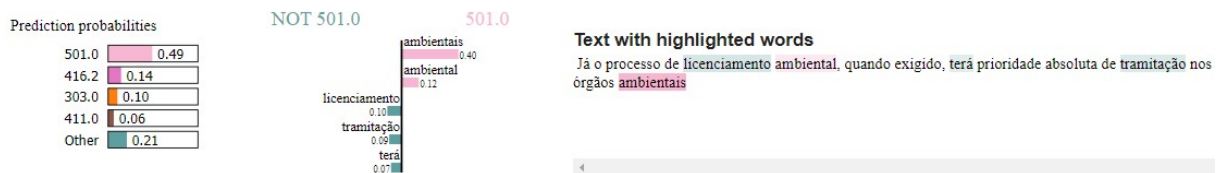


Figura 5.26: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo 501.0 - “Environmental Protection” (em rosa claro).

Tabela 5.7: Resultados do modelo classificador para o texto como um todo, relacionado ao artigo do PSL, com um viés político ligado a posições da direita e extrema direita.

Código	Categoria	Nº de sentenças	% de sentenças	Probabilidade Média
411.0	Technology and Infrastructure: +	5	33.33%	0.298
501.0	Environmental Protection	3	20%	0.357
416.2	Sustainability: +	2	13%	0.432
303.0	Governmental and Administrative Efficiency	1	7%	0.459
305.1	Political Authority: Party Competence	1	7%	0.112
410.0	Economic Growth: +	1	7%	0.193
504.0	Welfare State Expansion	1	7%	0.308
605.1	Law and Order: +	1	7%	0.360

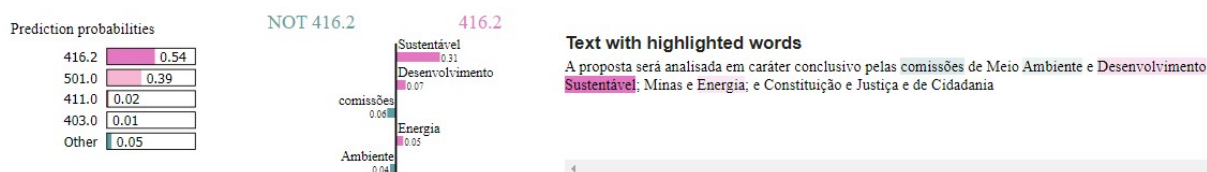


Figura 5.27: Explicação local referente a uma sentença (caixa de texto à direita) com categoria predita pelo modelo **416.2** - “Sustainability” (em rosa escuro).

# Capítulo 6

## Conclusão

Informação empodera e a comunicação é a base de todo o conhecimento. Compreender as intenções e vieses políticos de uma mensagem é um processo fundamental para a formação de uma sociedade mais esclarecida e que busca a dialética, no sentido de buscar uma verdade que está no todo, e não apenas partes. Entender estas tendências possibilita ao indivíduo uma visão muito mais ampla e que permite tomar decisões conscientes sobre o contexto em que está e seu futuro.

Este trabalho tenta auxiliar a explicitação destes vieses a partir da classificação de textos em assuntos políticos de uma forma automatizada e escalável e que encontra resultados razoáveis. Foram realizados diversos experimentos com o objetivo de entender qual modelo traria melhores, onde foram estudados seis modelos de aprendizagem de máquina e variadas configurações de hiperparâmetros. O experimento ainda abrange duas variações da base de dados, duas variações de pré-processamento no texto e a validação com textos fora da base de treino e teste.

O melhor modelo selecionado foi o BERTimbau para a base de dados agrupada em domínios políticos, que atinge um macro  $F-1$  score de 0.61. Estes resultados não chegam a alcançar o resultado de outros trabalhos relacionados na literatura atual. Todavia, a combinação do uso de 1) uma base de textos em na língua portuguesa, 2) categorizadas por um rótulo mais detalhado na classificação de viés e 3) treinado em modelos de alta performance que 4) tem resultados explicáveis e interpretáveis são contribuições inovadoras estudadas e apresentadas neste trabalho e não encontradas antes na literatura.

Neste estudo, foi possível complementar a análise dos resultados encontrados pelo modelo vencedor com explicações locais da classificação de diversas sentenças. Em adição, foram geradas explicações globais em todas as classes, onde algumas foram analisadas em profundidade. Estas análises tornaram estes resultados mais interpretáveis e possibilitaram a identificação de algumas falhas no pré-processamento dos dados e na baixa separabilidade de alguns rótulos específicos.

Infelizmente os resultados não permitem o uso deste modelo em qualquer texto genérico de forma satisfatoriamente confiável, ainda mais ao se tratar de um assunto tão sensível. Entretanto, mesmo com os resultados abaixo do esperado, o modelo encontra métricas que são significativamente maiores do que uma escolha ao acaso e que podem ser utilizadas para auxiliar e acelerar o processo de codificação manual feita por especialistas. Isto pode ser realizado em um ambiente de aprendizagem ativa, ao sugerir suas classes preditas para novas amostras e informar a probabilidade de predição, que em seguida são validadas por um humano qualificado e permitem um re-treino do modelo.

## 6.1 Trabalhos Futuros

O principal ponto que pode tornar estes modelos mais generalizáveis e com melhores resultados é expor o modelo a mais amostras codificadas nas categorias e subcategorias em sua fase de treinamento. Quanto mais exemplos, melhor tendem a ser os resultados e isto poderá ser feito em trabalhos futuros a partir de técnicas de *over-sampling* ou por meio de um agrupamento de categorias e subcategorias guiado por um especialista, para se diminuir o número de classes sem diminuir sua separabilidade.

Como o modelo que performa melhor é o BERTimbau, uma outra forma de melhorar o experimento é realizar um pré-processamento mais voltado a modelos de aprendizagem profunda. Como este tipo de modelo lida melhor com uma quantidade alta de atributos, pesquisas indicam que em alguns contextos este pré-processamento tradicional prejudica a capacidade do modelo de extrair contexto e significado do texto [69]. Ao invés de uniformizar os texto em letras minúsculas, mantém-se a caixa do texto original e o processo remover de pontuações e palavras de paradas é substituído por técnicas mais complexas de *word embeddings*, como o Word2Vec ou GloVe.

Outra melhoria que pode ser implementada é o aumento de recursos computacionais disponíveis para executar os experimentos, principalmente relacionado ao processamento paralelo. Isto facilitaria e aumentaria a escala dos experimentos com hiper-parâmetros, validação cruzada e geração de explicações, trazendo mais confiança aos resultados.



# Referências

- [1] Mullainathan, Sendhil e Andrei Shleifer: *Media bias*, 2002. 1
- [2] Lazer, David MJ, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild *et al.*: *The science of fake news*. *Science*, 359(6380):1094–1096, 2018. 1
- [3] Grimmer, Justin e Brandon M Stewart: *Text as data: The promise and pitfalls of automatic content analysis methods for political texts*. *Political analysis*, 21(3):267–297, 2013. 2, 24
- [4] Volkens, Andrea, Tobias Burst, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, Bernhard Weßels e Lisa Zehnter: *The manifesto data collection. manifesto project (mrg/cmp/marpor). version 2020b*, 2020. <https://doi.org/10.25522/manifesto.mpds.2020b>. 2, 30, 32
- [5] Chatsiou, Kakia e Slava Jankin Mikhaylov: *Deep learning for political science*. arXiv preprint arXiv:2005.06540, 2020. 2, 27
- [6] Frawley, William J, Gregory Piatetsky-Shapiro e Christopher J Matheus: *Knowledge discovery in databases: An overview*. *AI magazine*, 13(3):57–57, 1992. 4
- [7] Mining, What Is Data: *Data mining: Concepts and techniques*. Morgan Kaufmann, 10:559–569, 2006. 4
- [8] Han, Jiawei, Jian Pei e Micheline Kamber: *Data mining: concepts and techniques*. Elsevier, 2011. 5, 6, 14
- [9] Bonaccorso, Giuseppe: *Machine learning algorithms*. Packt Publishing Ltd, 2017. 5, 6
- [10] Dietterich, Tom: *Overfitting and undercomputing in machine learning*. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995. 6
- [11] Learning, Amazon Machine: *Developer guide*. Amazon Web Services, 2018. 6
- [12] Kotsiantis, Sotiris B, Ioannis Zaharakis, P Pintelas *et al.*: *Supervised machine learning: A review of classification techniques*. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007. 7
- [13] Mehra, Neha e Surendra Gupta: *Survey on multiclass classification methods*. 2013. 7

- [14] Goldberg, Yoav: *Neural network methods for natural language processing*. Synthesis lectures on human language technologies, 10(1):1–309, 2017. 7, 8
- [15] Liddy, Elizabeth D: *Natural language processing*. 2001. 8
- [16] Smelyakov, Kirill, Danil Karachevtsev, Denis Kulemza, Yehor Samoilenko, Oleh Patlan e Anastasiya Chupryna: *Effectiveness of preprocessing algorithms for natural language processing applications*. Em *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, páginas 187–191. IEEE, 2020. 8
- [17] Gentzkow, Matthew, Bryan Kelly e Matt Taddy: *Text as data*. Journal of Economic Literature, 57(3):535–74, September 2019. <https://www.aeaweb.org/articles?id=10.1257/jel.20181020>. 8, 9
- [18] Toman, Michal, Roman Tesar e Karel Jezek: *Influence of word normalization on text classification*. Proceedings of InSciT, 4:354–358, 2006. 9
- [19] Balakrishnan, Vimala e Ethel Lloyd-Yemoh: *Stemming and lemmatization: a comparison of retrieval performances*. 2014. 9
- [20] Ramos, Juan *et al.*: *Using tf-idf to determine word relevance in document queries*. Em *Proceedings of the first instructional conference on machine learning*, volume 242, páginas 29–48. Citeseer, 2003. 9
- [21] Zhang, Harry: *The optimality of naive bayes*. Aa, 1(2):3, 2004. 10
- [22] McCallum, Andrew, Kamal Nigam *et al.*: *A comparison of event models for naive bayes text classification*. Em *AAAI-98 workshop on learning for text categorization*, volume 752, páginas 41–48. Citeseer, 1998. 11
- [23] Rennie, Jason D, Lawrence Shih, Jaime Teevan e David R Karger: *Tackling the poor assumptions of naive bayes text classifiers*. Em *Proceedings of the 20th international conference on machine learning (ICML-03)*, páginas 616–623, 2003. 11
- [24] Gonzalez, Leandro de Azevedo: *Regressão logística e suas aplicações*. 2018. 11, 12
- [25] Shah, Kanish, Henil Patel, Devanshi Sanghvi e Manan Shah: *A comparative analysis of logistic regression, random forest and knn models for the text classification*. Augmented Human Research, 5(1):1–16, 2020. 11, 12
- [26] Aly, Mohamed: *Survey on multiclass classification methods*. Neural Netw, 19(1-9):2, 2005. 12
- [27] Aggarwal, Charu C e ChengXiang Zhai: *A survey of text classification algorithms*. Em *Mining text data*, páginas 163–222. Springer, 2012. 12
- [28] Dreiseitl, Stephan e Lucila Ohno-Machado: *Logistic regression and artificial neural network classification models: a methodology review*. Journal of biomedical informatics, 35(5-6):352–359, 2002. 12

- [29] Tong, Simon e Daphne Koller: *Support vector machine active learning with applications to text classification*. Journal of machine learning research, 2(Nov):45–66, 2001. 12
- [30] Burges, Christopher JC: *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery, 2(2):121–167, 1998. 13
- [31] Rennie, Jason DM e Ryan Rifkin: *Improving multiclass text classification with the support vector machine*. 2001. 12, 13
- [32] Lorena, Ana Carolina e André CPLF De Carvalho: *Uma introdução às support vector machines*. Revista de Informática Teórica e Aplicada, 14(2):43–67, 2007. 13
- [33] Lorena, Ana Carolina: *Investigação de estratégias para a geração de máquinas de vetores de suporte multiclass*. Tese de Doutorado, Universidade de São Paulo, 2006. 14
- [34] Natekin, Alexey e Alois Knoll: *Gradient boosting machines, a tutorial*. Frontiers in neurorobotics, 7:21, 2013. 14, 15
- [35] Charbuty, Bahzad e Adnan Abdulazeez: *Classification based on decision tree algorithm for machine learning*. Journal of Applied Science and Technology Trends, 2(01):20–28, 2021. 14
- [36] Friedman, Jerome H: *Stochastic gradient boosting*. Computational statistics & data analysis, 38(4):367–378, 2002. 15
- [37] Chen, Tianqi e Carlos Guestrin: *Xgboost: A scalable tree boosting system*. Em *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, páginas 785–794, 2016. 15
- [38] Devlin, Jacob, Ming Wei Chang, Kenton Lee e Kristina Toutanova: *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805, 2018. 16
- [39] González-Carvajal, Santiago e Eduardo C Garrido-Merchán: *Comparing bert against traditional machine learning text classification*. arXiv preprint arXiv:2005.13012, 2020. 16
- [40] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser e Illia Polosukhin: *Attention is all you need*. arXiv preprint arXiv:1706.03762, 2017. 16
- [41] Souza, Fábio, Rodrigo Nogueira e Roberto Lotufo: *BERTimbau: pretrained BERT models for Brazilian Portuguese*. Em *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020. 16
- [42] Grandini, Margherita, Enrico Bagli e Giorgio Visani: *Metrics for multi-class classification: an overview*. arXiv preprint arXiv:2008.05756, 2020. 17, 18

- [43] Hossin, Mohammad e Md Nasir Sulaiman: *A review on evaluation metrics for data classification evaluations*. International journal of data mining & knowledge management process, 5(2):1, 2015. 18
- [44] Burkart, Nadia e Marco F Huber: *A survey on the explainability of supervised machine learning*. Journal of Artificial Intelligence Research, 70:245–317, 2021. 19, 20, 40
- [45] Ribeiro, Marco Tulio, Sameer Singh e Carlos Guestrin: *" why should i trust you?" explaining the predictions of any classifier*. Em *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, páginas 1135–1144, 2016. 20, 21, 22, 23, 40
- [46] Krause, Andreas e Daniel Golovin: *Submodular function maximization*. Tractability, 3:71–104, 2014. 22
- [47] Miller, Tim: *Explanation in artificial intelligence: Insights from the social sciences*. Artificial intelligence, 267:1–38, 2019. 23
- [48] Adadi, Amina e Mohammed Berrada: *Peeking inside the black-box: a survey on explainable artificial intelligence (xai)*. IEEE access, 6:52138–52160, 2018. 23
- [49] Wilkerson, John e Andreu Casas: *Large-scale computerized text analysis in political science: Opportunities and challenges*. Annual Review of Political Science, 20:529–544, 2017. 25
- [50] Molinari, Adriana Miranda: *Aprendendo com discursos: uma análise em alta dimensão da ideologia e polarização política na câmara dos deputados federais do brasil*. 2020. 25, 50
- [51] Cavalcanti, Rafael Dutra: *Classificação de tendências políticas em notícias via mineração de texto e redes neurais sem peso*. Rio de Janeiro, 2017. 26
- [52] Temporão, Mickael, Corentin Vande Kerckhove, Clifton van der Linden, Yannick Dufresne e Julien M Hendrickx: *Ideological scaling of social media users: a dynamic lexicon approach*. Political Analysis, 26(4):457–473, 2018. 26, 50
- [53] Biessmann, Felix: *Automating political bias prediction*. arXiv preprint arXiv:1608.02195, 2016. 26, 49
- [54] Zirn, Căcilia, Goran Glavaš, Federico Nanni, Jason Eichorts e Heiner Stuckenschmidt: *Classifying topics and detecting topic shifts in political manifestos*. University of Zagreb, 2016. 27, 49, 52
- [55] Gangula, Rama Rohit Reddy, Suma Reddy Duggenpudi e Radhika Mamidi: *Detecting political bias in news articles using headline attention*. Em *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, páginas 77–84, 2019. 27, 50

- [56] Bibal, Adrien, Michael Lognoul, Alexandre De Streel e Benoît Frénay: *Legal requirements on explainability in machine learning*. Artificial Intelligence and Law, 29(2):149–169, 2021. 27
- [57] Karim, Md, Till Döhmen, Dietrich Rebholz-Schuhmann, Stefan Decker, Michael Cochez, Oya Beyan *et al.*: *Deepcovidexplainer: Explainable covid-19 predictions based on chest x-ray images*. arXiv preprint arXiv:2004.04582, 2020. 28
- [58] Chen, Yuanfang, Liu Ouyang, Forrest Sheng Bao, Qian Li, Lei Han, Baoli Zhu, Yaorong Ge, Patrick Robinson, Ming Xu, Jie Liu *et al.*: *An interpretable machine learning framework for accurate severe vs non-severe covid-19 clinical type classification*. Available at SSRN 3638427, 2020. 28
- [59] Kurasinski, Lukas e Radu Casian Mihailescu: *Towards machine learning explainability in text classification for fake news detection*. Em *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, páginas 775–781. IEEE, 2020. 28
- [60] Santos, Pedro Lamkowski dos: *Inteligência artificial explicável aplicada à classificação de desinformação*. 28
- [61] Abramovich, Felix e Marianna Pensky: *Classification with many classes: challenges and pluses*. Journal of Multivariate Analysis, 174:104536, 2019. 31
- [62] Sechidis, Konstantinos, Grigorios Tsoumakas e Ioannis Vlahavas: *On the stratification of multi-label data*. Em *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, páginas 145–158. Springer, 2011. 36
- [63] *Cross validation*. [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html). (Accessed on 30/08/2021). 37
- [64] Tibshirani, R.: *Regression shrinkage and selection via the lasso*. J. Royal. Statist. Soc B., 58:267–288, 1996. 38
- [65] Owen, Art B.: *A robust hybrid of lasso and ridge regression*. 2006. 38
- [66] *Hyperparameter tuning for support vector machines — c and gamma parameters*. <https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a509741> (Accessed on 30/08/2021). 38
- [67] Zou, H. e T. Hastie: *Regularization and variable selection via the elastic net*. Journal of the Royal Statistical Society, 67:301—320, 2005. 38
- [68] Holzinger, Andreas, Georg Langs, Helmut Denk, Kurt Zatloukal e Heimo Müller: *Causability and explainability of artificial intelligence in medicine*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(4):e1312, 2019. 40
- [69] Maslej-Krešňáková, Viera, Martin Sarnovský, Peter Butka e Kristína Machová: *Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification*. Applied Sciences, 10(23):8631, 2020. 65

# Apêndice A

## Tabela de códigos do Manifesto Project

### A.1 Base de Dados A

Tabela A.1: Distribuição de sentenças nos códigos das categorias e subcategorias da base original de sentenças, organizada em ordem decrescente

Código	Descrição	Amostras
000	Uncoded	9583
504.0	Welfare State Expansion	7868
411.0	Technology and Infrastructure: Positive	7145
506.0	Education Expansion	5254
503.0	Equality: Positive	5057
303.0	Governmental and Administrative Efficiency	4677
501.0	Environmental Protection	3901
502.0	Culture: Positive	3047
202.1	Democracy General: Positive	2881
416.2	Sustainability: Positive	2804
701.0	Labour Groups: Positive	2730
410.0	Economic Growth: Positive	2562
402.0	Incentives: Positive	2298
605.1	Law and Order: Positive	2235
703.1	Agriculture and Farmers: Positive	2015
403.0	Market Regulation	2012
414.0	Economic Orthodoxy	1756

301.0	Decentralization	1745
401.0	Free Market Economy	1227
304.0	Political Corruption	1174
408.0	Economic Goals	1114
305.1	Political Authority: Party Competence	1080
706.0	Non-economic Demographic Groups	995
104.0	Military: Positive	911
413.0	Nationalisation	898
409.0	Keynesian Demand Management	849
107.0	Internationalism: Positive	838
505.0	Welfare State Limitation	742
605.0	Law and Order	716
601.1	National Way of Life General: Positive	621
703.0	Agriculture and Farmers: Positive	608
103.2	Anti-Imperialism: Foreign Financial Influence	581
201.2	Human Rights	563
108.0	European Community/Union: Positive	553
202.0	Democracy	511
110.0	European Community/Union: Negative	484
305.2	Political Authority: Personal Competence	473
412.0	Controlled Economy	471
705.0	Underprivileged Minority Groups	455
305.0	Political Authority	449
606.1	Civic Mindedness General: Positive	443
404.0	Economic Planning	397
407.0	Protectionism: Negative	380
603.0	Traditional Morality: Positive	336
103.1	Anti-Imperialism: State Centred Anti-Imperialism	318
203.0	Constitutionalism: Positive	305
201.1	Freedom	268
415.0	Marxist Analysis	253
406.0	Protectionism: Positive	234
109.0	Internationalism: Negative	207
105.0	Military: Negative	200
101.0	Foreign Special Relationships: Positive	197
607.1	Multiculturalism General: Positive	185
605.2	Law and Order: Negative	184

607.3	Multiculturalism: Indigenous rights: Positive	166
204.0	Constitutionalism: Negative	152
405.0	Corporatism/Mixed Economy	152
202.4	Direct Democracy: Positive	146
201.0	Freedom and Human Rights	141
507.0	Education Limitation	121
704.0	Middle Class and Professional Groups	114
106.0	Peace	111
607.2	Multiculturalism: Immigrants Diversity	107
602.1	National Way of Life General: Negative	104
302.0	Centralisation	100
604.0	Traditional Morality: Negative	93
606.2	Civic Mindedness: Bottom-Up Activism	89
305.3	Political Authority: Strong government	86
606.0	Civic Mindedness: Positive	81
602.2	National Way of Life: Immigration: Positive	70
608.0	Multiculturalism: Negative	59
202.3	Representative Democracy: Positive	51
702.0	Labour Groups: Negative	51
416.0	Anti-Growth Economy: Positive	41
601.0	National Way of Life: Positive	38
305.5	Transition: Pre-Democratic Elites: Negative	38
103.0	Anti-Imperialism	35
416.1	Anti-Growth Economy: Positive	25
202.2	Democracy General: Negative	21
601.2	National Way of Life: Immigration: Negative	20
607.0	Multiculturalism: Positive	15
703.2	Agriculture and Farmers: Negative	13
608.1	Multiculturalism General: Negative	6
305.6	Transition: Rehabilitation and Compensation	2
608.2	Multiculturalism: Immigrants Assimilation	2
102.0	Foreign Special Relationships: Negative	1

## A.2 Base de Datos B



Tabela A.2: Distribuição de sentenças nos códigos das domínios da base original de sentenças, organizada em ordem decrescente..

<b>Código do Domínio</b>	<b>Domínio</b>	<b>Número de Sentenças</b>
5	Welfare and Quality of Life	25990
4	Economy	24618
3	Political System	9824
0	No Domain	9583
7	Social Groups	6981
6	Fabric of Society	5570
2	Freedom and Democracy	5039
1	External Relations	4436

# Apêndice B

## Explicações Globais das Classificações do BERTimbau

### B.1 Base de Dados A

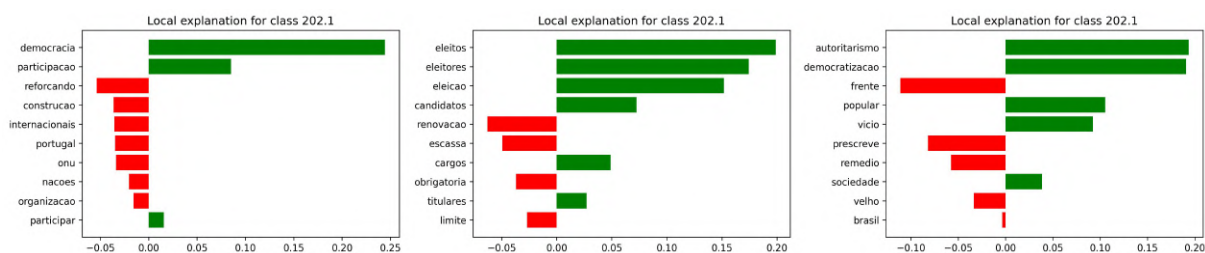


Figura B.1: Explicação global referente a predição da categoria **202.1** - “Democracy General: +”.

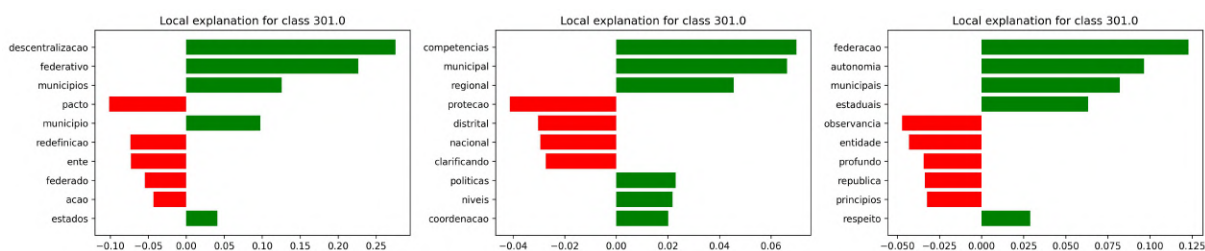


Figura B.2: Explicação global referente a predição da categoria **301.0** - “Decentralization”.

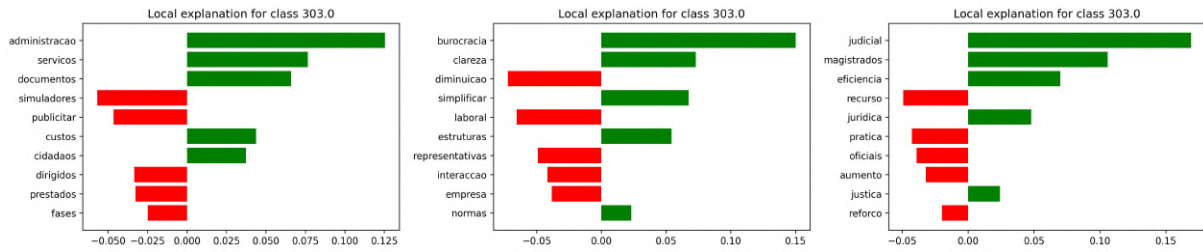


Figura B.3: Explicação global referente a predição da categoria 303.0 - “Governmental and Administrative Efficiency”.

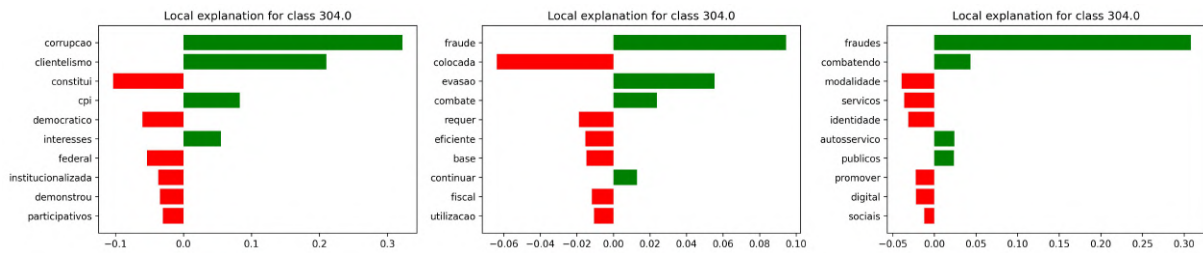


Figura B.4: Explicação global referente a predição da categoria 304.0 - “Political Corruption”.

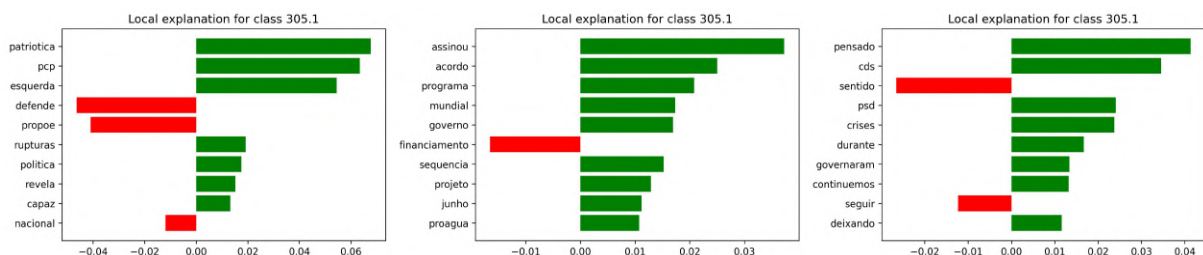


Figura B.5: Explicação global referente a predição da categoria 305.1 - “Political Authority: Party Competence”.

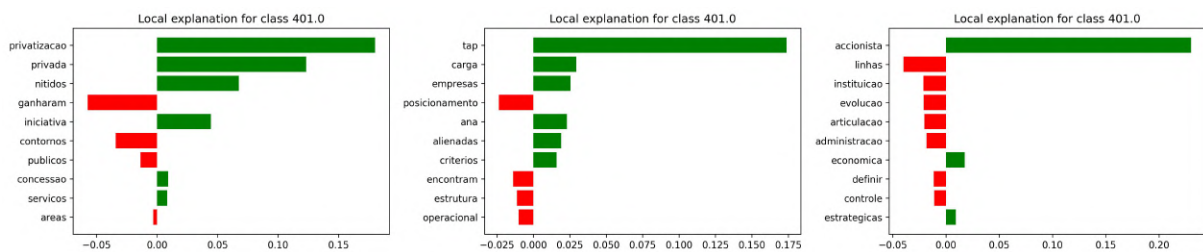


Figura B.6: Explicação global referente a predição da categoria 401.0 - “Free Market Economy”.

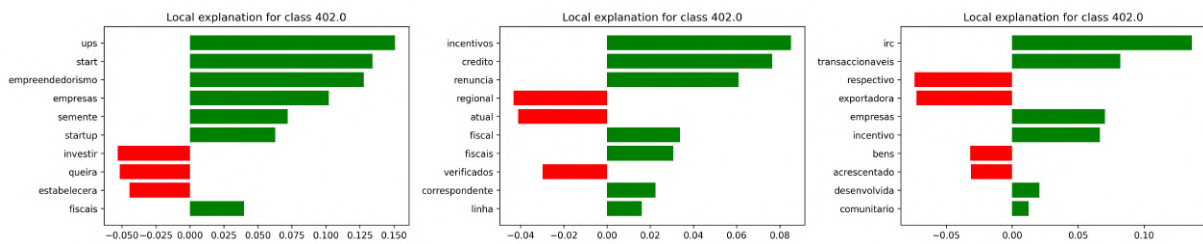


Figura B.7: Explicação global referente a predição da categoria 402.0 - “Incentives: +”.

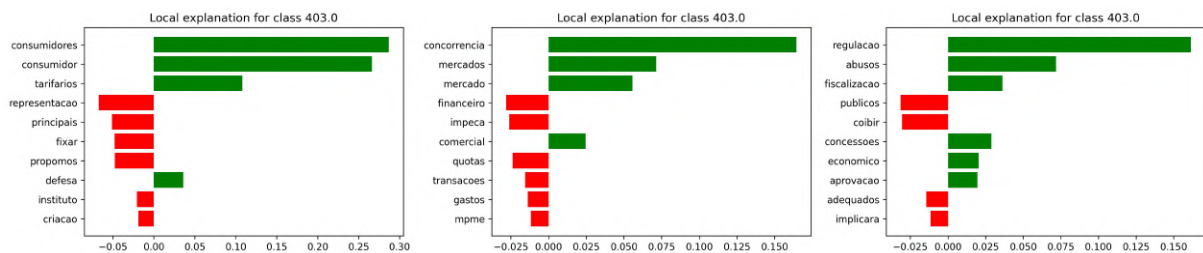


Figura B.8: Explicação global referente a predição da categoria 403.0 - “Market Regulation”.

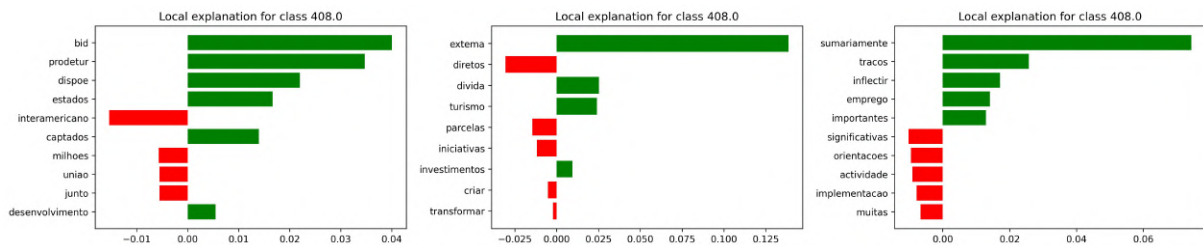


Figura B.9: Explicação global referente a predição da categoria 408.0 - “Economic Goals”.

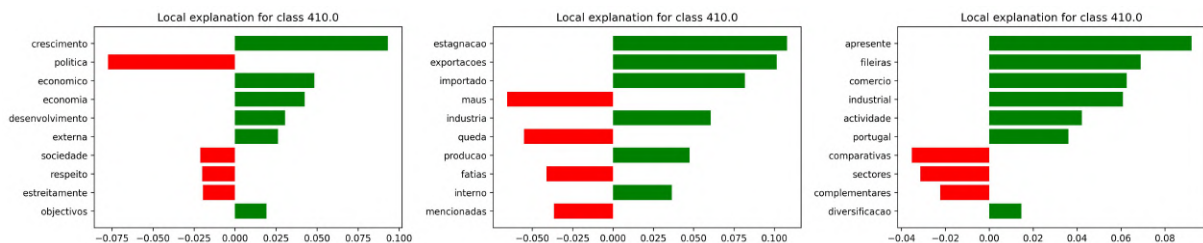


Figura B.10: Explicação global referente a predição da categoria 410.0 - “Economic Growth: +”.

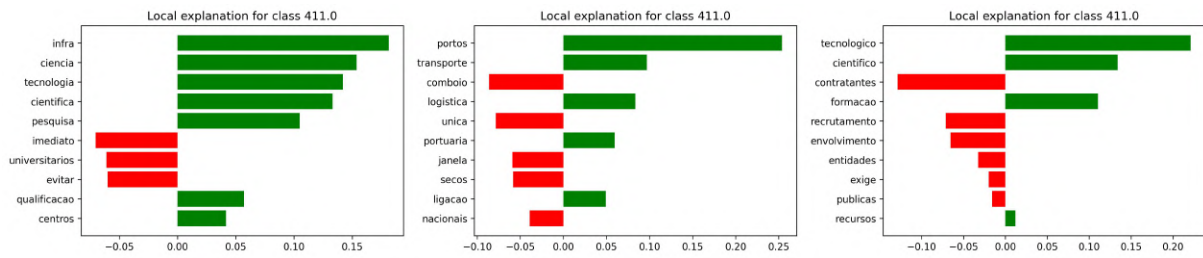


Figura B.11: Explicação global referente a predição da categoria 411.0 - “Technology and Infrastructure: +”.

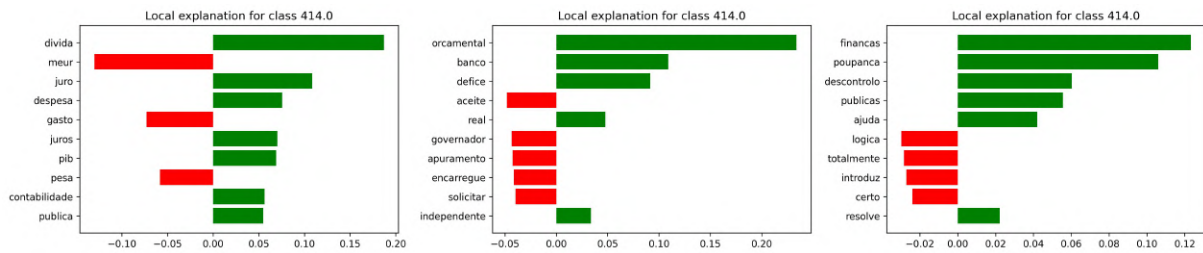


Figura B.12: Explicação global referente a predição da categoria 414.0 - “Economic Orthodoxy”.

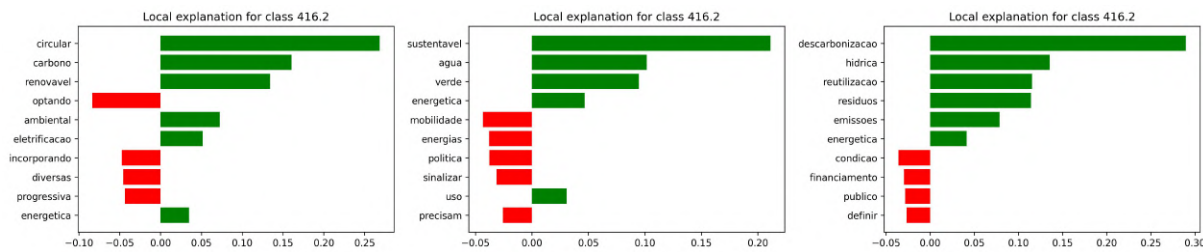


Figura B.13: Explicação global referente a predição da categoria 416.2 - “Sustainability: +”.

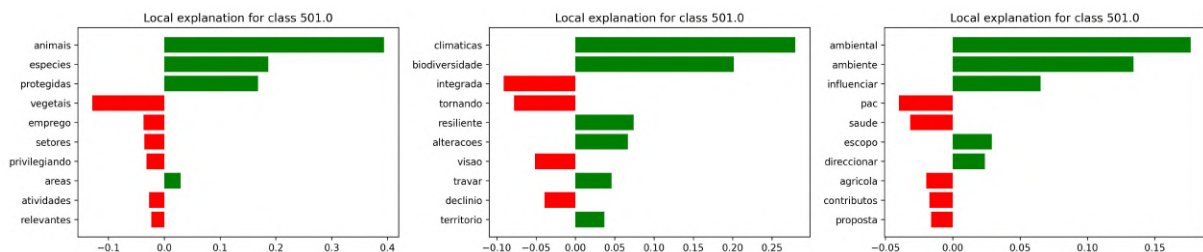


Figura B.14: Explicação global referente a predição da categoria 501.0 - “Environmental Protection”.

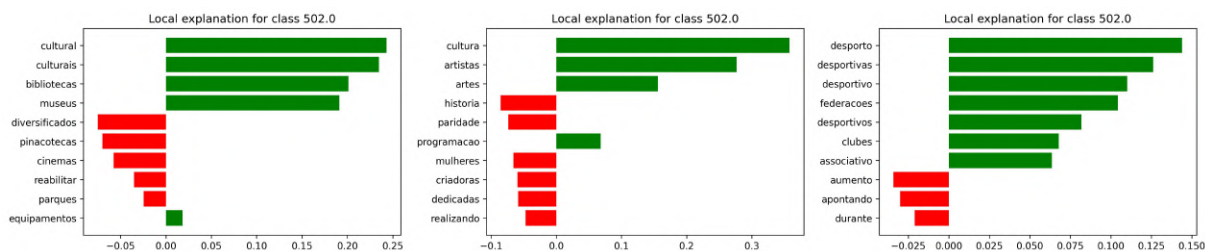


Figura B.15: Explicação global referente a predição da categoria 502.0 - “Culture: +”.

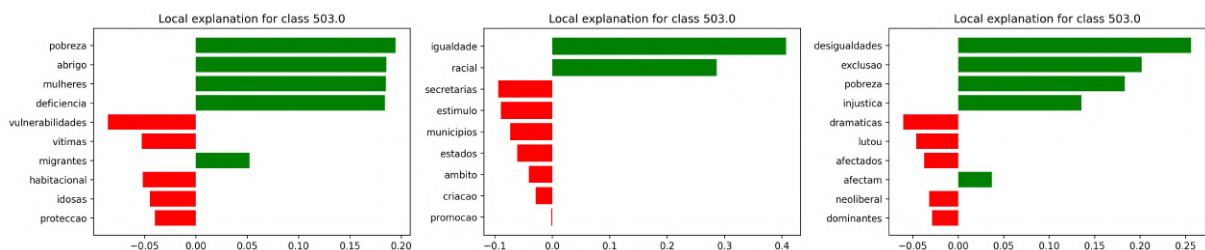


Figura B.16: Explicação global referente a predição da categoria 503.0 - “Equality: +”.

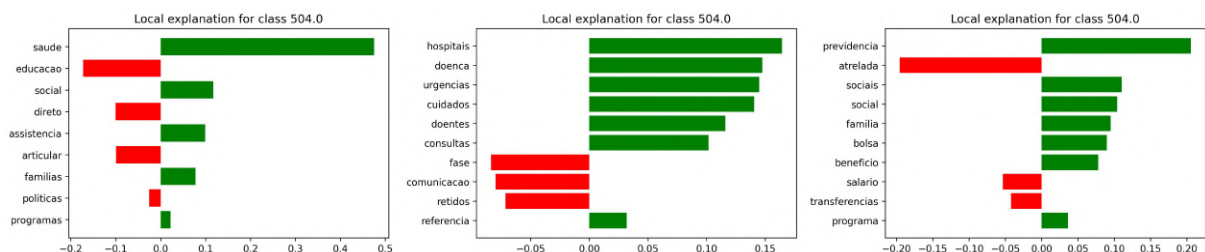


Figura B.17: Explicação global referente a predição da categoria 504.0 - “Welfare State Expansion”.

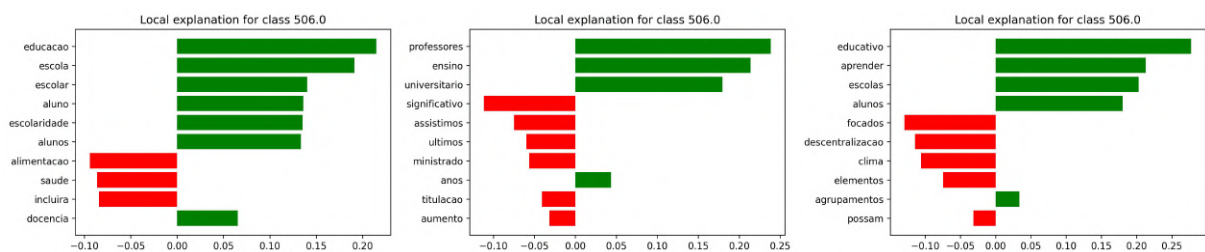


Figura B.18: Explicação global referente a predição da categoria 506.0 - “Education Expansion”.



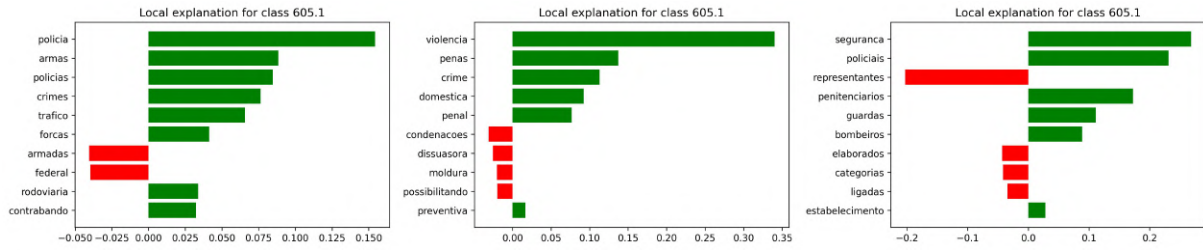


Figura B.19: Explicação global referente a predição da categoria 605.1 - “Law and Order: +”.

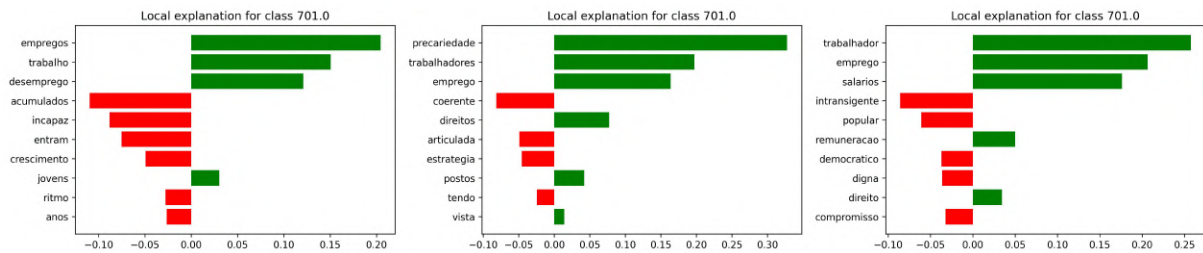


Figura B.20: Explicação global referente a predição da categoria 701.0 - “Labour Groups: +”.

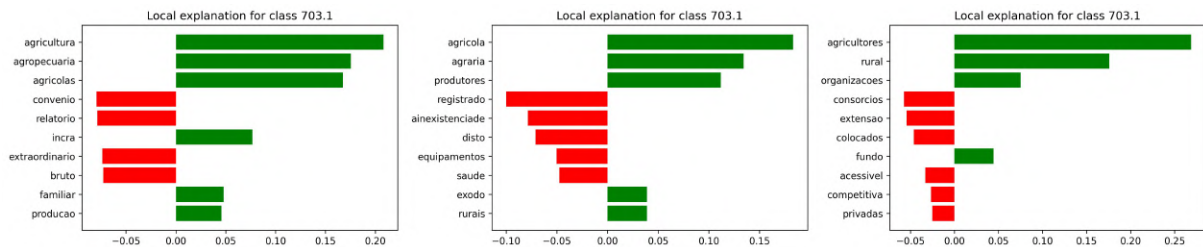


Figura B.21: Explicação global referente a predição da categoria 703.1 - “Agriculture and Farmers: +”.

## B.2 Base de Dados B

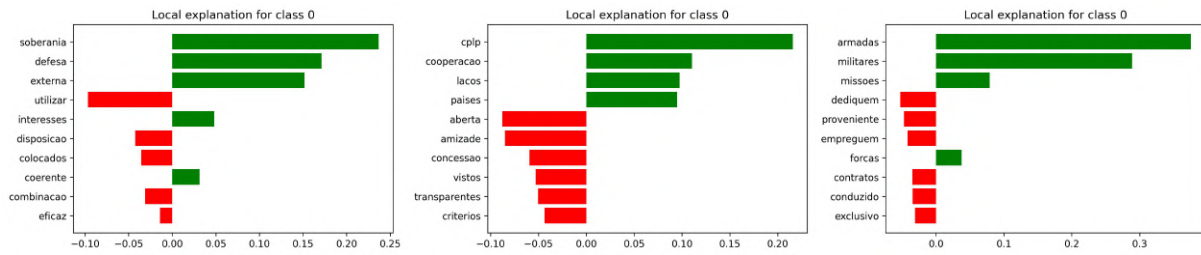


Figura B.22: Explicação global referente a predição do domínio 1 - “External Relations”.

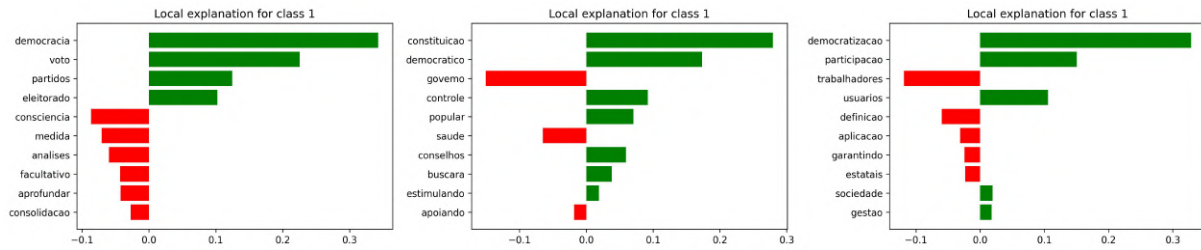


Figura B.23: Explicação global referente a predição do domínio 2 - “Freedom and Democracy”.

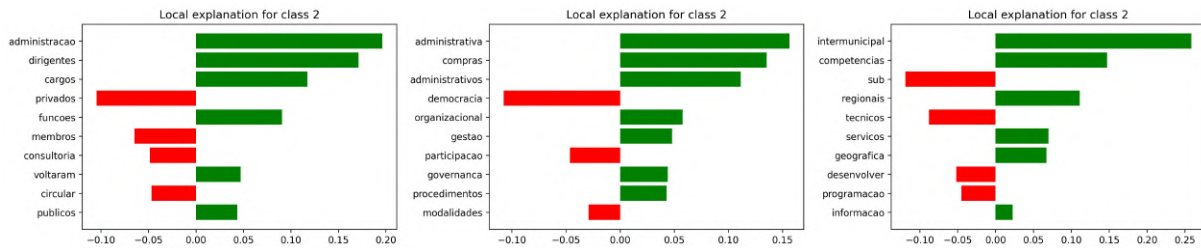


Figura B.24: Explicação global referente a predição do domínio 3 - “Political System”.



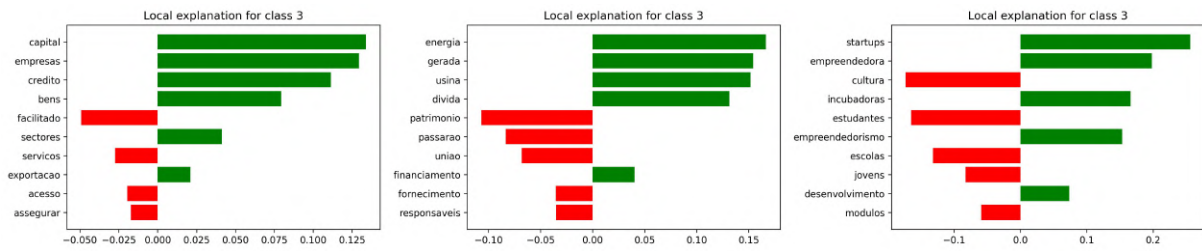


Figura B.25: Explicação global referente a predição do domínio 4 - “Economy”.

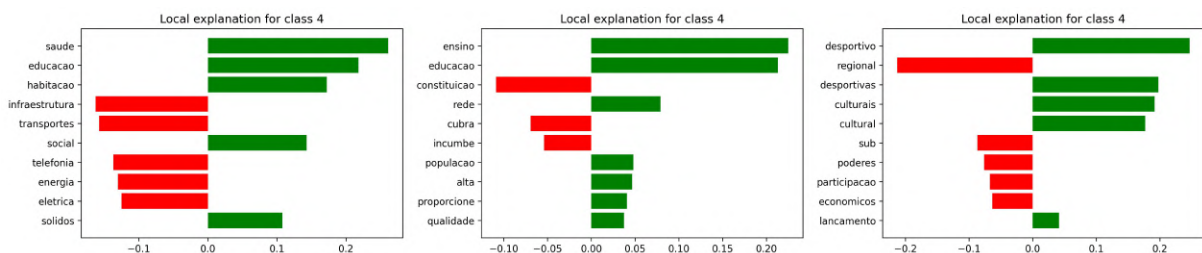


Figura B.26: Explicação global referente a predição do domínio 5 - “Welfare and Quality of Life”.

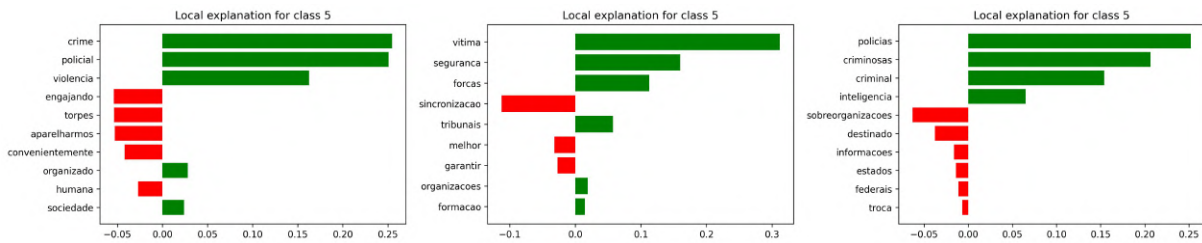


Figura B.27: Explicação global referente a predição do domínio 6 - “Fabric of Society”.

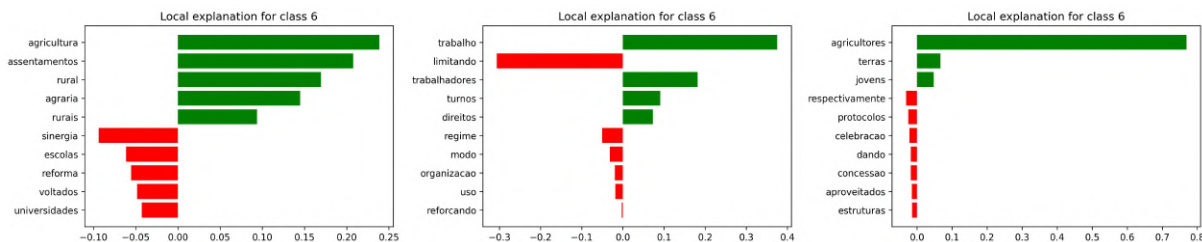


Figura B.28: Explicação global referente a predição do domínio 7 - “Social Groups”.