



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Classificação de Textos de Bases Desbalanceadas aplicada às Solicitações Cidadãs

Mayra de M. Campos

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Marcelo Antonio Marotta

Brasília
2026

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

dC198c	de Moraes Campos, Mayra Classificação de Textos de Bases Desbalanceadas aplicada às Solicitações Cidadãs / Mayra de Moraes Campos; orientador Marcelo Antonio Marotta. Brasília, 2026. 81 p. Dissertação(Mestrado Profissional em Computação Aplicada) Universidade de Brasília, 2026. 1. Processamento de Linguagem Natural. 2. Classes Desbalanceadas. 3. Classificação de Textos. 4. Automação de Serviços Públicos. 5. Energy based Flow Classifier. I. Marotta, Marcelo Antonio, orient. II. Título.
--------	--



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Classificação de Textos de Bases Desbalanceadas aplicada às Solicitações Cidadãs

Mayra de M. Campos

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Marcelo Antonio Marotta (Orientador)
UnB - PPCA

Prof. Dr. Marcos Fagundes Caetano
UnB - PPCA

Dr. Camila Thé Pontes
Barcelona Supercomputing Center - BSC

Prof.a Dr.a Edna Dias Canedo
Coordenadora do Programa de Pós-graduação em Computação Aplicada

Brasília, 27 de abril de 2026

Dedicatória

Eu dedico mais esta vitória a Deus e à minha família, pilares da minha vida, cuja presença foi essencial ao longo de toda a trajetória de construção deste trabalho.

Agradecimentos

Agradeço a Deus, que nunca falha.

À minha família, que é o meu porto seguro.

Ao meu orientador, Prof. Dr. Marcelo Antonio Marotta, pelas orientações e apoio.

Aos demais professores do Programa de Pós-graduação em Computação Aplicada, pelo suporte.

E à Subcontroladoria de Tecnologia da Informação e Comunicação da CGDF pela confiança.

Resumo

Este estudo de caso investigou alternativas para automatizar a classificação binária de textos com classes desbalanceadas no contexto do Participa DF, uma plataforma de participação social do GDF. Esse canal digital tem como objetivo ampliar a participação cidadã na gestão pública do Distrito Federal por meio de manifestações textuais. Esses textos são enviados pelos cidadãos ao Participa DF e categorizados manualmente pelos usuários entre dois sistemas: o e-SIC DF e o Ouv-DF. Diversas técnicas de PLN foram aplicadas, como redução de dimensionalidade por PCA, métodos de redimensionamento (como o NM), além de múltiplos algoritmos de AM, tais como BERTimbau, LR, SVM e LGBM. A avaliação priorizou a capacidade de identificar a classe minoritária e-SIC DF, cuja classificação correta é especialmente difícil. O desbalanceamento normalmente compromete a eficácia de modelos tradicionais de AM. Entre os métodos analisados, destaca-se o EFC, originalmente desenvolvido para detecção de anomalias em redes e explorado neste trabalho pela primeira vez em tarefas de classificação textual. Esse algoritmo utiliza estatísticas inversas para treinar modelos. Os resultados mostram que o EFC, quando treinado exclusivamente com exemplos da classe e-SIC DF e submetido a um nível reduzido de pré-processamento, atinge alto *Recall* (94%), demonstrando forte capacidade de reconhecer a classe minoritária. Entretanto, apresentou elevada taxa de FDR, o que implicaria aumento significativo do retrabalho por parte dos servidores responsáveis pelo tratamento das manifestações e pela reclassificação. Em comparação com o EFC, modelos clássicos e modelos baseados em *transformers* apresentaram desempenho mais equilibrado entre *Recall*, *F1-Score*, AUC-ROC e Acurácia. Conclui-se que, apesar de o EFC ser promissor para cenários de detecção de anomalias em conjuntos de dados textuais desbalanceados, ele gera muitos falsos positivos. Para implantação no ambiente de produção da plataforma Participa DF, o modelo treinado com o algoritmo SVM, com redução de dimensionalidade via PCA e uso de textos sintéticos, mostrou-se a alternativa tecnicamente mais viável.

Palavras-chave: Processamento de Linguagem Natural, Classes Desbalanceadas, Classificação de Textos, Automação de Serviços Públicos, Aprendizagem de Máquina, *Energy-based Flow Classifier*

Abstract

This case study investigated alternatives for automating the binary classification of texts with imbalanced classes in the context of Participa DF, a social participation platform of the GDF. This digital channel aims to enhance citizen participation in the public administration of the Federal District through textual submissions. These texts are submitted by citizens to Participa DF and manually categorized by users into two systems: e-SIC DF and Ouv-DF. Several NLP techniques were applied, including dimensionality reduction via PCA, resampling methods (such as NM), as well as multiple ML algorithms, including BERTimbau, LR, SVM, and LGBM. The evaluation prioritized the ability to identify the minority class, e-SIC DF, whose correct classification is particularly challenging. Class imbalance typically compromises the effectiveness of traditional ML models. Among the methods analyzed, the EFC algorithm stands out. Originally developed for anomaly detection in networks, it is explored in this work for the first time in textual classification tasks. This algorithm employs inverse statistics to train models. The results show that EFC, when trained exclusively on examples from the e-SIC DF class and subjected to a reduced level of preprocessing, achieves high Recall (94%), demonstrating a strong ability to recognize the minority class. However, it also exhibited a high FDR, which would imply a significant increase in rework for public servants responsible for handling and reclassifying the submissions. In comparison with EFC, classical models and transformer -based models exhibited a more balanced performance across Recall, F1-Score, AUC-ROC, and Accuracy. It is concluded that, although EFC is promising for anomaly detection scenarios in imbalanced textual datasets, it generates a large number of false positives. For deployment in the production environment of the Participa DF platform, the model trained using the SVM algorithm, combined with dimensionality reduction via PCA and the use of synthetic texts, proved to be the most technically viable alternative.

Keywords: Natural Language Processing, Imbalanced Data, Text Classification, Public Services Automation; Machine Learning, Energy-based Flow Classifier

Sumário

1	Introdução	1
1.1	Contexto Geral	1
1.2	Contexto Específico	1
1.3	Problema e Hipótese	3
1.3.1	Definição do Problema	3
1.3.2	Hipótese	4
1.4	Justificativa e Contribuições	4
1.4.1	Justificativa do tema	4
1.4.2	Contribuições Esperadas	5
1.5	Estrutura do Trabalho	6
2	Revisão do Estado da Arte	7
2.1	Trabalhos Relacionados	8
2.1.1	Classificação de Textos Desbalanceados na Administração Pública	8
2.1.2	Classificadores de Uma Classe	10
2.1.3	Redução de Dimensionalidade	11
2.1.4	Custo de Treinamento e Desempenho	12
2.1.5	Geração de Textos Sintéticos	12
2.1.6	Métricas para Classificação de Textos	13
2.1.7	Metodologias	14
2.2	Considerações	15
3	Fundamentação Teórica	17
3.1	Trajetória de Ciência de Dados - TCDs	17
3.1.1	Atividades Exploratórias	18
3.1.2	Atividades de Gerenciamento de Dados	19
3.1.3	Atividades Direcionadas a Objetivos (CRISP-DM)	19
3.1.4	Trajetória Escolhida	21
3.2	Processamento de Linguagem Natural - PLN	21

3.2.1	Classificação de Textos	21
3.2.2	Classificação Binária	21
3.3	Pré-processamento de Textos	22
3.3.1	Normalização Textual	22
3.3.2	Vetorização de Textos	23
3.4	Balanceamento de Dados - Redimensionamento	24
3.4.1	Subamostragem	24
3.4.2	Sobreamostragem	24
3.5	Redução de Dimensionalidade	25
3.5.1	Análise de Componentes Principais - PCA	25
3.6	Modelos	26
3.6.1	Energy-based Flow Classifier - EFC	26
3.6.2	BERTimbau	28
3.6.3	Light Gradient Boosting Machine - LGBM	28
3.6.4	Support Vector Machine - SVM	29
3.6.5	Logistic Regression - LR	29
3.6.6	K-Nearest Neighbors - KNN	29
3.7	Métrica de Avaliação	29
3.7.1	Recall	30
3.7.2	False Discovery Rate - FDR	30
3.7.3	Acurácia	31
3.7.4	AUC-ROC	31
3.7.5	F1-Score	32
4	Experimentos	33
4.1	Metodologia - Trajetória de Ciência de Dados (TCD)	33
4.2	Exploração do Objetivo	34
4.3	Entendimento do Negócio	34
4.3.1	Objetivos da aplicação	34
4.3.2	Situação	35
4.3.3	Determinação das metas	36
4.3.4	Plano do Projeto	36
4.4	Exploração da Fonte de Dados	37
4.5	Aquisição dos Dados	37
4.6	Entendimento dos Dados	38
4.7	Arquitetura dos Dados	38
4.8	Preparação dos Dados	40
4.8.1	Aplicadas a todo o corpus	41

4.8.2	Aplicadas em diferentes etapas	41
4.8.3	Dados de Treinamento, Teste e Validação	43
4.9	Modelagem	45
4.10	Avaliação do Modelo	48
5	Resultados e Análises	50
5.1	Resultados	50
5.2	Análises	56
6	Considerações Finais	60
6.1	Conclusões	60
6.2	Trabalhos Futuros	61
	Referências	62

Lista de Figuras

3.1	Diagrama do CRISP-DM. Fonte: adaptação do artigo de MARTÍNEZ-PLUMED <i>et al.</i> [1]	18
3.2	Diagrama com os componentes possíveis das Trajetórias de Ciência de Dados. Fonte: adaptação do artigo de MARTÍNEZ-PLUMED <i>et al.</i> [1].	18
4.1	Trajetória de Ciência de Dados selecionada	33
4.2	Trajetória de Ciência de Dados Escolhida	34
4.3	Quantidade de Registros por mês no sistema e-SIC DF	39
4.4	Quantidade de Registros por mês no sistema Ouv-DF	40
4.5	Histograma Palavras por Relato - classe e-SIC DF	41
4.6	Histograma Palavras por Relato - classe Ouv-DF	41
4.7	Pré-processamentos realizados	42
4.8	Pré-processamentos de Textos Algoritmos Clássicos	46
4.9	Pré-processamentos de Textos para o BERTimbau	47
4.10	Pré-processamentos de Textos para o EFC	47
4.11	Treinamento dos Modelos	48
5.1	Curvas ROC	57

Lista de Tabelas

2.1	Trabalhos Relacionados	15
3.1	Matriz de Confusão	30
4.1	Exemplos de pré-processamento textual	42
5.1	Avaliação dos Modelos	51
5.2	EFC+TFIDF – Classe e-SIC DF – Sem texto Sintético	52
5.3	EFC+TFIDF – Classe Ouv-DF – Sem texto Sintético	52
5.4	EFC+TFIDF+LPCA – Classe e-SIC DF – Sem texto Sintético	52
5.5	EFC+TFIDF+LPCA – Classe Ouv-DF – Sem texto Sintético	53
5.6	EFC+TFIDF+SPCA – Classe e-SIC DF – Sem texto Sintético	53
5.7	EFC+TFIDF+SPCA – Classe Ouv-DF – Sem texto Sintético	54
5.8	EFC+TFIDF+RPCA – Classe e-SIC DF – Sem texto Sintético	54
5.9	EFC+TFIDF+RPCA – Classe Ouv-DF – Sem texto Sintético	55
5.10	EFC+TFIDF+Pefc – Classe e-SIC DF – Sem texto Sintético	55
5.11	EFC+TFIDF+Pefc – Classe Ouv-DF – Sem texto Sintético	56
5.12	Top 10 modelos por Recall no e-SIC DF	56
5.13	Top 10 Modelos ranqueados pelos melhores valores de $F1-Score$ (e-SIC DF), Recall (e-SIC DF), FDR (e-SIC DF), Acurácia e AUC-ROC	58
5.14	Resumo dos hiperparâmetros utilizados no treinamento dos modelos da Tabela 5.13	59

Lista de Abreviaturas e Siglas

AM Aprendizagem de Máquina.

AUC-ROC Area Under the Curve - Receiver Operating Characteristic.

BERT Bidirectional encoder representations from transformers.

BERTimbau Bidirectional Encoder Representations from Transformers para o português brasileiro.

CC Custo Computacional.

CD Ciência de Dados.

CGDF Controladoria-Geral do Distrito Federal.

CIUC Classificadores de Uma Classe.

CRISP-DM Cross Industry Standard Process for Data Mining.

CT Classificação de Textos.

CTD Classificação de Textos Desbalanceados.

CTM Custo de Treinamento de Modelos.

DICID Diretoria de Ciência de Dados.

e-SIC DF Sistema Eletrônico do Serviço de Informações ao Cidadão do Distrito Federal.

EFC Energy-based Flow Classifier.

FDR False Discovery Rate.

GDF Governo do Distrito Federal.

GT Geração de Textos.

IFOREST Isolation Forest.

KNN K-Nearest Neighbors.

LDA Latent Dirichlet Allocation.

LGBM Light Gradient Boosting Machine.

LLM Large Language Model.

LOF Local Outlier Factor.

LR Logistic Regression.

ML Machine Learning.

MPPR Ministério Público do Estado do Paraná.

NIDS Network Intrusion Detection Systems.

NLP Natural Language Processing.

NM NearMiss.

OCSVM One-Class Support Vector Machine.

Ouv-DF Sistema de Ouvidoria do Distrito Federal.

OVA One-Versus-All.

PCA Principal Component Analysis.

PGDF Procuradoria Geral do Distrito Federal.

PLN Processamento de Linguagem Natural.

QLoRA Quantized Low-Rank Adaptation.

RIR Relative Imbalance Ratio.

RoBERTa Robustly Optimized BERT Pretraining Approach.

SVM Support Vector Machine.

TCD Trajetória de Ciência de Dados.

TCE/PI Tribunal de Contas do Estado do Piauí.

TF-IDF Term Frequency-Inverse Document Frequency.

TF-IGM Term Frequency - Inverse Gravity Moment.

Capítulo 1

Introdução

1.1 Contexto Geral

A participação da sociedade na avaliação das políticas públicas é um mecanismo de controle que permite cobrar do governo aquilo a que a sociedade tem direito [2], permite avaliar as políticas públicas implementadas, e possibilita que os cidadãos demandem novas ações governamentais ([3] e [4]). Os canais governamentais de participação popular são uma das mais recentes formas de governança participativa, em um processo democrático direto da sociedade civil [5]. Para garantir a boa comunicação com o cidadão, o governo deve melhorar os portais governamentais oferecendo serviços de qualidade, simplificando o seu acesso e a sua utilização a despeito do seu conhecimento no assunto [6]. Portanto, a administração pública tem buscado incorporar mecanismos inovadores a partir do uso de técnicas de Aprendizagem de Máquina (AM) nas suas atividades, visando assegurar uma melhor interação com a população, aumentando a sua produtividade na entrega dos serviços à sociedade ([7], [8], [3] e [4]).

1.2 Contexto Específico

Com o propósito de melhorar a interação com a população, aumentando a produtividade na entrega dos serviços à sociedade ([7], [8], [3] e [4]), a Controladoria-Geral do Distrito Federal (CGDF)¹ desenvolveu a plataforma Participa DF ², canal único online voltado para o aprimoramento da participação popular nas ações do Governo do Distrito Federal (GDF). A CGDF, órgão especializado da administração direta do GDF ³, em condição de Secretaria de Estado, por ter dentre as suas diversas competências:

¹<https://www.cg.df.gov.br/>

²<https://www.participa.df.gov.br/>

³<https://www.df.gov.br/>

- a supervisão e a coordenação dos serviços das ouvidorias públicas do Distrito Federal;
- a defesa do patrimônio público e da transparência;
- a coordenação de projetos e ações de governança e *compliance*;
- e outras atribuições.

A CGDF administra os seguintes sistemas hospedados pelo Participa DF, sob pre-suposto de oferecer "*serviços de melhor qualidade, mais simples, acessíveis e a um custo menor para o cidadão, os serviços passam a ser disponibilizados pela internet*" [9]:

- Sistema de Ouvidoria do Distrito Federal (Ouv-DF) - Lei nº 4.896 de 31/07/2012 ⁴ e
- Sistema Eletrônico do Serviço de Informações ao Cidadão do Distrito Federal (e-SIC DF) - Lei de Acesso à Informação (LAI)⁵

Nestes sistemas os cidadãos encaminham demandas de Ouvidoria (Ouv-DF) e solicitações de acesso à informação (e-SIC DF). As solicitações submetidas por meio do Participa DF são rotuladas manualmente pelos usuários no momento em que escolhem em qual sistema (Ouv-DF ou e-SIC DF) desejam realizar o registro. Para que os usuários realizem corretamente essa categorização inicial, é necessário que possuam conhecimento prévio sobre a finalidade de cada um dos dois sistemas (Ouv-DF ou e-SIC DF).

Então, estas demandas inicialmente rotuladas pelos usuários são encaminhadas às unidades administrativas responsáveis pela análise dos méritos descritos nos registros. Nesta etapa, é verificado se a rotulagem do cidadão foi efetuada corretamente. Tal verificação é realizada por servidores públicos, os quais podem reclassificar, reatribuir, encerrar ou dar prosseguimento à análise dos pedidos. Este processo demorado pode introduzir variabilidade interpretativa, consequentemente podendo aumentar o tempo necessário para responder às solicitações dos cidadãos [10, 11] — ou resultar no encerramento de demandas submetidas ao sistema errado, uma vez que não é possível transferir pedidos de um sistema para outro.

Automatizar a classificação das demandas dos cidadãos por algoritmos de Aprendizagem de Máquina supervisionada pode: reduzir o tempo gasto pelo usuário na seleção do destino do pedido no Participa DF; reduzir a variabilidade interpretativa; agilizar o processo de encaminhamento das solicitações; reduzir o tempo de resposta da Administração Pública ao cidadão; otimizar a força de trabalho do Estado, liberando os servidores públicos de atividades rotineiras de triagem e permitindo que se concentrem em análises

⁴https://www.sinj.df.gov.br/sinj/Norma/72016/Lei_4896_31_07_2012.html

⁵https://www.sinj.df.gov.br/sinj/Norma/72983/Lei_4990_12_12_2012.h

mais complexas e estratégicas ([4], [3] e [12]); e pode melhorar a experiência do cidadão na plataforma de participação social. Automatizar a classificação das demandas dos cidadãos por algoritmos de Aprendizagem de Máquina é uma forma de alcançar a premissa da Administração Pública que é a entrega de serviços de melhor qualidade, mais simples e acessíveis à população [9], sendo uma boa abordagem para o contexto do Participa DF.

Contudo, as solicitações recebidas pela plataforma Participa DF formam um conjunto de dados altamente desbalanceado e automatizar a rotulação das solicitações recebidas pelo portal demanda maior cuidado do que bases balanceadas. Porque classes minoritárias — como denúncias e pedidos de acesso à informação — que exigem elevada precisão, tendem a ser mal classificadas por modelos tradicionais de AM, embora desbalanceamento em bases de dados reais seja um problema comum [13]. Para aumentar a assertividade da classificação de *corpora* desbalanceados, costuma-se empregar técnicas de redimensionamento como subamostragem (*undersampling*) e sobreamostragem (*oversampling*). Entretanto, estas técnicas apresentam limitações como sobreajustamento (*overfitting*), perda de informações relevantes e baixa generalização [14] e [15].

Os problemas gerados pelas técnicas de redimensionamento advêm da tentativa de mitigar a baixa assertividade dos modelos de classificação sobre base de dados desbalanceados. TSAI e WEI [13] observaram que a classificação de duas classes desbalanceadas pode ser comparada a problemas de detecção de anomalias ou pontos discrepantes. Neste contexto, o algoritmo *Energy-based Flow Classifier (EFC)*, apresenta ótimos resultados em evitar os problemas citados anteriormente. O EFC é um classificador baseado em anomalias que usa estatísticas inversas para inferir um modelo estatístico com base somente em exemplos benignos rotulados, ou seja, metade das informações de um *corpora* balanceado. O algoritmo apresenta grande vantagem sobre outros algoritmos de AM baseados em tarefas de classificação binária para detecção de anomalias, [16] e [17]. O estudo da classificação de demandas raras como anomalias pelo algoritmo EFC, na classificação binária de textos, ainda não foi explorado na administração pública brasileira.

A presente dissertação tem por foco o estudo da melhor forma para classificar as solicitações presentes na base desbalanceada de duas classes da plataforma de participação social Participa DF.

1.3 Problema e Hipótese

1.3.1 Definição do Problema

A classificação binária para detecção de anomalias é uma atividade similar às classificações binárias de *corpora* com alto grau de desbalanceamento, visto que as anomalias

formam uma classe minoritária [13]. Assim como há muitas bases de dados desbalanceadas nos problemas de domínio do mundo real [13], as plataformas de participação popular podem gerar *corpora* desbalanceados que possuem casos especiais [11], como denúncias ou pedidos à Ouvidoria ou acesso à informação, que exigem atenção elevada e precisam ser classificados com grande assertividade. Geralmente os modelos de classificação tendem a ter baixa assertividade na rotulação de exemplares de classes minoritárias, como denúncias e pedidos de acesso à informação, de certa forma ignorando-as, uma vez que aprenderam mais sobre a classe com mais exemplares [18]. Dessa forma, existe uma necessidade de se investigar novos mecanismos de Aprendizagem de Máquina (AM) para aumentar a assertividade na rotulação de exemplares pouco representativos no conjunto de dados.

1.3.2 Hipótese

A técnica de Aprendizagem de Máquina *Energy-based Flow Classifier* ([19], [16], [17]) apresenta uma assertividade maior para a rotulação de exemplares minoritários, classes de baixa representatividade no conjunto de dados, ou seja, desbalanceadas, na métrica *Recall* da classe minoritária.

1.4 Justificativa e Contribuições

1.4.1 Justificativa do tema

O algoritmo *Energy-based Flow Classifier (EFC)* ([19], [16], [17]) é um classificador baseado em anomalias (dados extremamente desbalanceados) que usa estatísticas inversas para inferir um modelo estatístico com base somente em exemplos rotulados como benignos. Precisando somente de metade das informações geralmente necessárias para criar classificadores clássicos de AM, apresentando também um Custo Computacional menor [16]. Assim, o EFC([19], [16], [17]) tem demonstrado grande potencial em tarefas de classificação binária para detecção de anomalias ([16] e [17]) e não ter sido ainda explorado em tarefas de Classificação de Textos (CT) na Administração Pública brasileira.

Relevância acadêmica

Uma vez que o EFC([19], [16], [17]) ainda não foi aplicado em classificação de textos, é relevante a busca por novas técnicas de classificação em Processamento de Linguagem Natural (PLN) que apresentem desempenho melhor que os algoritmos clássicos da área. A busca por técnicas que apresentem custo computacional menor é de grande valia para

o mundo real, no qual a escassez de recursos e presença de bases de dados desbalanceadas é constante [13].

Relevância social

Ao incorporar mecanismos inovadores a partir do uso de técnicas de AM nas suas atividades, a Administração Pública visa aumentar sua produtividade na entrega dos serviços à sociedade ([7], [3] e [4]).

Assim a aplicação de Aprendizagem de Máquina (AM) na página inicial da plataforma Participa DF, pretende assegurar uma melhor interação com a população, desempenhando papel importante em relação à transparência dos atos (princípio fundamental da Administração Pública) [12] e constituindo importante mecanismo para possibilitar o controle social e o acesso democrático aos serviços públicos [8].

1.4.2 Contribuições Esperadas

Ao classificar automaticamente e melhorar os relatos de usuários de plataformas sociais com dados desbalanceados, a solução proveniente do EFC ([19], [16], [17]), facilitaria o uso dos serviços do Participa DF pelo cidadão mais leigo, sem que os mesmos conheçam os pormenores dos trâmites dos serviços presentes na plataforma. Logo, a necessidade em criar modelos que atendam melhor os utilizadores das plataformas sociais com base de dados desbalanceada, permite explorar a aplicabilidade do EFC([19], [16], [17]) na classificação binária de texto de bases desbalanceadas.

Ampliar o uso de Aprendizagem de Máquina (AM) na classificação binária de textos de bases desbalanceadas pode possibilitar a melhora do atendimento ao cidadão e a otimização das atividades da administração pública ao:

- facilitar a escolha do serviço e por conseguinte diminuir o tempo desta atividade;
- reduzir a variabilidade interpretativa das classificações;
- melhorar a destinação correta das solicitações dos cidadãos e por conseguinte diminuir o tempo de trâmite e resposta a elas;
- permitir a alocação desse tempo economizado pelos servidores, na execução de tarefas mais complexas;
- estudar a aplicabilidade do EFC em Classificação de Textos Desbalanceados (CTD).

Desta forma, entregando um serviço melhor à população e garantindo a participação da maior variedade de perfis de usuários.

1.5 Estrutura do Trabalho

Após esta breve introdução sobre o escopo desta dissertação, os demais capítulos estão estruturados da seguinte forma. No Capítulo 2, são apresentados os trabalhos que discorrem sobre o *estado da arte* relacionado à temática deste estudo. No Capítulo 3, são descritos os principais conceitos que fundamentam este trabalho. O Capítulo 4 apresenta a solução proposta para a problemática definida, considerando a literatura discutida nos capítulos anteriores, assim como os experimentos, seguindo a estrutura da metodologia TCD [1]. O Capítulo 5 apresenta os resultados obtidos a partir dos experimentos descritos no capítulo anterior, bem como a análise desses resultados. Por fim, no Capítulo 6, são apresentadas as considerações finais, as conclusões e as perspectivas de trabalhos futuros.

Capítulo 2

Revisão do Estado da Arte

Nesta seção são apresentados, concisamente, os artigos de estudos científicos listados na Tabela 2.1 que podem ser considerados novos padrões de referência (*estado da arte*) em Ciência de Dados (CD) e na temática deste estudo, apresentado no Capítulo 1. As abstrações e soluções encontradas nestes estudos servem como fundamentação teórica para a solução apresentada no Capítulo 4.

Além da fundamentação teórica, neste trabalho utilizou-se técnica de documentação direta de fontes secundárias, por meio de pesquisa bibliográfica e documental, além de técnica de documentação de fontes primárias com pesquisa-ação.

As técnicas de documentação aliadas às buscas por publicações relacionadas ao tema do presente trabalho, utilizaram parte da abordagem da Teoria do Enfoque Meta Analítico – TEMAC ¹ de MARIANO e SANTOS[20].

As pesquisas foram realizadas nas bases:

- *Science Direct* ²
- Google Acadêmico ³
- *SBC Open Lib* ⁴
- *IEEE Xplore* ⁵
- *ACM Digital Library* ⁶

As palavras-chaves e "*strings*" utilizadas foram "*One-class Classifier*", "*imbalance dataset*", "*imbalance dataframe*", "*Natural Language Processing*", "*Text Classification*", "*Class*

¹<https://www.pesquisatemac.com/>

²<https://www.sciencedirect.com/>

³<https://scholar.google.com.br/?hl=pt>

⁴<https://sol.sbc.org.br/index.php/indice>

⁵<https://ieeexplore.ieee.org/Xplore/home.jsp>

⁶<https://dl.acm.org/>

Imbalance", "Imbalanced Text Classification", "Text Generation", "Imbalanced Data" e "Text Public Services Automation".

2.1 Trabalhos Relacionados

Na busca por incorporar mecanismos inovadores em suas atividades e que sejam baseados em Aprendizagem de Máquina (AM), a Administração Pública busca uma melhor interação com a população e aumentar sua produtividade na entrega dos serviços a ela, sociedade ([7], [8], [3] e [4]).

2.1.1 Classificação de Textos Desbalanceados na Administração Pública

NOGUTI, VELLASQUES e OLIVEIRA descrevem em seu artigo [10] o estudo realizado na automatização do processo de classificação das petições da população ao Ministério Público do Estado do Paraná (MPPR), como forma de otimizar os serviços prestados à sociedade. O MPPR recebe petições da população por diferentes meios e precisa encaminhá-las ao setor jurídico adequado, identificando a área do direito relacionada. A utilização de várias abordagens de AM permitiria: a redução da variabilidade interpretativa dos casos (geradas pela análise dos atendentes), a diminuição do tempo gasto pelo usuário na seleção da área jurídica no sistema, e por fim, permitiria alocar o tempo gasto pelos atendentes nas classificações das petições para tarefas mais complexas. Apesar do experimento ter alcançado acurácia de 90%, superior à acurácia humana de 72%, e *F1-Score* de 85% na classificação em 18 categorias de áreas do direito, os pesquisadores descobriram que a aplicação de técnicas de balanceamento aleatório para equalizar as classes desbalanceadas, resultou em desempenho inferior com queda no *F1-Score* para todos os classificadores avaliados, possivelmente pela redução do tamanho do *corpus* de treinamento pela subamostragem aplicada.

Ainda sobre Classificação de Textos Desbalanceados (CTD), SOUZA em sua dissertação [21] também relata a dificuldade em rotular de forma eficiente base de textos desbalanceadas no contexto da Administração Pública. A referida dissertação tem como foco a investigação sobre o emprego de técnicas de AM para criar um mecanismo automatizado de classificação de documentos jurídicos registrados na Procuradoria Geral do Distrito Federal (PGDF), e se este emprego de técnicas de AM apresentaria maior eficiência do que o trabalho manual já realizado na classificação de documentos recebidos pela PGDF. A criação de tal mecanismo para utilização da PGDF impactaria na redução de tarefas repetitivas e ampliaria a quantidade de dados para serem usados em análises estratégicas,

aumentando a produtividade na entrega de serviços à sociedade. Por trabalhar com uma base desbalanceada, o pesquisador constatou observando outros trabalhos, a dificuldade em melhorar o desempenho em classificações com base de dados desbalanceados. Assim utilizou e comparou várias técnicas de pré-processamento e classificação de bases de dados desbalanceados, utilizando algoritmos de aprendizado não supervisionado, como *K-means* e *Latent Dirichlet Allocation (LDA)*; algoritmos de aprendizado supervisionado, como *Ridge Classifier*, *Perceptron* e *Random Forest*; além da abordagem *One-Versus-All (OVA)* para lidar com o desbalanceamento. O trabalho demonstrou a viabilidade de um classificador automatizado, capaz de rotular processos rapidamente, mas enfrentou limitações relacionadas à criação de classificadores sem ter dados previamente rotulados e enfrentou dificuldades com o tratamento de grandes volumes de dados desbalanceados. O estudo observou aspectos que influenciam na geração de modelos com bom desempenho, como o nível de desbalanceamento da base de dados e a qualidade dos dados para treinamento de modelos (que reflete no custo computacional). Também fez observações sobre aspectos práticos importantes na implementação das soluções desses modelos, como modelos com desempenho parecido, mas com grande diferença no tempo de treinamento. Observações relevantes para contextos com limitações computacionais.

Continuando sob a temática da Classificação de Textos Desbalanceados (CTD) e na busca por empregar AM para aumentar a produtividade dos serviços entregues à sociedade, o estudo de CORDEIRO, RABELO E MOURA [11] comparou técnicas de algoritmos clássicos de AM, modelos *Support Vector Machine (SVM)* com representação de combinações de modelos de AM e modelos de aprendizado profundo, *Bidirectional encoder representations from transformers (BERT)*, para rotular manifestações e comunicados de irregularidades recebidos pela Ouvidoria do Tribunal de Contas do Estado do Piauí (TCE/PI) em uma base de textos desbalanceada. O trabalho objetivou melhorar a triagem dos comunicados, acelerando as respostas e reduzindo o custo de classificação. Para tratar o desbalanceamento foram aplicadas, técnicas de redimensionamento, como sobreamostragem para as classes minoritárias, utilizando a biblioteca NLPAUG ⁷ e o modelo BERTimbau ⁸ e subamostragem para a classe majoritária. Considerando o *F1-Score* como a melhor métrica para avaliar conjuntos desbalanceados, os pesquisadores obtiveram melhor desempenho com o modelo BERTimbau para conjunto de dados redimensionado, atingindo 96% nas métricas de *F1-Score*, *Recall* e Acurácia. Com o conjunto sem redimensionamento, o SVM apresentou os melhores resultados, com acurácia de 84%, *Recall* de 84% e *F1-Score* de 83%. O redimensionamento dos dados resultou em um ganho de desempenho na classificação dos comunicados de irregularidades. Considerando que

⁷<https://nlpaug.readthedocs.io/en/latest/>

⁸<https://huggingface.co/neuralmind/bert-base-portuguese-cased>

o artigo concluiu que o algoritmo de redes neurais obteve o melhor desempenho dentre os demais, talvez fosse interessante que o estudo fizesse considerações sobre o Custo de Treinamento de Modelos (CTM) para gerar os códigos, caso os mesmos precisassem ser retreinados com mais frequência.

2.1.2 Classificadores de Uma Classe

Ampliando a discussão de bases compostas por texto com classes desbalanceadas para as compostas por outros tipos de dados, TSAI e WEI [13] enfatizaram como é corriqueiro encontrar bases de dados desbalanceados no mundo real e por este motivo a Aprendizagem de Máquina (AM) para criação de classificadores para estas bases é um importante problema a ser estudado em Classificadores de Uma Classe (CIUC) (traduzido do inglês, *One Class Classifier*). Os autores apresentaram um estudo empírico sobre o emprego de seleção de atributos e o cascadeamento de modelos nos classificadores de uma classe para base de dados desbalanceada de duas classes. No estudo esclarecem que em base de dados com classes desbalanceadas, a classe minoritária é aquela que apresenta um tamanho menor que as demais por não apresentar exemplares suficientes, causando um desbalanceamento da distribuição das classes. Ressaltam que em problemas que envolvam desbalanceamento, as classes minoritárias comumente são mal rotuladas pelos classificadores automáticos de AM em contraposição às classes majoritárias. Situação frequente no problema de detecção de anomalias ou pontos discrepantes (que tratam da classificação de bases desbalanceadas com duas classes), que trivialmente são abordados por Classificadores de Uma Classe (CIUC), segundo a pesquisa. Os pesquisadores utilizaram três classificadores de classe única: *One-Class Support Vector Machine (OCSVM)*, *Isolation Forest (IFOREST)* e *Local Outlier Factor (LOF)*, além de combinações deles, sempre comparando com o algoritmo de referência C4.5. A pesquisa utilizou a área sob a curva ROC (AUC-ROC) como métrica e concluiu que os CIUC são especialmente eficazes em conjuntos de dados com alta proporção de desequilíbrio entre as duas classes. E que as diferentes combinações de CIUCs, com ou sem seleção de características, mostrou-se promissora para melhorar o desempenho em cenários com alto desequilíbrio.

Além dos algoritmos CIUC apresentados anteriormente, ressaltamos o algoritmo apresentado por PONTES *et al* [16], o *Energy-based Flow Classifier (EFC)* ([19], [16], [17]). O EFC⁹ foi projetado para classificação baseada em anomalias, para atuar como sistema de detecção de intrusão de rede baseado em fluxo, classificando tráfego em tempo real. O EFC utiliza estatística inversa para inferir um modelo estatístico a partir de exemplos rotulados como benignos. Os autores consideraram o algoritmo como adequado para sistemas de detecção de intrusão de rede (do inglês, *Network Intrusion Detection Systems*

⁹<https://efc-package.readthedocs.io/en/latest/index.html>

(*NIDS*)), e demonstraram que o EFC obteve desempenho comparado aos classificadores tradicionais de AM. O estudo de PONTES *et al* [16] também avaliou o desempenho dos classificadores quando treinados em um domínio e testados em outro relacionado, demonstrando que o EFC se adapta melhor a diferentes distribuições de dados em comparação com algoritmos como KNN, árvore de decisão e *perceptron* multicamadas. Possuindo ainda, o EFC, um CC menor do que algoritmos que usam duas classes, por usar metade dos dados para treinamento. Desta forma, o EFC alcançou *F1-Score* de aproximadamente 97% e AUC-ROC de cerca de 99%, resultados similares aos de outros classificadores clássicos. Por apresentar um CC menor, o algoritmo aproxima-se da atual tendência em reduzir a "*pegada ecológica*" no treinamento dos modelos de AM.

2.1.3 Redução de Dimensionalidade

Além da abordagem dos algoritmos CIUC para melhorar a classificação de bases desbalanceadas, outras formas de pré-processamento podem ser empregadas neste contexto, para obter mecanismos baseados em Aprendizagem de Máquina (AM) que auxiliem a Administração Pública na busca por uma melhor interação com a população e aumento da produtividade na entrega dos serviços à sociedade ([7], [8], [3] e [4]).

No artigo "*Multi-class text classification based in oversampling for highly imbalanced dataset*" [22] foi apresentada uma arquitetura de classificação multiclasse projetada para rotular conjuntos de dados textuais altamente desbalanceados e de alta dimensionalidade. A arquitetura proposta abrange etapas de pré-processamento, balanceamento de dados, engenharia de atributos, redução de dimensionalidade e aplicação de algoritmos de classificação. A motivação do estudo está na automatização de processos de classificação de documentos em diversas instituições — como órgãos governamentais —, nas quais a classificação manual se torna impraticável devido ao volume cada vez maior de documentos recebidos. Tais conjuntos de dados frequentemente apresentam desafios como desbalanceamento entre classes e alta dimensionalidade, os quais dificultam o processamento algorítmico e o desenvolvimento de classificadores robustos, exigindo, portanto, a adoção de técnicas de pré-processamento e de redimensionamento. Segundo os autores, a abordagem proposta mostrou-se eficaz por meio da combinação de técnicas de pré-processamento, como lematização e estemização, em conjunto com redução de dimensionalidade pela Análise de Componentes Principais, *Principal Component Analysis* (PCA) [23], e métodos de sobreamostragem, como o *Synthetic Minority Oversampling Technique* (SMOTE) e o *Random Over-Sampling* (ROS). Os resultados evidenciaram bom desempenho em métricas como *F1-Score* e *Recall*, com Acurácia variando entre 77% e 99%, a depender do conjunto de dados utilizado.

2.1.4 Custo de Treinamento e Desempenho

Apesar das inúmeras estratégias para abordar tarefas de Classificação de Textos Desbalanceados (CTD), deve-se observar a relação Custo de Treinamento de Modelos (CTM) e Desempenho, principalmente em ambiente com recursos escassos. O artigo de REUSENS *et al* [24] aborda, entre outros pontos, sobre o Custo de Treinamento de Modelos no treinamento de modelos de Classificação de Textos (CT) e os desdobramentos destes treinos, a citar o elevado consumo de energia e a "pegada ecológica" deixada pela atividade. O estudo apresenta um *benchmark* para o treinamento de modelos de Classificação de Textos (CT), investigando se os algoritmos de ponta, mais avançados e complexos ("estado da arte" em tarefas de Processamento de Linguagem Natural (PLN) são sempre superiores (em desempenho e outros quesitos) aos algoritmos mais antigos e simples de AM, clássicos. Os pesquisadores realizaram mais de 42.800 testes com 11 arquiteturas de diferentes complexidades como LR, SVM, *Random Forest*, XGBoost, BiLSTM e RoBERTa. Os autores destacam que métodos simples podem competir com modelos complexos em tarefas de classificação em Processamento de Linguagem Natural como detecção de *fake news*, classificação de tópicos, detecção de emoções, polaridade e sarcasmo, tanto em desempenho como em CTM. No estudo, o algoritmo BiLSTM apesar de ter se destacado na CT, não apresentou diferenças significativas em desempenho em comparação a abordagens mais simples como LR+TF-IDF e RoBERTa. Observaram que alguns algoritmos simples de AM obtiveram melhor desempenho ou ganho computacional que algoritmos considerados "estado da arte". Assim, o artigo concluiu que o esforço computacional para treinar modelos de aprendizado profundo nem sempre é justificável, ressaltando que a escolha de métodos deve considerar a complexidade do problema a solucionar e a explicabilidade dos modelos.

2.1.5 Geração de Textos Sintéticos

O estudo [25] conduzido por JOSHI *et al* apresenta a biblioteca Python NLPAUG¹⁰ que apresenta funcionalidades para geração de texto sintéticos (*text data augmentation*) usando no trabalho de CORDEIRO, RABELO E MOURA [11]. Nele relatam sobre o impacto do aumento de dados textuais para contornar a escassez de conjuntos rotulados para o treinamento de modelo, enfatizando a aplicação LLMs e *transformers* pré-treinados para gerar conteúdos sintéticos. A pesquisa destaca que os LLMs criam textos prevendo sequências lógicas de palavras, viabilizando a geração de amostras textuais artificiais altamente informativas para o treinamento de classificadores. Aumentando a precisão, a robustez contra ruídos e a capacidade de generalização dos modelos criados para as mais

¹⁰<https://nlpaug.readthedocs.io/en/latest/>

variadas tarefas de Processamento de Linguagem Natural (PLN). Contudo, os autores alertam que o uso dessas arquiteturas generativas exige um alto CC, além do risco de produzir textos sem sentido lógico e com a introdução de vieses nocivos para o modelo a ser treinado. Em treinamentos em que foram utilizados 50% dos dados originais da base real e 50% de textos gerados, obteve-se acurácia equivalente ao treinamento com 100% dos textos originais da base real.

2.1.6 Métricas para Classificação de Textos

Dentre as diversas abordagens apresentadas anteriormente, para comparar qual apresentaria melhor desempenho no contexto desta dissertação é necessário escolher a métrica que melhor avaliaria a Classificação de Textos Desbalanceados (CTD).

O artigo de SOKOLOVA E LAPALME [26] realizou uma análise sistemática de 24 medidas de desempenho usadas em tarefas de classificação em Aprendizagem de Máquina (AM), abrangendo classificações binária, multiclasse, multirrótulo e hierárquica. O estudo relacionou mudanças na matriz de confusão e características específicas dos dados, destacando que a invariância das medidas em relação a essas transformações é crucial, pois conecta as métricas de avaliação a cenários específicos de aprendizado. O estudo focou em como as características dos dados afetam a objetividade das medidas, em vez de comparar empiricamente o desempenho dos algoritmos. O artigo [26] também apresenta métricas utilizadas em Processamento de Linguagem Natural (PLN), no subcampo de Classificação de Textos (CT), assim como tarefas de classificação binária. Apontou que as métricas empregadas nestas tarefas, comumente usadas na CT, têm sua origem na área de "*Extração de Informação*" e focam principalmente na identificação eficaz de exemplos positivos (Precisão, *Recall*, *F1-Score*). Para a classe negativa, a AUC-ROC avaliaria melhor que as medidas *FScore*. Enfatizando a importância dada à identificação precisa de instâncias relevantes em tarefas de classificação textual.

No estudo de CUNHA *et al* intitulado "*On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study*" [27], os autores realizaram uma análise crítica de artigos científicos que abordam métodos e representações neurais e não neurais para classificação automática de textos. Eles conduziram uma comparação abrangente entre os trabalhos, motivada pela identificação de problemas relacionados ao rigor científico em grande parte da literatura recente, tais como o uso de protocolos experimentais inadequados, a ausência de tratamento estatístico, a falta de detalhamento sobre o ajuste de hiperparâmetros, bem como a não consideração da relação entre a efetividade da classificação e os custos de treinamento dos modelos de Classificação de Textos. O trabalho destaca que, embora arquiteturas baseadas em *transformers* apresentem maior efetividade em conjuntos de dados de grande

escala quando comparadas às melhores soluções clássicas não neurais, como TF-IDF + SVM, os ganhos obtidos pelos modelos neurais são relativamente modestos (em média, cerca de 5%) e com tempos de treinamento até 23 vezes maiores, tornando-os consideravelmente mais lentos para o retreino. Ao questionar o real custo-benefício dessas soluções, os resultados revelaram que abordagens tradicionais, quando devidamente ajustadas, oferecem uma melhor relação custo-benefício, especialmente em conjuntos de dados de menor porte. Os autores concluem que outras pesquisas futuras devem adotar maior rigor metodológico na apresentação dos resultados, incluindo a aplicação de testes estatísticos e a justificativa explícita dos métodos e das *baselines* escolhidas.

2.1.7 Metodologias

E após observar as diversas abordagens possíveis para melhorar a Classificação de Textos Desbalanceados (CTD), é preciso escolher a metodologia para desenvolvimento deste estudo.

No artigo de MARTÍNEZ-PLUMED *et al* [1], os pesquisadores revisitaram a metodologia *Cross Industry Standard Process for Data Mining (CRISP-DM)* para investigar sua relevância no contexto atual, após mais de 20 anos da sua criação. Eles avaliaram se a metodologia ainda é aplicável a projetos de Ciência de Dados (CD) contemporâneos e propuseram uma nova sistemática, a Trajetória de Ciência de Dados (TCD). Baseada no CRISP-DM, a TCD foca em projetos de Ciência de Dados (CD) orientados por objetivos, processos exploratórios ou de gestão de dados. O estudo identificou limitações no CRISP-DM original e em outras metodologias, especialmente em relação ao tratamento da diversidade de projetos de CD modernos. Também reconheceu a necessidade de atividades exploratórias, como a exploração de objetivos e fontes de dados, que não são abordadas pelo CRISP-DM, levando à criação de um modelo mais flexível e abrangente. O artigo mostrou que atividades de CD realizadas nos casos estudados foram bem representadas nas três áreas metodológicas classificadas pelos autores: CRISP-DM, exploratória e gerenciamento de dados. O *framework* Trajetória de Ciência de Dados por ser mais voltado para atividade de Ciência de Dados, apresenta uma abordagem diferente da metodologia tradicional CRISP-DM, que é mais voltada para Mineração de Dados. Na TCD, metodologia utilizada nesta dissertação, é possível escolher a ordem da realização das atividades desenvolvidas em um projeto de Ciência de Dados (CD) e alterá-las a depender das decisões e descobertas do cientista de dados, o que permite flexibilidade ao trabalho desenvolvido nesta dissertação.

2.2 Considerações

Os trabalhos mencionados neste capítulo, serviram de base para nortear as atividades de Aprendizagem de Máquina (AM) planejadas para solucionar a problemática citada no Capítulo 1. Eles estão elencados na Tabela 2.1. As características presentes nos referidos artigos geraram as colunas da Tabela 2.1, a saber:

- Classificação de Textos Desbalanceados (CTD), provenientes de bases em que suas classes não possuem a mesma proporção de registros;
- Classificadores de Uma Classe (CIUC), algoritmos que treinam modelos utilizando como entrada somente uma classe;
- Redução de Dimensionalidade;
- Custo de Treinamento de Modelos (CTM);
- Geração de Textos (GT), de forma sintética, artificial, para balancear classes desproporcionais;
- Métricas para Classificação de Textos (CT)
- Metodologia, com benefícios mais significativos no desenvolvimento de trabalhos em Ciência de Dados (CD).

Tabela 2.1: Trabalhos Relacionados

Artigos	Classificação Textos Desbalanceados	Classificadores Uma Classe	Redução de Dimensionalidade	Custo Treinamento Modelos	Geração de Textos	Métricas	Método
[10]	X						
[21]	X						
[11]	X				X		
[13]		X					
[16]		X					
[22]	X		X				
[24]	X			X			
[25]	X				X		
[26]	X					X	
[27]						X	
[1]							X
Esta disserta- ção	X	X	X	X	X	X	X

Sobre o problema do tratamento do desbalanceamento de base de textos (representado na coluna Classificação de Textos Desbalanceados), as investigações presentes nos artigos de NOGUTI, VELLASQUES E OLIVEIRA [10]; SOUZA [21]; CORDEIRO, RABELO E MOURA [11]; REUSENS *et al* [24]; JOSHI *et al* [25]; SOKOLOVA E LAPALME

[26]; e SANTOS *et al* [22] apresentaram as dificuldades encontradas e as mais variadas abordagens adotadas frente ao caso, tendo atuações que miraram do pré-processamento à escolha do tipo de algoritmo.

Cabe ressaltar que os trabalhos de TSAI e LIN [13] e de PONTES [16] trataram sobre desbalanceamento de classes sob a ótica dos Classificadores de Uma Classe, sem abordar a Classificação de Textos Desbalanceados. Contudo, os estudos poderiam ser aplicados no contexto de CTD.

Outra atuação no tratamento do desbalanceamento de base de textos, seria pela Geração de Textos de forma sintética citado nos artigos de JOSHI *et al* [25]; e CORDEIRO, RABELO E MOURA [11].

Contudo, para avaliar a aplicabilidade dos procedimentos acima referidos para o contexto do desta dissertação, ou seja a plataforma ParticipaDF, é necessário avaliar o Custo de Treinamento de Modelos para criar os modelos neste cenário. Escolher o algoritmo que melhor atende aos anseios e as limitações do estudo. Os artigos de SOUZA [21]; CORDEIRO, RABELO E MOURA [11]; CUNHA *et al* [27]; e REUSENS *et al* [24]; discorrem sobre este aspecto em seus estudos.

Por fim, todo estudo precisa de planejamento e metodologia, assim o trabalho de MARTÍNEZ-PLUMED *et al* [1] faz uma revisão da metodologia de mineração de dados consagrada, o CRISP-DM. E questiona sobre a sua aplicabilidade para o contexto da Ciência de Dados.

Capítulo 3

Fundamentação Teórica

O presente capítulo trata dos principais conceitos relativos ao Processamento de Linguagem Natural (PLN) e Classificação de Textos Desbalanceados (CTD). Ele inicia com a metodologia escolhida Trajetória de Ciência de Dados (TCD), utilizada para determinar as etapas do projeto em Ciência de Dados. Depois, são apresentados os conceitos básicos sobre Classificação de Textos (CT), Balanceamento de Dados, Redução de Dimensionalidade, também são citados os modelos utilizados neste estudo e as principais métricas aplicadas para avaliar o desempenho dos modelos CT criados.

3.1 Trajetória de Ciência de Dados - TCDs

Para o desenvolvimento deste estudo em Ciência de Dados (CD) definiu-se a metodologia Trajetória de Ciência de Dados (TCD) como a mais adequada para resolver o problema apresentado na Capítulo 1 e considerando que ao longo do desenvolvimento do estudo, novos achados poderiam exigir a execução de atividades não previstas inicialmente, assim como na atual dinâmica dos projetos de Ciência de Dados.

A TCD foi proposta por MARTÍNEZ-PLUMED *et al* [1] (Figura 3.2), e para além de englobar as atividades presentes na relevante metodologia *Cross Industry Standard Process for Data Mining (CRISP-DM)* (Figura 3.1), que foca mais a mineração de dados, a TCD engloba outras atividades que focam a exploração dos dados, permitindo selecionar quais tarefas serão executadas e a ordem em que essas atividades ocorrerão em um projeto de CD, um caminho flexível através de um mapa de atividades.

As etapas e atividades da TCD são divididas em três categorias principais:

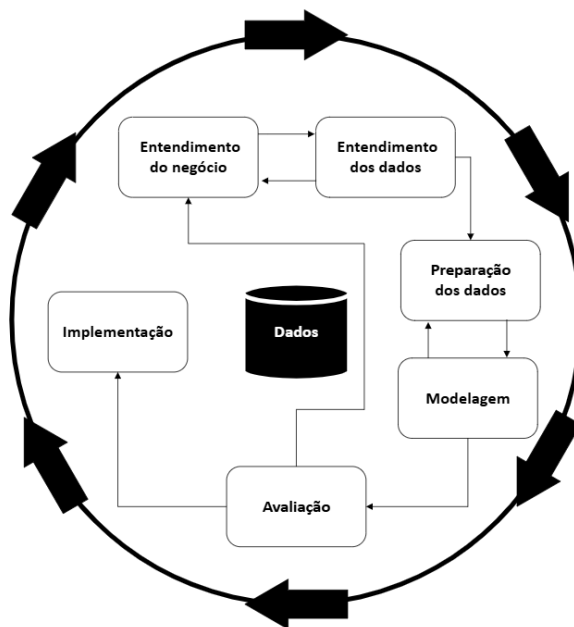


Figura 3.1: Diagrama do CRISP-DM. Fonte: adaptação do artigo de MARTÍNEZ-PLUMED *et al.*[1] .

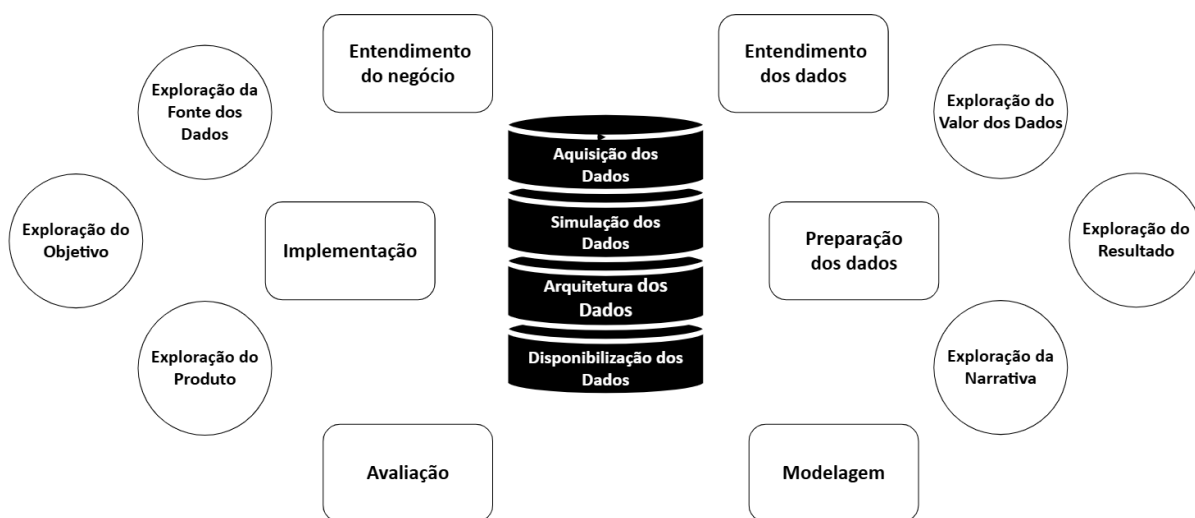


Figura 3.2: Diagrama com os componentes possíveis das Trajetórias de Ciência de Dados. Fonte: adaptação do artigo de MARTÍNEZ-PLUMED *et al.*[1].

3.1.1 Atividades Exploratórias

Investigativas, estas atividades são fundamentais atuando como um processo de "*prospecção*", incluem:

- ***Exploração do Objetivo*** - são definidas as metas de negócios a ser solucionadas de forma orientada à dados;
- ***Exploração da Fontes dos Dados*** - são descobertas novas e valiosas fontes de informação;
- ***Exploração do Valor dos Dados*** - são investigados a qual utilidade que pode ser extraída dos dados que se tem em mãos;
- ***Exploração do Resultado*** - são conectados os resultados encontrados aos objetivos de negócios da organização;
- ***Exploração da Narrativa*** - são extraídas e construídas as histórias valiosas e compreensíveis a partir dos dados;
- ***Exploração do Produto*** - são identificadas as formas de transformar o conhecimento extraído em serviço ou aplicativo útil para clientes e usuários.

3.1.2 Atividades de Gerenciamento de Dados

Estas atividades lidam com os dados como o produto de valor, exigindo massivo trabalho de curadoria e infraestrutura:

- ***Aquisição dos Dados*** - obtenção ou geração de dados relevantes;
- ***Simulação dos Dados*** - simulação de sistemas complexos para produzir dados úteis e permitir análises de causa e efeito;
- ***Arquitetura de Dados*** - projeto do esquema físico e lógico dos dados e realização da integração de múltiplas fontes de dados;
- ***Disponibilização dos Dados*** - disponibilização dos dados finais obtidos de forma estruturada por meio de interfaces, bancos de dados e visualizações.

3.1.3 Atividades Direcionadas a Objetivos (CRISP-DM)

Correspondem ao ciclo tradicional do CRISP-DM, que envolve o Entendimento do Negócio, o Entendimento dos Dados, a Preparação dos Dados, a Modelagem, a Avaliação e a Implantação [28].

Entendimento do Negócio

Foco em entender os objetivos e requisitos do projeto sob a perspectiva do negócio e convertendo-os em definição de problema.

Sendo as principais atividades: a determinação dos objetivos de negócio e dos critérios de sucesso; a avaliação da situação atual (os recursos, os requisitos, as restrições, entre outros); e determinação das metas específicas de CD, com a produção de plano de projeto.

Entendimento dos Dados

Busca pela familiarização com a base de dados. Serão realizadas atividades de: coleta inicial dos dados; descrição e exploração dos dados para formular *insights* e hipóteses; verificação da qualidade dos dados e identificação dos possíveis problemas.

Preparação dos Dados

Consiste na construção (a partir dos dados brutos) do conjunto de dados que alimentará as modelagem. Ocorre: a seleção das tabelas, registros e atributos relevantes; a limpeza dos dados e criação de atributos derivados; e a integração dos dados.

Modelagem

Fase onde ocorre a aplicação dos algoritmos para treinamento dos modelos. Incluem: a seleção das técnicas de modelagem mais adequadas para o problema; a definição do *design* dos testes para validação do modelo; o treinamentos dos modelos e as configurações de hiperparâmetros; a avaliação inicial do modelo treinado.

Avaliação

Neste momento o modelo é submetido a uma avaliação minuciosa para verificar se resolve o problema de negócio. Possui as seguintes atividades: avaliação dos resultados estritamente em relação aos critérios de sucesso do negócio definidos; revisão do processo de treinamento do modelo para certificar que nenhuma questão de negócio crítica foi esquecida; e determinação dos próximos passos e sobre o uso dos resultados.

Implantação

O conhecimento obtido precisa ser organizado para consumo pelo cliente. As atividades podem ser simples ou complexas e incluem: o planejamento da implantação e o monitoramento e a manutenção do modelo em produção; a produção do relatório final e a apresentação dos achados; e a revisão da documentação do projeto para aprendizado futuro.

3.1.4 Trajetória Escolhida

Deste modo a TCD (Capítulo 2) por ser flexível em comparação com a metodologia CRISP-DM [28], permite na montagem do método pelo Cientista de Dados, a escolha das atividades, a ordem delas, a repetição ou execução em paralelo, para melhor "*atacar*" a situação estudada. Dependendo exclusivamente do domínio de aplicação e das descobertas feitas pela equipe de CD.

3.2 Processamento de Linguagem Natural - PLN

O Processamento de Linguagem Natural (PLN) é uma disciplina do campo da Ciência da Computação, desenvolvida a partir da década de 1950 sendo a interseção entre a inteligência artificial e a linguística. O propósito desta área é o desenvolvimento de sistemas computacionais capazes de analisar, reconhecer e processar a linguagem humana de maneira eficiente. Em seus primórdios o PLN dependia de abordagens simplistas e de regras gramaticais codificadas manualmente, abordagens estritamente simbólicas que falhavam ao lidar com a vasta gama de estruturas gramaticais das linguagens naturais. E evoluiu para sistemas que conseguem extrair o significado real (a semântica) e o contexto de sentenças da linguagem natural. Para superar o fato que a linguagem humana possui uma natureza flexível frequentemente ambígua o PLN moderno passou a empregar intensamente modelos estatísticos de AM capazes de inferir as sentenças e o sentido delas baseando-se em probabilidades, fato que viabilizou aplicações como a extração de informações, sistemas complexos de perguntas e respostas e geração de textos sintéticos [29].

3.2.1 Classificação de Textos

A Classificação de Textos (CT) no contexto da AM é uma tarefa fundamental em PLN que envolve atribuir automaticamente classes a documentos textuais a partir de rótulos predefinidos [30] ao passar por um algoritmo [27].

3.2.2 Classificação Binária

O artigo de SOKOLOVA e LAPALME [26] define a classificação binária como a tarefa em que um documento é rotulado em uma, e apenas uma, de duas classes existentes e que não se sobrepõem. Afirma que este tipo de classificação pode ser avaliada por uma matriz de confusão, que computa a quantidade de exemplares os corretamente atribuídos (verdadeiros positivos e verdadeiros negativos) e incorretamente reconhecidos (falsos positivos e falsos negativos).

3.3 Pré-processamento de Textos

O pré-processamento de textos é uma etapa fundamental na mineração de textos, porque diminui os vieses nos modelos de classificação e melhorar a base de textos que será processada pelos algoritmos de Aprendizagem de Máquina (AM) [22]. Esta fase de limpeza e transformação dos textos é estritamente necessária para que os textos possam ser convertidos em números, permitindo que a linguagem natural seja estruturada em formatos matemáticos que efetivamente sejam processados pelos algoritmos de AM, [21].

O pré-processamento de textos é constituído por tarefas para "*limpeza do texto*", removendo elementos considerados ruidosos (*stopwords*, pontuações, entre outros) e aplicando segmentação do texto, que visa dividir o texto em partes menores [31].

Geralmente esta etapa inicia-se com tarefas de remoção de caracteres especiais, sinais de pontuação e números do *corpus* [22]. Em seguida, ocorre a conversão das palavras para letras minúsculas, acompanhada da exclusão das *stopwords* (palavras irrelevantes) e normalização das palavras, [22] e [10]. Após, os textos são simplificados e segmentados em unidades menores (*tokens*), para a posterior a extração de características na vetorização, [31] e [10].

3.3.1 Normalização Textual

A normalização de textos é uma das tarefas da etapa de Pré-processamento de Textos que visa diminuir a variabilidade por flexão das palavras considerando que o português é uma língua altamente flexionada [10]. Para aplicar a Normalização Textual, existem as técnicas estematização e lematização disponíveis em bibliotecas conhecidas de AM como *spaCy*¹ e NLTK²[22].

Lematização

A lematização é uma técnica de normalização que almeja reduzir as distorções vocabulares criadas pelas inflexões gramaticais muito presentes na língua portuguesa [10]. Esta técnica de normalização procura extrair a forma base (o lema) apoiando-se em dicionários [10]. O processo da lematização garante que a palavra normalizada permaneça sendo uma entidade gramaticalmente válida e existente no idioma [24].

Estematização

A estematização é um processo focado em lidar com as variações linguísticas das palavras, onde as diversas flexões das palavras em torno de uma raiz são mapeadas e as palavras são

¹<https://spacy.io/>

²<https://www.nltk.org/index.html>

forçadamente convertidas para sua raiz[32]. O intuito desta técnica é obter o radical (*stem*) das palavras derivadas, partindo do princípio de que as palavras derivadas compartilham fortes semelhanças com sua origem, raiz (*stem*) [31]. O algoritmo de estematização remove os afixos (sufixos e prefixos) das palavras de forma abrupta para obter as raízes, gerando muitas vezes sequencias de letras que não correspondem às palavras existentes [22].

3.3.2 Vetorização de Textos

A Vetorização de Textos consiste na transformação dos dados textuais brutos e não estruturados (linguagem natural) para um formato numérico estruturado, gerando representações matemáticas na forma de matrizes ou vetores de valores numéricos, [33]. Em essência, esse procedimento reduz sentenças de comprimentos arbitrários a listas e sequências de números com tamanhos predefinidos [34].

Esta conversão dos textos em formato numérico é fundamental para os algoritmos de AM, porque não possuem a capacidade de interpretar, analisar ou processar textos puros diretamente sem a Vetorização de Textos [33]. A vetorização viabiliza a aplicação de operações e propriedades de álgebra linear sobre as informações textuais, [34]. Dessa forma, fornecendo entradas numéricas exatas de que as redes neurais e os modelos clássicos de AM necessitam para manipular os dados, calcular métricas como distância Euclidiana ou similaridade por cosseno, descobrir o grau de relacionamento entre palavras e extrair os padrões necessários para realizar tarefas como classificação, agrupamento e recomendações.

TF-IDF

O *Term Frequency-Inverse Document Frequency (TF-IDF)* (Frequência do Termo-Frequência Inversa do Documento) é um método estatístico de vetorização que atribui pesos matemáticos aos termos (*tokens*) com base no seu padrão de distribuição em documentos e em toda a coleção de documento (*corpus*) [22]. Onde TF é a frequência do termo no documento e o IDF a frequência do termo no total dos documentos. O IDF é calculado utilizando somente os dados de treinamento [27]. Esta fórmula inibe a dominância de palavras muito comuns nos *corpus*, destacando as palavras menos comuns e os termos que são verdadeiramente únicos e representativos para o modelo no conjunto de textos de treinamento, [22], [27].

Word Embedding

O *Word embedding* é uma modelagem para redes neurais que substitui as abordagens tradicionais ao representar as palavras como vetores numéricos contínuos projetados em

um espaço de alta dimensionalidade ao invés de matrizes dispersas baseadas na contagem de frequência de termos, [35], [36]. A arquitetura *Word embedding* avalia coocorrências e contextos para encapsular conexões posicionais e de proximidade, assim, baseando-se nessas representações numéricas, os modelos conseguem absorver de forma superior as complexas relações e analogias semânticas do texto, [35], [36]. Esse detalhado mapeamento de significados através de vetores otimiza os cálculos de similaridade e confere um altíssimo poder de generalização. Essas representações densas consolidaram-se como a espinha dorsal da moderna arquitetura *Transformers*, [35], [36].

3.4 Balanceamento de Dados - Redimensionamento

3.4.1 Subamostragem

A subamostragem (*undersampling*) é a exclusão de registros das classes majoritárias com o objetivo de evitar o enviesamento do aprendizado dos modelos durante o treinamento [11]. Esta atividade pode ser efetuada por diversas formas. O *Random Under-Sampling* (RUS), método simples e muito utilizado, elimina os exemplares da classe majoritária de forma aleatória, tendo como principal desvantagem o risco de eliminar dados potencialmente importante para o processo de aprendizagem durante o treinamento do modelo [21]. Assim, outras técnicas de subamostragem mais refinadas podem ser utilizadas para persistir a integridade estatística da classe majoritária enquanto minimiza a sua quantidade de registros como *Cluster Centroids*, *Tomek Links* ou *NearMiss* ³ [37].

3.4.2 Sobreamostragem

Segundo SOUZA [21], a sobreamostragem (*oversampling*) visa eliminar os danos de uma distribuição distorcida por meio da criação de novas amostras para as classes minoritárias, por meio, por exemplo da replicação de alguns exemplos da classe minoritária, ou por criação de exemplos sintéticos a partir das instâncias originais, como faz o método SMOTE (*Synthetic Minority Oversampling Technique*). No artigo "*Imbalanced Data Problem in Machine Learning: A Review*" [37] os autores explicam o SMOTE é um método de sobreamostragem que identifica instâncias próximas dentro do espaço de características, estabelecendo conexões entre elas e gerando novas amostras sintéticas ao longo dessas conexões.

³https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.NearMiss.html

Geração de Textos Sintéticos

Ainda a respeito da sobreamostragem, CORDEIRO, RABELO e MOURA [11] salientam que o aumento de dados focado na criação de amostras artificiais para as classes minoritárias, tem o intuito de balancear a base desbalanceada antes do treinamento dos modelos. Naquele estudo os autores utilizaram a biblioteca Python *NLPAUG* descrita no estudo [25] de JOSHI *et al.*

De acordo com o artigo, o *framework NLPAUG* disponibiliza uma ampla variedade de técnicas para a geração de textos sintéticos e para a realização de aumento de dados textuais (*text data augmentation*), que contribui para a melhoria do desempenho e da capacidade de generalização dos modelos para Classificação de Textos Desbalanceados (CTD), permitindo que aprendam a partir de uma gama mais diversificada de exemplos e, conseqüentemente, aprimorando a acurácia e a robustez. A biblioteca *NLPAUG* possibilita a aplicação de técnicas como substituição por sinônimos, inserção de palavras e modificações em nível de caracteres e criação de textos por LLMs.

3.5 Redução de Dimensionalidade

No artigo [22] de SANTOS *et al.*, eles explicam que a Redução de Dimensionalidade é o gerenciamento da alta quantidade de dimensões (variáveis) dos dados, que frequentemente torna o processamento dos algoritmos excessivamente lentos. O trabalho lista técnicas de redução de dimensionalidade como PCA, LDA (*Linear Discriminant Analysis*), SVD (*Singular Value Decomposition*). Estas técnicas diminuem as dimensões das características (termos) dos dados, melhorando o desempenho dos algoritmos de AM por exigir menos computacionalmente [38].

3.5.1 Análise de Componentes Principais - PCA

SEHGAL *et al* tratam da técnica *Principal Component Analysis (PCA)*, Análise de Componentes Principais, no artigo [23]. Definem PCA como uma técnica que reconhece padrões ao analisar dados de alta dimensionalidade, muito difíceis de interpretar dado o volume de característica para reduzi-los. Através de algoritmos de álgebra linear, o método diminui um grande conjunto de variáveis correlacionadas para um número pequeno e representativo de variáveis, os componentes principais, mantendo a quantidade máxima de informação nas restantes.

No artigo SANTOS *et al* [22], os autores aplicam PCA em CTD, para reduzir as dimensões das características dos textos, e melhorar significativamente o desempenho dos algoritmos de AM.

3.6 Modelos

3.6.1 Energy-based Flow Classifier - EFC

O algoritmo *Energy-based Flow Classifier (EFC)*⁴ é um classificador desenvolvido para a detecção de anomalias e intrusões em fluxos de redes, estruturado sob o paradigma de CIUC e fundamentado na estatística inversa do modelo de Potts, conceito da mecânica quântica. O EFC infere o modelo estatístico baseando-se exclusivamente em amostras benignas, eliminando a dependência de grandes bases de dados contendo ataques maliciosos previamente rotulados e torna o modelo mais resiliente e adaptável a novos domínios, isto é, a diferentes distribuições de dados. Quando a energia calculada para um determinado fluxo ultrapassa um limiar estatístico (*threshold*) preestabelecido — indicando que o comportamento difere consideravelmente do tráfego normal aprendido — a amostra é classificada como anômala ou maliciosa.

Modelo Matemático

O conceito matemático e computacional do EFC consiste no cálculo da energia total de cada fluxo, análogo à função Hamiltoniana. Seguindo o Princípio da Maximização da Entropia, a energia de uma configuração de fluxo k é dada por:

$$H(a_{k1}, \dots, a_{kN}) = - \sum_{i,j|i < j} e_{ij}(a_{ki}, a_{kj}) - \sum_i h_i(a_{ki}) \quad (3.1)$$

Para determinar os parâmetros matemáticos do modelo a partir da amostra de treinamento benigna, o algoritmo utiliza uma aproximação Gaussiana que infere os acoplamentos e_{ij} por meio da inversão da matriz de covariância C das frequências empíricas conjuntas, conforme:

$$e_{ij}(a_i, a_j) = -(C^{-1})_{ij}(a_i, a_j) \quad (3.2)$$

Os campos locais h_i , associados ao estado individual de cada atributo, são definidos por uma aproximação de campo médio, resultando:

$$h_i(a_i) = \ln \left(\frac{f_i(a_i)}{f_i(Q)} \right) - \sum_{j, a_j} e_{ij}(a_i, a_j) f_j(a_j) \quad (3.3)$$

Execução do Algoritmo

A execução do EFC está descrita no Algoritmo 1, retirado do trabalho de PONTES [19]:

⁴<https://efc-package.readthedocs.io/en/latest/index.html>

Algorithm 1 Energy-based Flow Classifier

Require: $\text{normal_flows} \in R^{K \times N}$, Q , α , $cutoff$

```
1: Import all model inference functions
2:  $f_i \leftarrow \text{SiteFreq}(\text{normal\_flows}, Q, \alpha)$ 
3:  $f_{ij} \leftarrow \text{PairFreq}(\text{normal\_flows}, f_i, Q, \alpha)$ 
4:  $e_{ij} \leftarrow \text{Couplings}(f_i, f_{ij}, Q)$ 
5:  $h_i \leftarrow \text{LocalFields}(e_{ij}, f_i, Q)$ 
6: while scanning the network do
7:    $\text{flow} \leftarrow \text{wait\_for\_incoming\_flow}()$ 
8:    $e \leftarrow 0$ 
9:   for  $i \leftarrow 1$  to  $N - 1$  do
10:     $a_i \leftarrow \text{flow}[i]$ 
11:    for  $j \leftarrow i + 1$  to  $N$  do
12:       $a_j \leftarrow \text{flow}[j]$ 
13:      if  $a_i \neq Q$  and  $a_j \neq Q$  then
14:         $e \leftarrow e - e_{ij}[i, a_i, j, a_j]$ 
15:      end if
16:    end for
17:    if  $a_i \neq Q$  then
18:       $e \leftarrow e - h_i[i, a_i]$ 
19:    end if
20:  end for
21:  if  $e \geq cutoff$  then
22:     $\text{stop\_flow}()$ 
23:     $\text{forward\_to\_DPI}()$ 
24:  else
25:     $\text{release\_flow}()$ 
26:  end if
27: end while
```

O algoritmo pode ser analisado em duas partes: pré-cálculo e monitoramento contínuo. Inicialmente, calcula-se as frequências, os acoplamentos e os campos locais a partir do tráfego benigno, estabelecendo um limiar de corte ($cutoff$, $threshold$, limiar estatístico), frequentemente definido no percentil 95 da distribuição das energias do conjunto de treinamento.

Na fase de monitoramento em tempo real, quando um novo fluxo é capturado, sua energia e é inicializada em zero e sofre deduções iterativas baseadas nas penalizações correspondentes aos seus atributos observados:

$$e \leftarrow e - e_{ij}[\dots] - h_i[\dots] \quad (3.4)$$

Se a energia final computada satisfizer $e \geq cutoff$, o modelo infere que o padrão foge da normalidade e o fluxo é classificado como malicioso, sendo encaminhado para uma inspeção profunda de pacotes (*Deep Packet Inspection* - DPI). Caso contrário, o tráfego é validado e liberado.

3.6.2 BERTimbau

O *Bidirectional Encoder Representations from Transformers para o português brasileiro* (*BERTimbau*) é um modelo treinado na arquitetura *Bidirectional encoder representations from transformers* (*BERT*) para a língua portuguesa do Brasil com o intuito de utilizá-lo para tarefas de Processamento de Linguagem Natural (PLN). O estudo [39] utilizou a estratégia de *transfer learning* baseada em modelos de linguagem (LMs) pré-treinados, os quais consistem em modelos de grande escala treinados sobre extensos corpora textuais. O BERTimbau foi estudado em três tarefas subsequentes de PLN: Similaridade Textual entre Sentenças (*Sentence Textual Similarity* - *STS*), Reconhecimento de Inferência Textual (*Recognizing Textual Entailment* - *RTE*) e Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - *NER*). E segundo os autores superou o mBERT em todas as tarefas avaliadas (STS, RTE e NER), validando, assim, a eficácia do uso de modelos de linguagem pré-treinados específicos por idioma para o português.

O BERTimbau foi empregado no trabalho de [11] para tarefas de Geração de Textos (GT) e CT, utilizando o *framework NLPAUG*, apresentado em [25].

3.6.3 Light Gradient Boosting Machine - LGBM

O *Light Gradient Boosting Machine* (*LGBM*) é um algoritmo muito eficiente de *Gradient Boosting* projetado para superar desafios de escalabilidade ao lidar com grandes volumes de dados e com alta dimensionalidade. Para acelerar o treinamento em até 20 vezes sem sacrificar a precisão, o modelo adota uma abordagem baseada em histogramas e introduz duas técnicas principais: o GOSS (*Gradient-based One-Side Sampling*), que reduz as instâncias analisadas focando naquelas com maiores gradientes, e o EFB (*Exclusive Feature Bundling*), que diminui o número de dimensões agrupando variáveis mutuamente exclusivas. Essa arquitetura combinada garante um desempenho preditivo robusto com menor consumo de memória e maior velocidade de treinamento [40].

3.6.4 Support Vector Machine - SVM

O *Support Vector Machine (SVM)* é um modelo linear de AM amplamente utilizado na resolução de problemas de classificação. O algoritmo atua representando as observações em um espaço de dimensão N para encontrar uma fronteira linear, ou hiperplano, capaz de separar os dados em suas respectivas classes. O grande diferencial do SVM está na forma como essa segmentação é feita: a escolha do hiperplano baseia-se na obtenção da "*margem máxima de separação*", ou seja, a maior distância possível entre a fronteira de decisão e os pontos de dados (vetores de suporte) das classes. Ao focar na maximização dessa margem por meio de fronteiras lineares, o algoritmo garante alta eficiência e robustez na separação das classes ([22] e [21]).

3.6.5 Logistic Regression - LR

O *Logistic Regression (LR)* é um algoritmo de AM supervisionado que utiliza uma função logística para deduzir um modelo matemático preditivo a partir dos dados de treinamento. Estruturalmente, o método aplica a função sigmoide (também conhecida como curva S) para ajustar o modelo aos dados, reconhecendo um vetor contendo variáveis previamente rotuladas e avaliando os coeficientes correspondentes a cada variável de entrada. Assim, o algoritmo é capaz de prever a classe de um novo dado, destacando-se por sua eficiência, uma vez que exige um CC relativamente menor quando comparado a técnicas de classificação mais complexas ([29], [21] e [13]).

3.6.6 K-Nearest Neighbors - KNN

O *K-Nearest Neighbors (KNN)* é um algoritmo de classificação clássico de AM supervisionado baseado em instâncias, chamado de (*lazy learner*), porque executa a inferência apenas no momento em que uma nova amostra de teste é dada. Ao avaliar um novo item, o algoritmo calcula a sua similaridade em relação aos dados de treinamento e seleciona as k amostras mais semelhantes, que são denominadas como seus "*vizinhos*". A classificação final do novo objeto é obtida por meio de uma votação de maioria simples entre esses k vizinhos mais próximos. A partir da probabilidade de um determinado ponto pertencer a uma certa vizinhança, o modelo atribui a nova amostra à classe predominante entre os seus vizinhos.([11], [18] e [21])

3.7 Métrica de Avaliação

Para comparar os modelos e selecionar aqueles que melhor atendem ao problema estudado, é necessário utilizar múltiplas métricas de avaliação e, posteriormente, compará-las entre

si ([41]). Assim, considerando as métricas apontadas nos artigos presentes no Capítulo 2, assim como a problemática deste estudo apontada no Capítulo 1 destaco as seguintes métricas que são calculados a partir das relações observadas na Matriz de Confusão, conforme ilustrada na 3.1.

Tabela 3.1: Matriz de Confusão

Observações	Previsões do Modelo	
	Positivas	Negativas
Positivas	TP (Verdadeiro Positivo)	FN (Falso Negativo)
Negativas	FP (Falso Positivos)	TN (Verdadeiro Negativo)

3.7.1 Recall

A métrica de ***Micro Recall*** reflete a necessidade de classificar corretamente os itens pertencentes a cada classe, calculado separadamente para cada classe, representa a proporção de instâncias dessa classe que são corretamente identificadas.

$$\text{Recall}_{\text{Positive}} = \frac{TP_{\text{pos}}}{TP_{\text{pos}} + FN_{\text{pos}}} \quad (3.5)$$

$$\text{Recall}_{\text{Negative}} = \frac{TP_{\text{neg}}}{TP_{\text{neg}} + FN_{\text{neg}}} \quad (3.6)$$

onde:

- $TP_{\text{pos}} = \text{true positives}$ (verdadeiro positivos) da classe positiva ($= TN_{\text{neg}}$)
- $FN_{\text{pos}} = \text{false negatives}$ (falsos negativos) da classe positiva ($= FP_{\text{neg}}$)
- $TP_{\text{neg}} = \text{true positives}$ (verdadeiro positivos) da classe negativa ($= TN_{\text{pos}}$)
- $FN_{\text{neg}} = \text{false negatives}$ (falsos negativos) da classe negativa ($= FP_{\text{pos}}$)

3.7.2 False Discovery Rate - FDR

A métrica *False Discovery Rate (FDR)*, taxa de descobertas falsas, mede a proporção de *falsos alarmes*, isto é, falsos positivos, em relação ao total de predições positivas.

$$\text{FDR}_{\text{Positive}} = \frac{FP_{\text{pos}}}{TP_{\text{pos}} + FP_{\text{pos}}} \quad (3.7)$$

$$\text{FDR}_{\text{Negative}} = \frac{FP_{\text{neg}}}{TP_{\text{neg}} + FP_{\text{neg}}} \quad (3.8)$$

onde:

- $FP_{\text{pos}} = \text{false positives}$ (falsos positivos) da classe positiva ($= FN_{\text{neg}}$)
- $TP_{\text{pos}} = \text{true positives}$ (verdadeiros positivos) da classe positiva ($= TN_{\text{neg}}$)
- $FP_{\text{neg}} = \text{false positives}$ (falsos positivos) da classe negativa ($= FN_{\text{neg}}$)
- $TP_{\text{neg}} = \text{true positive}$ (verdadeiros positivos) da classe negativa ($= TN_{\text{pos}}$)

É importante enfatizar a relação do FDR com a métrica de Precisão.

A métrica Precisão é definida como:

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (3.9)$$

Tanto a FDR quanto a Precisão compartilham o mesmo denominador e, portanto, apresentam uma relação direta:

$$\text{FDR} = 1 - \text{Precisão}.$$

Quanto menor a FDR, maior a Precisão.

3.7.3 Acurácia

Avalia o desempenho geral do modelo em todas as previsões, sem distinguir entre classes majoritárias e minoritárias. A acurácia é definida como a proporção de instâncias classificadas corretamente (verdadeiros positivos + verdadeiros negativos) em relação ao número total de previsões (verdadeiros positivos + falsos positivos + verdadeiros negativos + falsos negativos).

$$\text{Acurácia} = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.10)$$

onde:

- $TP = \text{true positives}$ (verdadeiros positivos)
- $TN = \text{true negatives}$ (verdadeiros negativos)
- $FP = \text{false positives}$ (falsos positivos)
- $FN = \text{false negatives}$ (falsos negativos)

3.7.4 AUC-ROC

A métrica *Area Under the Curve - Receiver Operating Characteristic (AUC-ROC)* avalia a capacidade de um classificador binário em distinguir entre duas classes, como Ouv-DF e e-SIC DF. Assim, quanto maior o valor da AUC-ROC, melhor o desempenho do modelo.

$$\text{AUC} = \int_0^1 \text{TPR}(t) d\text{FPR}(t) \quad (3.11)$$

onde:

- $\text{TPR}(t) = \frac{TP(t)}{TP(t)+FN(t)}$ é a taxa de verdadeiros positivos (*True Positive Rate*) no limiar t
- $\text{FPR}(t) = \frac{FP(t)}{FP(t)+TN(t)}$ é a taxa de falsos positivos (*False Positive Rate*) no limiar t
- $TP = \text{true positives}$ (verdadeiros positivos)
- $TN = \text{true negatives}$ (verdadeiros negativos)
- $FP = \text{false positives}$ (falsos positivos)
- $FN = \text{false negatives}$ (falsos negativos)

3.7.5 F1-Score

O *F1-Score* é uma métrica que combina Precisão e *Recall* por meio de sua média harmônica. Como a métrica de Precisão é definida como o complemento do FDR, o *F1-Score* também reflete na taxa de falsas descobertas. Valores elevados de FDR implicam em redução da Precisão e, conseqüentemente, diminuição do *F1-Score*, mesmo quando o *Recall* é alto. Portanto, o *F1-Score* penaliza modelos que apresentam elevado número de falsos positivos, promovendo um equilíbrio entre acertos e erros do classificador.

$$\text{F1-Score}_{\text{Micro}} = \frac{2 \cdot \text{Precisão}_{\text{Micro}} \cdot \text{Recall}_{\text{Micro}}}{\text{Precisão}_{\text{Micro}} + \text{Recall}_{\text{Micro}}} \quad (3.12)$$

$$\text{F1-Score}_{\text{Micro}} = \frac{2TP_{\text{Micro}}}{2TP_{\text{Micro}} + FP_{\text{Micro}} + FN_{\text{Micro}}} \quad (3.13)$$

onde:

- $TP = \text{true positives}$ (verdadeiros positivos)
- $FP = \text{false positives}$ (falsos positivos)
- $TP = \text{true positive}$ (verdadeiros positivos)
- $FP = \text{false positives}$ (falsos positivos)

A definição do *F1-Score* micro é apresentada na Equação 3.12. As Equações 3.12 a 3.13 apresentam o cálculo do *F1-Score* micro.

Capítulo 4

Experimentos

Este capítulo apresenta a metodologia de pesquisa empregada neste estudo e os experimentos realizados. Os tópicos deste Capítulo, assim como a execução dos experimentos deste estudo, foram divididos pelas etapas definidas na Trajetória de Ciência de Dados (TCD) Escolhida considerando o exposto na metodologia TCD[1], Figura 3.2.

4.1 Metodologia - Trajetória de Ciência de Dados (TCD)

Para determinar a trajetória a ser utilizada neste estudo, consideraram-se o contexto e os problemas abordados neste trabalho, descritos no Capítulo 1. E também os problemas discutidos no Capítulo 2 em contextos semelhantes, artigos de NOGUTI, VELLASQUES e OLIVEIRA [10]; SOUZA [21]; CORDEIRO, RABELO E MOURA [11]; e SANTOS (*et al*)[22].

Dessa forma, a trajetória escolhida é apresentada nas Figura 4.1 e Figura 4.2, sendo composta pelas seguintes etapas: Exploração do Objetivo, Entendimento do Negócio, Exploração da Fonte dos Dados, Aquisição dos Dados, Entendimento dos Dados, Arquitetura dos Dados, Preparação dos Dados, Modelagem, Avaliação, Implementação e Disponibilização dos Dados.

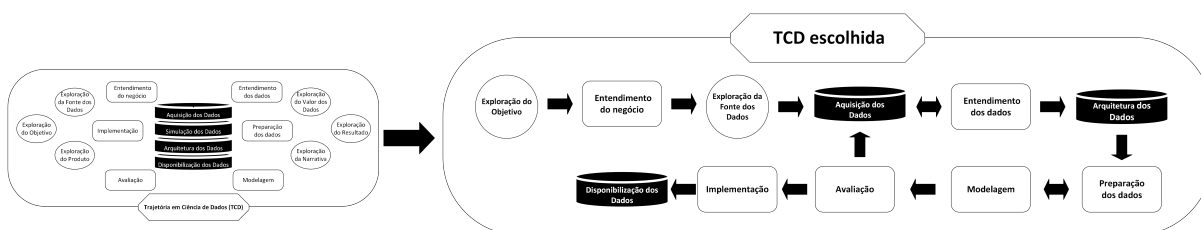


Figura 4.1: Trajetória de Ciência de Dados selecionada

Observando a Figura 4.2, as três primeiras etapas podem ser compreendidas como uma única fase, ocorrendo de forma sequencial e executadas apenas uma vez: Exploração do Objetivo, Entendimento do Negócio, Exploração da Fonte dos Dados.

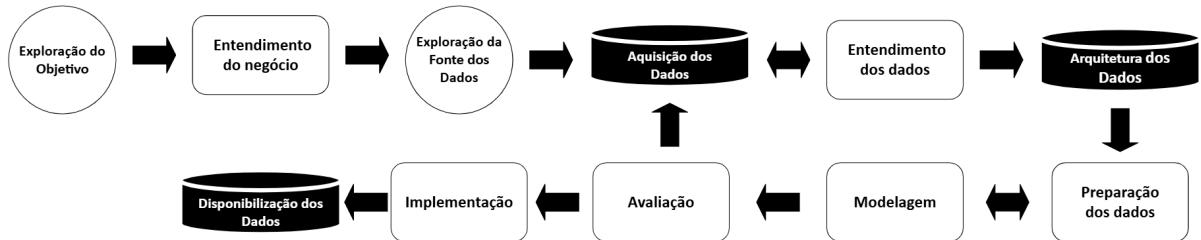


Figura 4.2: Trajetória de Ciência de Dados Escolhida

4.2 Exploração do Objetivo

Aqui delimitaram-se os objetivos do negócio (CGDF) e qual dentre os vários determinados podem ser alcançados de uma forma orientada a dados. E estão descritos nos itens *1.2 Contexto Específico*, *1.4.1 Justificativa do tema* e *1.4.2 Contribuições Esperadas* do Capítulo 1

4.3 Entendimento do Negócio

Nesta seção o foco é entender os objetivos e requisitos do projeto sob a perspectiva do negócio e convertendo-os em definição de problema.

4.3.1 Objetivos da aplicação

A aplicação proposta neste estudo tem como cenário a plataforma ParticipaDF, de responsabilidade da Controladoria-Geral do Distrito Federal. Sendo o seu objetivo geral automatizar a classificação dos relatos redigidos pelos cidadãos no ParticipaDF entre o Sistema Eletrônico do Serviço de Informações ao Cidadão do Distrito Federal (e-SIC DF) e o Sistema de Ouvidoria do Distrito Federal (Ouv-DF) por meio de Aprendizagem de Máquina (AM). E tendo como *"critérios de sucesso da aplicação"* a redução da variabilidade interpretativa dos casos, a redução do tempo gasto pelo usuário no registro do pedido, e o aumento da facilidade de uso do Participa DF para o mais leigo usuário.

4.3.2 Situação

Contexto

A presente dissertação terá como contexto os pedidos registrados no site da plataforma de participação popular Participa DF, administrada pela Controladoria-Geral do Distrito Federal (CGDF), que hospeda os sistemas Sistema Eletrônico do Serviço de Informações ao Cidadão do Distrito Federal (e-SIC DF) e Sistema de Ouvidoria do Distrito Federal (Ouv-DF).

Infraestrutura

A infraestrutura computacional adotada para a modelagem foi o próprio ambiente da CGDF, de modo a garantir a preservação da confidencialidade das informações contidas nas manifestações dos cidadãos.

Os experimentos foram realizados em uma GPU NVIDIA Tesla T4, com 15 GB de memória dedicada e 234 GB de memória RAM disponíveis no servidor.

Como parte da infraestrutura tecnológica, utilizou-se o Jupyter Hub, Fast API, TFS, e Jenkins para experimentação.

A linguagem de programação adotada neste projeto é a Python, porque é a adotada pela Diretoria de Ciência de Dados (DICID) em seus projeto de Aprendizagem de Máquina (AM).

Os dados foram provenientes de bases institucionais sob responsabilidade da CGDF, e-SIC DF e Ouv-DF.

Limitações

Como consequência do contexto e da infraestrutura citamos a limitações do projeto quanto:

- à dificuldade em treinar modelos ou aplicar técnica com elevado custo computacional, devido à limitações da GPU;
- à linguagem de programação, a adotada pela Diretoria de Ciência de Dados (DICID) em seus projeto de AM, que é a linguagem Python;
- à temporalidade dos dados, pois a memória da GPU é limitada para treinar os modelos clássicos com a quantidade disponível de dados.

Custos

Quanto aos custos advindos da criação da aplicação, podemos citar o de utilização da própria infraestrutura da CGDF e dos servidores que trabalham na CGDF.

4.3.3 Determinação das metas

O cliente a ser atendido pela aplicação será a população do Distrito Federal. Os especialistas consultados foram os dos setores da CGDF responsáveis respectivamente pelo e-SIC DF e pelo Ouv-DF, a Subcontroladoria de Transparência e Controle Social (SUBTC) e a Ouvidoria-Geral do Distrito Federal (OGDF) respectivamente. O "*dono do problema*" foi a Diretoria de Ciência de Dados (DICID) responsável pelo desenvolvimento dos modelos e da aplicação.

As metas da aplicação foram:

- reduzir o tempo gasto nas atividades rotineiras que serão automatizadas, apresentando desempenho superior ao humano;
- reduzir os encerramentos de solicitações que foram direcionadas para o sistema errado;
- reduzir o tempo de preenchimento das solicitações;
- aumentar a facilidade de uso do Participa DF, permitindo a utilização por até o mais leigo dos usuários;
- aumentar os índices de satisfação do cidadão com o Participa DF.

Para atingir estas metas, as ferramentas disponíveis na infraestrutura da CGDF são adequadas.

4.3.4 Plano do Projeto

Considerando a literatura apresentada no Capítulo 2, o presente estudo propõe como solução para o problema apresentado no Capítulo 1, aplicar a técnica de Aprendizagem de Máquina EFC a base de textos desbalanceadas da plataforma Participa DF¹ para criar modelos que alcancem maior assertividade na rotulação da classe minoritária e que possuam um menor Custo de Treinamento de Modelos, em contraposição às técnicas corriqueiramente utilizadas em problemas de rotulação de bases textuais desbalanceadas. O presente trabalho propõe a classificação de um *corpora* composto por duas classes

¹<https://www.participa.df.gov.br/>

desbalanceadas, provenientes do *corpus* do sistema e-SIC DF (classe minoritária) e do *corpus* do Ouv-DF (classe majoritária).

O trabalho de CORDEIRO, RABELO e MOURA [11] apresenta o cenário mais parecido ao do presente trabalho, a classificação de comunicações à órgão de controle com base de dados desbalanceada. Apesar de trabalhar em rotulação de múltiplas classes e não categorização binária (foco do presente estudo), o estudo de CORDEIRO, RABELO e MOURA [11] apresenta uma abordagem prática propícia para solucionar o caso desta dissertação, ao classificar de forma eficiente as manifestações de classes desbalanceadas.

Apesar do trabalho de PONTES *et al* [16] não tratar de classificação de textos, ele apresenta uma abordagem diferente para rotulação de *dataframes* severamente desbalanceados e pode ser aplicado para tarefas de Classificação de Textos Desbalanceados (CTD), sendo uma lacuna de estudo.

Dessa maneira, este trabalho propõe aplicar o algoritmo EFC em um contexto de desbalanceamento de textos para classificação binária.

4.4 Exploração da Fonte de Dados

Tanto o e-SIC DF quanto o Ouv-DF possuem uma base de dados estruturada com registros superiores à dez anos, que representam valiosas fontes de dados.

Contudo, os dados do e-SIC DF passaram a ser armazenado em outro banco de dados a partir do lançamento do Participa DF em 08/12/2022, possuindo nova estrutura. Até o momento não foi definida a correspondência das colunas da base antiga para a base nova.

4.5 Aquisição dos Dados

As etapas subsequentes da TCD escolhida correspondem às fases que podem envolver iterações, conforme ilustrado na Figura 4.2. Essas iterações têm como objetivo convergir para uma solução do problema estudado a cada repetição. As iterações compreendem as etapas: *Aquisição dos Dados*. *Entendimento dos Dados*; *Arquitetura dos Dado*, *Preparação dos Dados*; *Modelagem e Avaliação*. Estas repetições podem ocorrer em função dos resultados obtidos em cada etapa, de forma individual ou conjunta. Com base nas informações obtidas, os "*donos do problema*" decidem se as etapas devem ser repetidas, alterando ou não a abordagem adotada em cada fase, sempre com o objetivo de aproximar a solução do problema em estudo.

Na etapa Aquisição dos Dados foram extraídos registros cujas manifestações já haviam sido concluídas, isto é, já analisadas pela área especializada e rotuladas com a classificação final. Uma amostra estática foi extraída da base de dados do Sistema Eletrônico do

Serviço de Informações ao Cidadão do Distrito Federal (e-SIC DF) e disponibilizada no ambiente de homologação. Quanto aos registros da base de dados do Sistema de Ouvidoria do Distrito Federal (Ouv-DF), eles são extraídos diretamente do ambiente de produção considerando o frequente acesso a ele para outros estudos. Destas extrações dos bancos de dados do Participa DF, obteve-se o *corpus* do sistema e-SIC DF e o *corpus* do sistema Ouv-DF. Este *corpora* formou um único *dataframe* foco deste estudo formado por duas classes, e-SIC DF e Ouv-DF.

4.6 Entendimento dos Dados

As amostras obtidas dos bancos de dados do Participa DF representam um período de 32 meses, com aproximadamente 58.000 registros da classe e-SIC DF e 864.000 registros da classe Ouv-DF.

Das observações obtidas na Análise Exploratória de Dados (AED) destacamos as distribuição dos relatos em cada sistema ao longo do tempo (Figura 4.3) e Figura 4.4), na qual verificamos a sazonalidade das demandas; e a distribuição de palavras por relatos na qual verificamos as dimensionalidades dos textos (Classe e-SIC DF, Figura 4.5. Classe Ouv-DF, Figura 4.6).

Contudo, devido às limitações de processamento da GPU, o conjunto de dados utilizado para o treinamento dos modelos foi reduzido. Da base e-SIC DF foram mantidos 7.977 registros, e da base Ouv-DF mantidos 160.698 registros. Verificou-se que os registros de ambos os sistemas, formaram um *dataframe* com as duas classes (e-SIC DF e Ouv-DF) totalizando 168.675 registros. O *corpus* resultante apresentou uma proporção de 4,73% para a classe minoritária e-SIC DF e de 95,27% para a classe majoritária Ouv-DF, resultando em uma razão de desbalanceamento[13] de 20,15.

4.7 Arquitetura dos Dados

A etapa de Arquitetura dos Dados, conforme preconizada pela metodologia de Trajetória de Ciência de Dados (TCD), consiste em projetar o esquema lógico das informações para a integração de diferentes fontes de dados.

Sob esse rigor conceitual, o projeto arquitetural desta pesquisa objetivou unificar dados de duas bases de naturezas distintas, assim, extraíndo registros dinâmicos do ambiente de produção do sistema Ouv-DF e estáticos do ambiente de homologação do sistema e-SIC DF, no período de 08/12/2022 a 01/09/2025.

Para consolidar essa integração, mapeou-se a correspondência dos atributos de interesse, sincronizando a variável textual "*Relato*" e a variável de data "*Abertura*" do sis-

Comparação Volume Registros/Mês
Dez/2022 à Nov/2023 X Dez/2023 à Nov/2024 X Dez/2024 à Ago/2025
eSicDF

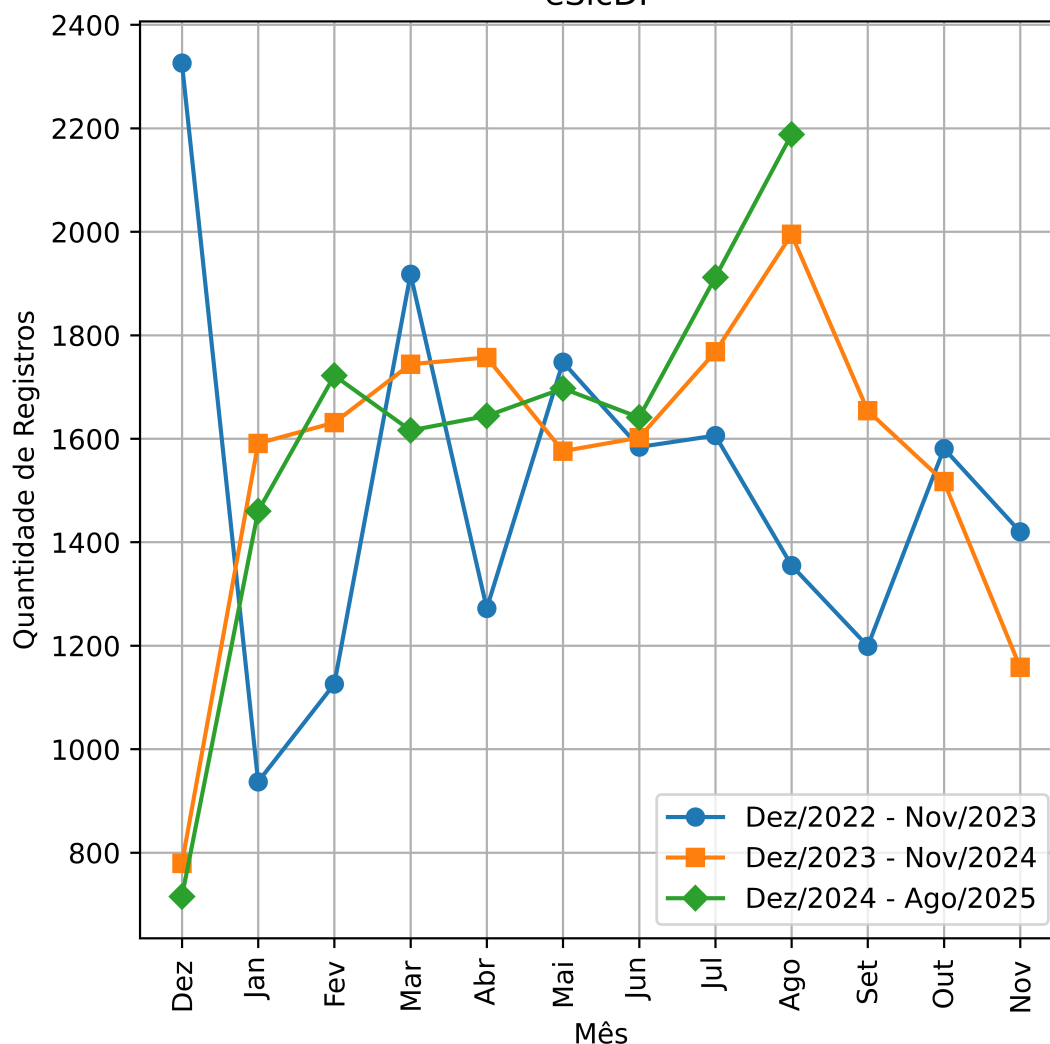


Figura 4.3: Quantidade de Registros por mês no sistema e-SIC DF .

tema Ouv-DF com os respectivos campos *"DS_TEXTO_PEDIDO"* e *"DT_PEDIDO"* do banco do e-SIC DF.

A partir deste desenho de compatibilidade, e a determinação do volume de dados a ser processado na modelagem, executaram-se as transformações para garantir que as colunas correspondentes possuíssem os mesmos tipos de dados. A efetivação destas transformações e arranjos assegurou a construção de uma base estruturada e consistente, habilitando a massa de dados para a subsequente etapa de modelagem preditiva.

Comparação Volume Registros/Mês
 Dez/2022 à Nov/2023 X Dez/2023 à Nov/2024 X Dez/2024 à Ago/2025
 OuvDF

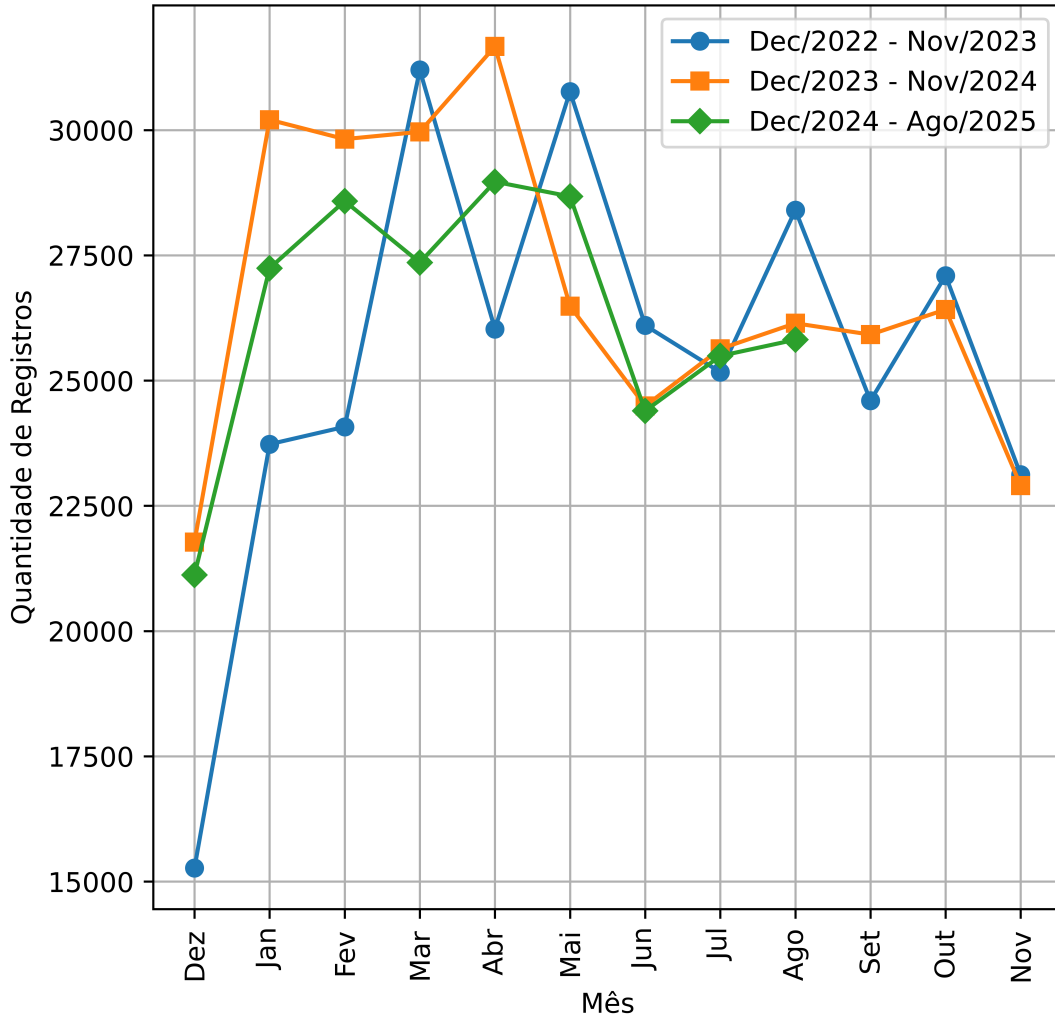


Figura 4.4: Quantidade de Registros por mês no sistema Ouv-DF .

4.8 Preparação dos Dados

Devido às limitações de processamento da GPU, o conjunto de dados utilizado para o treinamento dos modelos foi reduzido para 168.675 registros, 7.977 registros da classe e-SIC DF, e 160.698 registros da classe Ouv-DF.

Com o objetivo de comparar o desempenho dos modelos considerando também variações no pré-processamento, foram selecionadas diversas técnicas, aplicadas ou não conforme o caso, resultando em diferentes *pipelines* de pré-processamento que foram submetidas aos algoritmos para treinamento dos modelos, Figura 4.7.

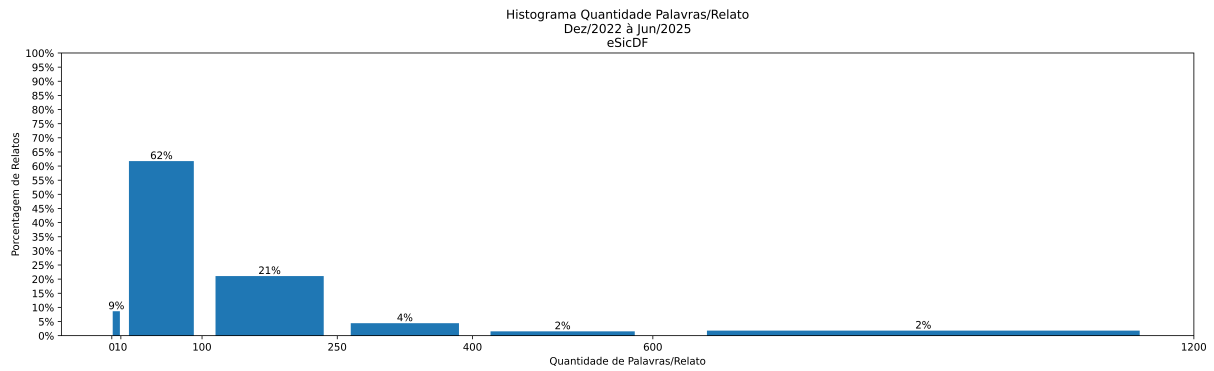


Figura 4.5: Histograma Palavras por Relato - classe e-SIC DF .

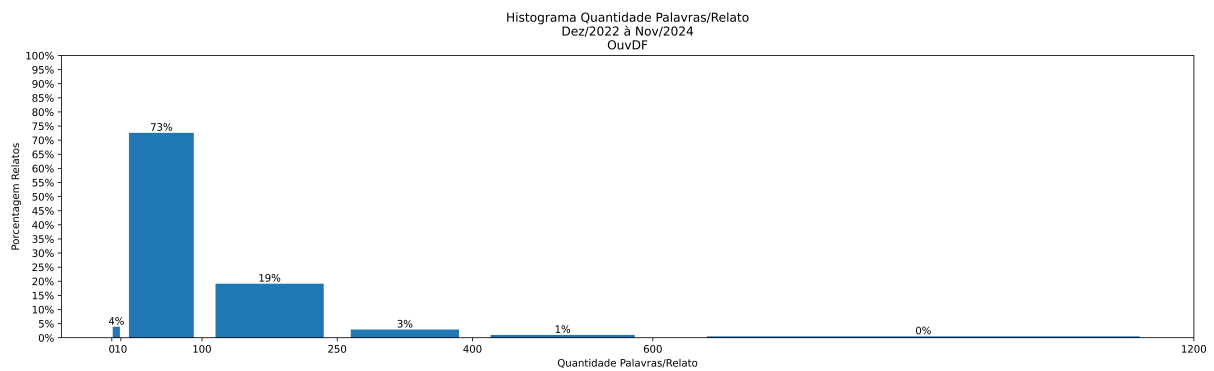


Figura 4.6: Histograma Palavras por Relato - classe Ouv-DF .

4.8.1 Aplicadas a todo o corpus

- **Engenharia de atributos** – criação de um atributo categórico para diferenciar as classes; ajustes nos nomes e nos tipos das colunas.
- **Remoção de valores ausentes e duplicados** – realizada antes e após a junção dos conjuntos de dados provenientes dos sistemas e-SIC DF e Ouv-DF.

4.8.2 Aplicadas em diferentes etapas

Normalização de palavras

Executada antes da aplicação da redução de dimensionalidade por PCA[23], foi necessária a normalização textual por meio de duas técnicas:

- lematização ²(Exemplos Tabela 4.1)

²<https://spacy.io/models/pt>

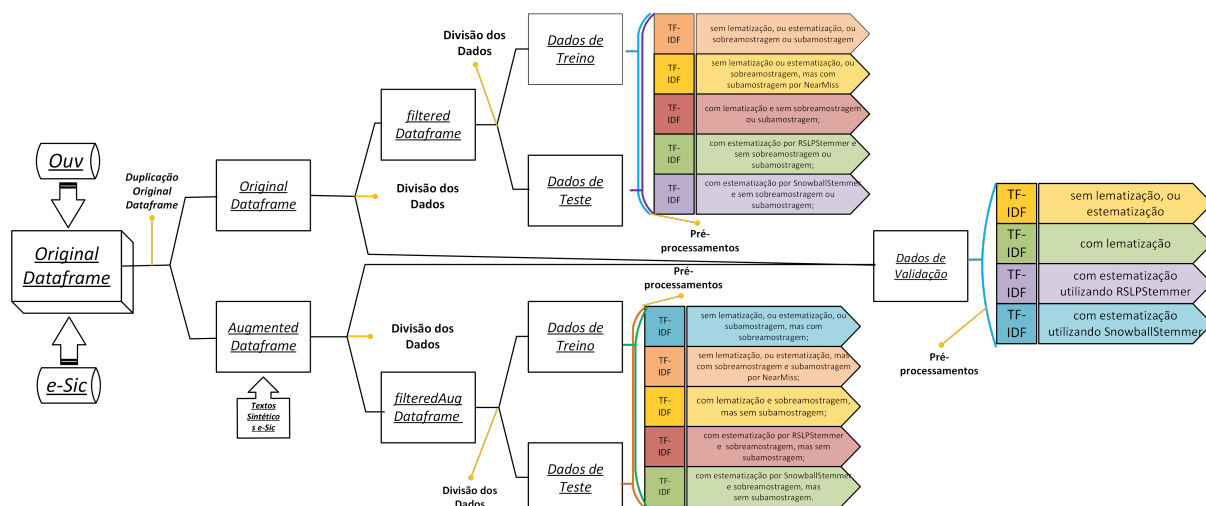


Figura 4.7: Pré-processamentos realizados

- stemming - função *RSLPStemmer*³ (Exemplos Tabela 4.1)
- stemming - função *SnowballStemmer*⁴ (Exemplos Tabela 4.1)

Tabela 4.1: Exemplos de pré-processamento textual

Relatos	Limpos	Lematizados	RSLPStemmer	SnowballStemmer
Boa noite, gostaria de informações sobre as ru...	boa noite gostaria de informações sobre as rua...	bom noite gostar de informação sobre o rua de ...	boa noit gost de inform sobr as rua do sol nas...	boa noit gost de inform sobr as ruas do sol na...
Prezados, Com fundamento na Lei nº xx.xxx/20xx...	prezados com fundamento na lei nº xxxxx20xx l...	prezar com fundamento em o lei nº lei de acess...	prez com fundament na lei nº xxxxx20xx lei de ...	prez com fundament na lei nº xxxxx20xx lei de ...
buraco nesta via	buraco nesta via	buraco em este via	burac nest via	burac nest via

Redução de dimensionalidade

Realizada por *Principal Component Analysis (PCA)* em conjunto com a normalização de palavras realizada anteriormente([23]). Foram testados 100 e 500 componentes nos testes de hiperparâmetros da função *GridSearchCV*.

Extração de atributos e limpeza dos dados

- Vetorização foi realizada por meio de *Term Frequency-Inverse Document Frequency (TF-IDF)*, limitada às 5.000 palavras mais frequentes;
- Remoção de *stopwords* em português; (Exemplos -Tabela 4.1)

³https://www.nltk.org/_modules/nltk/stem/rsdp.html

⁴<https://www.nltk.org/api/nltk.stem.SnowballStemmer.html>

- Conversão dos textos para letras minúsculas; (Exemplos - Tabela 4.1)
- Remoção de caracteres especiais, sinais de pontuação e espaços excedentes. (Esse último conjunto de remoções nem sempre foi aplicado ao *pipeline* de pré-processamento do EFC.)

Redimensionamento

Foram utilizadas duas técnicas:

- **Subamostragem** – por meio da função *NearMiss, Version 2*.
- **Sobreamostragem** – por meio da biblioteca NLPAUG⁵ ([25] e [11]).

Sobreamostragem utilizando os modelos BERT:

- **BERTimbau**⁶ ([39]),
- **BERT multilíngue**,
- **DistilBERT multilíngue**,
- **DistilBERTimbau** ([42]) e
- **XLM-RoBERTa**.

E as funções:

- ***nlpaug.augmenter.char.random***⁷: para aplicar caracteres aleatórios ao texto;
- ***nlpaug.augmenter.char.keyboard***⁸: para simular erros de digitação ao texto;
- ***nlpaug.augmenter.word.context__word__embs***⁹: aplica operações lexicais à entrada textual com base em representações contextuais das palavras.

A partir de 7.977 registros, foram criados 40.560 textos sintéticos para o treinamento dos modelos.

4.8.3 Dados de Treinamento, Teste e Validação

A divisão dos textos em "*Dados de Treinamento, Teste e Validação*", Figura 4.7, considerou que foram criados dois *dataframes*:

⁵<https://nlpaug.readthedocs.io/en/latest/index.html>

⁶<https://huggingface.co/neuralmind/bert-large-portuguese-cased>

⁷<https://nlpaug.readthedocs.io/en/latest/augmenter/char/random.html>

⁸<https://nlpaug.augmenter.char.keyboard>

⁹https://nlpaug.readthedocs.io/en/latest/augmenter/word/context_word_embs.html

- um **sem** *sobreamostragem* da classe minoritária (*Original Dataframe*);
- outro **com** *sobreamostragem* da classe minoritária (*Augmented Dataframe*).

Inicialmente, esses conjuntos foram divididos em *Dados de Validação* e *Dados de Treinamento e Teste*.

Os *Dados de Validação* correspondem aos três últimos meses da amostra, totalizando 79.804 registros e sem textos sintéticos. Dessa forma, os *Dados de Validação* provenientes tanto do *Original Dataframe* e do *Augmented Dataframe* são idênticos.

Os *Dados de Treinamento e Teste* do *Original Dataframe* totalizaram 88.871 registros (*filtered Dataframe*), enquanto os do *Augmented Dataframe* totalizaram 129.431 registros (*filteredAug Dataframe*).

A partir dos *dataframes filtered* e *filteredAug*, os conjuntos de treinamento e teste foram separados na proporção de 70%/30%.

Para o *filtered Dataframe*, foram selecionados 62.209 registros para treinamento e 26.662 registros para teste. Para o *filteredAug Dataframe*, foram selecionados 90.601 registros para treinamento e 38.830 registros para teste.

A partir do *filtered Dataframe* e do *filteredAug Dataframe* foram criados diversos *dataframes* de treino e teste de acordo com o pré-processamento que estes dados foram submetidos:

- ***Dataframes Derivados do filtered Dataframe***
 - sem lematização, estematização, sobreamostragem ou subamostragem;
 - sem lematização ou estematização e sem sobreamostragem, mas com subamostragem;
 - com lematização e sem sobreamostragem ou subamostragem;
 - com estematização utilizando RSLPStemmer e sem sobreamostragem ou subamostragem;
 - com estematização utilizando SnowballStemmer e sem sobreamostragem ou subamostragem;
- ***Dataframes Derivados do filteredAug Dataframe***
 - sem lematização ou estematização e sem subamostragem, mas com sobreamostragem;
 - sem lematização ou estematização e com sobreamostragem e subamostragem;
 - com lematização e sobreamostragem, mas sem subamostragem;

- com estematização utilizando RSLPStemmer e com sobreamostragem, mas sem subamostragem;
- com estematização utilizando SnowballStemmer e com sobreamostragem, mas sem subamostragem.

4.9 Modelagem

Com os dados devidamente pré-processados e organizados em formato apropriado para leitura pelos algoritmos citados no item 3.6 *Modelos* do Capítulo 3, procede-se ao treinamento dos modelos de Aprendizagem de Máquina, dando início à fase de Modelagem.

Para selecionar os algoritmos que compõem este estudo na comparação com o algoritmo EFC, foram considerados:

- modelos apresentados na literatura do Capítulo 2;
- classificadores que representassem as diferentes famílias de algoritmos de Aprendizagem de Máquina;
- modelos viáveis para serem treinados na infraestrutura computacional disponível.

Assim, os algoritmos selecionados para este estudo foram:

- ***Logistic Regression (LR)*** - modelo linear;
- ***Light Gradient Boosting Machine (LGBM)*** - modelo do tipo *ensemble* baseado em árvores de decisão;
- ***K-Nearest Neighbors (KNN)*** - modelo baseado em instâncias (*lazy learning*);
- ***Support Vector Machine (SVM)*** - classificador de margem máxima;
- ***Bidirectional Encoder Representations from Transformers para o português brasileiro (BERTimbau)*** ([39]) - modelo de aprendizado profundo e;
- ***Energy-based Flow Classifier (EFC)*** ([16]), foco principal deste estudo.

Para treinar os modelos de classificação, assim como para fazer as inferências, cada algoritmo precisa receber os dados pré-processados nos formatos esperados pelas suas arquiteturas.

Para os algoritmos clássicos de AM, os pré-processamentos realizados para os treinamentos estão representados na Figura 4.8. Em que os 5 *pipelines* de pré-processamento foram realizados para os dados de treino extraídos dos conjuntos de dados *Original Dataframe* e *Augmented Dataframe* (Figura 4.7).

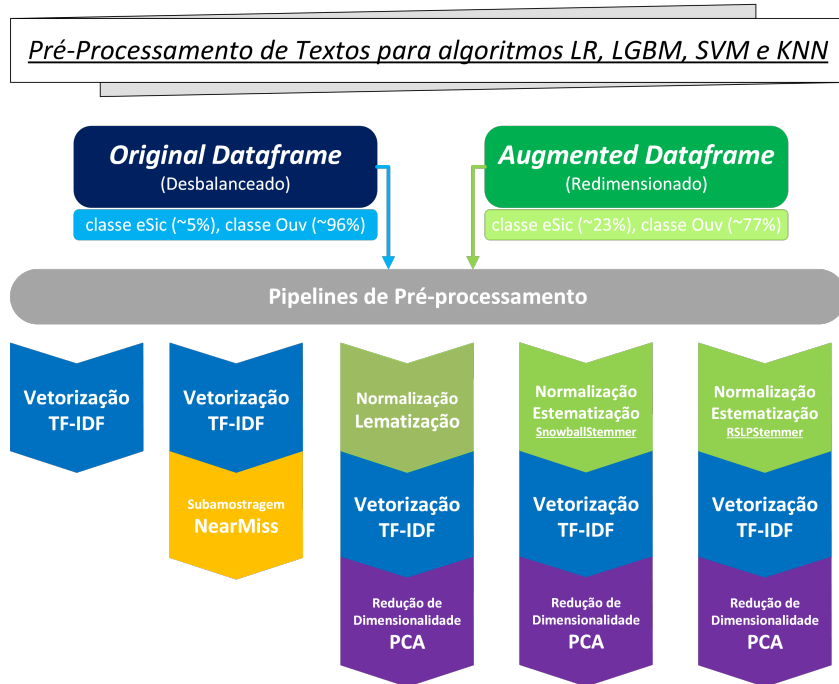


Figura 4.8: Pré-processamentos de Textos Algoritmos Clássicos

Da mesma forma para o BERTimbau foram aplicados os pré-processamentos representados na Figura 4.9. Em que os 2 *pipelines* de pré-processamento foram realizados para os dados de treino extraídos dos conjunto de dados do *Filtered Dataframe* e *FilteredAug Dataframe* (Figura 4.7).

Já para o algoritmo EFC foram aplicados os pré-processamentos representados na Figura 4.10. Em que os 8 *pipelines* de pré-processamento foram realizados para os dados de treino extraídos dos conjuntos de dados *Original Dataframe* (Figura 4.7).

Os algoritmos clássicos de Aprendizagem de Máquina possuem hiperparâmetros que devem ser definidos para personalizar o treinamento dos modelos. Considerando a variedade desses hiperparâmetros e, conseqüentemente, as múltiplas combinações possíveis, a escolha da configuração que resulte no melhor desempenho pode ser desafiadora. Assim, foi empregada a técnica de *Grid Search* em conjunto com *Cross Validation*, pela função *GridSearchCV*¹⁰. Logo, os hiperparâmetros dos modelos clássicos de AM (LR, LGBM, SVM e KNN) foram definidos por esta função *GridSearchCV*, utilizando a métrica de *Recall* como critério para a seleção do melhor conjunto de hiperparâmetros.

Para o EFC, o único hiperparâmetro definido no *pipeline* de treinamento foi a classe a ser utilizada para o treinamento. Como o EFC é um Classificadores de Uma Classe (CIUC), ele foi treinado separadamente para cada classe, e-SIC DF e Ouv-DF.

¹⁰https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

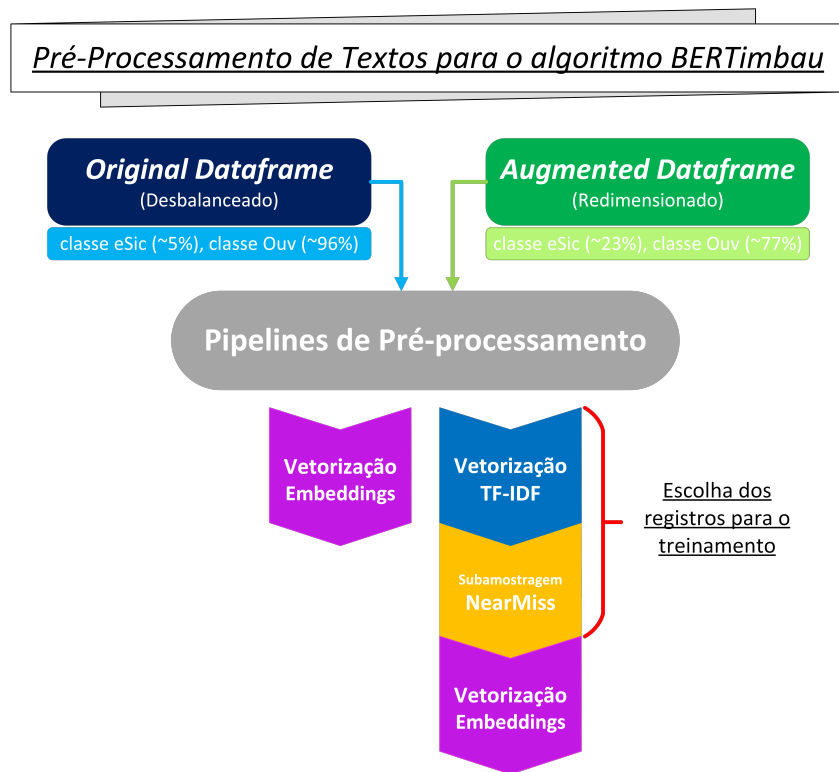


Figura 4.9: Pré-processamentos de Textos para o BERTimbau

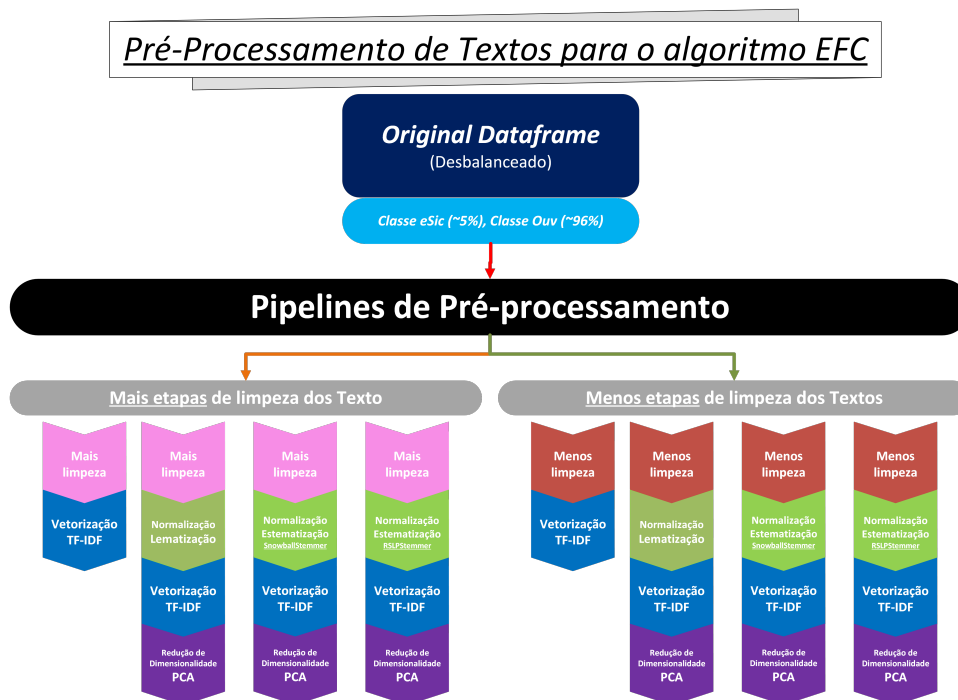


Figura 4.10: Pré-processamentos de Textos para o EFC

No *fine tuning* do BERTimbau, os hiperparâmetros foram definidos de modo a atender aos requisitos da técnica *Quantized Low-Rank Adaptation (QLoRA)*¹¹, utilizada em razão das limitações da GPU onde ocorreu o treinamento, conforme descrito na etapa de *Entendimento do Negócio*. Assim, foi realizada quantização em 4 bits e adaptação por LoRA (*Low-Rank Adaptation*). A métrica escolhida para o *early stopping* de 2 épocas foi o *Recall*; e o otimizador utilizado foi AdamW em 8 bits.

Os processos de treinamento pelos algoritmos de AM foram representados na Figura 4.11. No topo estão os processos dos algoritmos clássicos de AM, em seguida os do BERTimbau e por último os processos do EFC. Observe que nem todos os processos foram concluídos com sucesso, sendo no final obtidos 52 modelos em vez de 60 modelos. As falhas de treinamento ocorreram devido as limitações de memória da GPU.

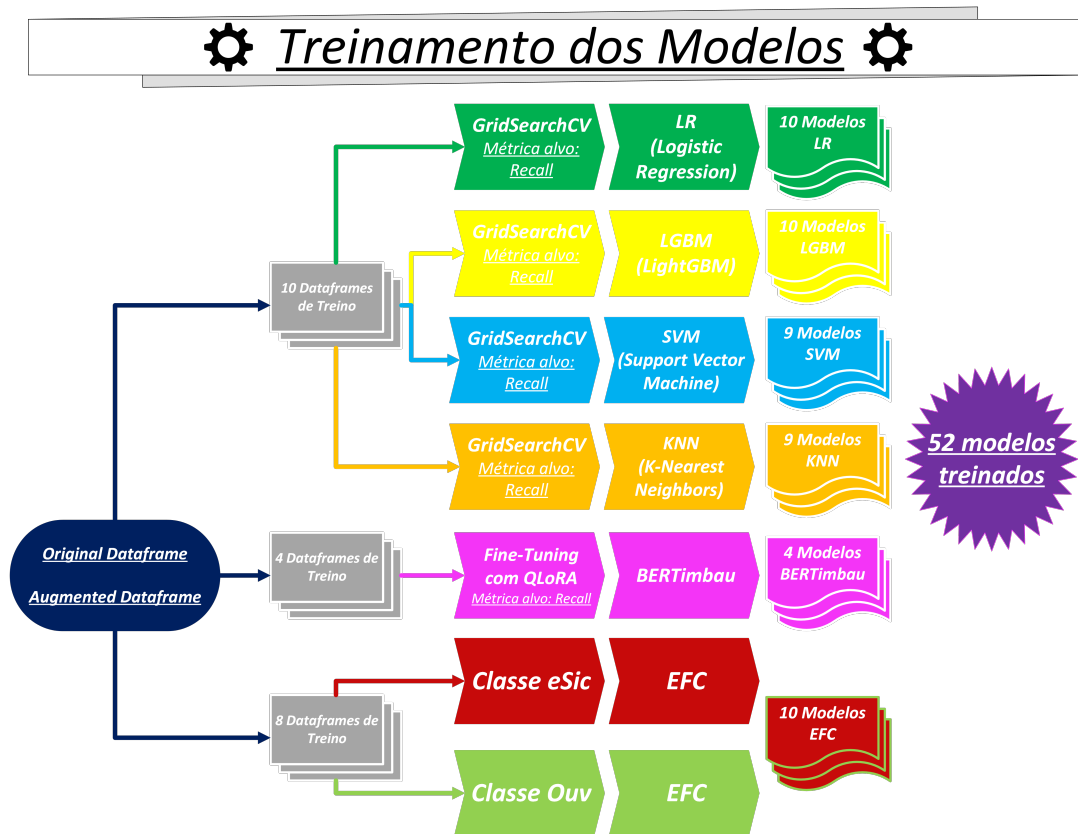


Figura 4.11: Treinamento dos Modelos

4.10 Avaliação do Modelo

Para comparar os modelos e selecionar aqueles que melhor atendem ao problema estudado, é necessário utilizar múltiplas métricas de avaliação simultaneamente e, posteriormente,

¹¹<https://huggingface.co/docs/transformers/quantization/bitsandbytes>

comparar os desempenhos dos modelos entre si ([41]). Na etapa de *Entendimento do Negócio*, foram definidos o propósito e as necessidades da aplicação, os quais orientam a escolha das métricas de avaliação definidas no item 3.7 *Métrica de Avaliação* do Capítulo 3. Dessa forma, as principais métricas selecionadas para a avaliação dos modelos treinados foram:

- ***Recall*** – No escopo deste trabalho, a correta classificação da classe minoritária com alta assertividade é fundamental. A métrica de *Micro Recall* reflete a necessidade de classificar corretamente os itens pertencentes a cada classe, especialmente as instâncias da classe minoritária. Assim, busca-se um modelo que apresente alto valor de *Recall* para a classe minoritária.
- ***Area Under the Curve - Receiver Operating Characteristic (AUC-ROC)***
 - A solução-alvo deste estudo de caso consiste na classificação binária automática dos textos de solicitações dos usuários da plataforma Participa DF. Um classificador binário deve apresentar desempenho superior ao acaso e igual ou superior à rotulagem fornecida pelos próprios usuários. A métrica AUC-ROC avalia a capacidade de um classificador binário em distinguir entre duas classes, como Ouv-DF e e-SIC DF. Assim, quanto maior o valor da AUC-ROC, melhor o desempenho do modelo.
- ***False Discovery Rate (FDR)*** – Após a rotulagem das solicitações na plataforma Participa DF, os servidores públicos analisam os registros para o devido tratamento. Nessa etapa, pode ser necessária a reclassificação do registro, o que corresponde à identificação de falsos positivos. Essa atividade é custosa e impactante, uma vez que o número de servidores responsáveis por ela é limitado. A métrica FDR mede a proporção de *falsos alarmes*, isto é, falsos positivos, em relação ao total de predições positivas. No contexto deste estudo, um baixo valor de FDR indica menor necessidade de reclassificação dos registros, permitindo que um maior número de solicitações seja processado no mesmo intervalo de tempo. Cabe ressaltar que a métrica FDR é o complementar da métrica *Precisão*.
- ***Acurácia*** – Avalia o desempenho geral do modelo em todas as previsões, sem distinguir entre classes majoritárias e minoritárias.
- ***F1-Score*** - combina as métricas *Precisão* e *Recall* por meio de média harmônica. É especialmente útil em cenários com classes desbalanceadas, penalizando os modelos que apresentam elevado número de falsos positivos, promovendo equilíbrio entre acertos e erros do classificador.

Capítulo 5

Resultados e Análises

5.1 Resultados

Os resultados obtidos na etapa de *Avaliação* dos modelos criados na fase anterior, item 4.9 *Modelagem* do Capítulo 4 (Figura 4.11), foram compilados na Tabela 5.1.

Cada linha da Tabela 5.1 representa os resultados obtidos para cada modelo treinado. A coluna "*Modelos Pipelines*" apresenta os nomes dos modelos. Esses nomes são compostos pelo algoritmo que gerou o modelo, seguido pelos nomes das atividades do *pipeline* a partir dos quais o modelo se originou. Assim, o acrônimo do nome do algoritmo aparece primeiro; seguido ou não pelo acrônimo "*NM*" (indicando subamostragem NearMiss); seguido ou não pelo acrônimo "*LPCA*" (vetorização baseada em lematização e redução de dimensionalidade via PCA); seguido ou não pelo acrônimo "*SSPCA*" (indicando vetorização baseada em estematização utilizando a técnica *SnowballStemmer* e redução de dimensionalidade via PCA); seguido ou não pelo acrônimo "*RSPCA*" (vetorização utilizando a técnica *RSLPStemmer* para redução de dimensionalidade por PCA); seguido ou não pelo acrônimo "*Aug*" (indicando sobreamostragem por meio de geração sintética de textos para redução do desbalanceamento); seguido ou não pelo acrônimo "*Orig*" (indicando ausência de sobreamostragem).

Uma exceção é feita para os nomes dos modelos derivados do EFC, nos quais o nome do algoritmo aparece primeiro, seguido pela classe utilizada no treinamento, depois opcionalmente pelo pré-processamento, após seguido ou não pelo acrônimo da técnica de redução de dimensionalidade e, por fim, pelo tipo de sobreamostragem que foi somente "*Orig*" (ausência de sobreamostragem).

As demais colunas da Tabela 5.1 representam as métricas de desempenho *Recall*, FDR, *Precisão*, *F1-Score*, *Acurácia* e AUC-ROC, escolhidas no item 4.10 *Avaliação do Modelo* do Capítulo 4. Algumas destas métricas foram calculadas por classe.

Tabela 5.1: Avaliação dos Modelos

Algoritmos Pipelines	Tempo de Execução	Recall		FDR		Precisão		F1-Score		Acurácia	ROC AUC
		e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF		
Modelos Energy-based Flow Classifier											
EFCeSic+Orig	23.94 s	72.20%	27.47%	94.88%	5.20%	5.12%	94.80%	9.56%	42.60%	29.77%	49.83%
EFCeSicPefc+Orig	2157.42 s	94.05%	34.83%	92.74%	0.92%	7.26%	99.08%	13.47%	51.54%	37.88%	64.44%
EFCeSic+LPCA+Orig	9.99 s	64.72%	66.38%	90.55%	2.80%	9.45%	97.20%	16.49%	78.88%	66.29%	65.55%
EFCeSic+SPCA+Orig	13.88 s	57.63%	73.36%	89.50%	3.04%	10.50%	96.96%	17.76%	83.53%	72.55%	65.49%
EFCeSic+RPCA+Orig	13.80 s	58.92%	72.48%	89.60%	2.98%	10.40%	97.02%	17.68%	82.97%	71.78%	65.70%
EFCOuv+Orig	144.26 s	24.68%	93.13%	83.69%	4.20%	16.31%	95.80%	19.64%	94.45%	89.61%	58.91%
EFCOuvPefc+Orig	35527.84 s	3.17%	93.89%	97.26%	5.30%	2.74%	94.70%	2.94%	94.30%	89.23%	48.53%
EFCOuv+LPCA+Orig	23.94 s	19.93%	84.55%	93.46%	4.88%	6.54%	95.12%	9.85%	89.52%	81.23%	52.24%
EFCOuv+SPCA+Orig	48.55 s	24.27%	75.60%	94.88%	5.15%	5.12%	94.85%	8.45%	84.14%	72.96%	49.94%
EFCOuv+RPCA+Orig	48.83 s	22.98%	75.74%	95.12%	5.22%	4.88%	94.78%	8.06%	84.20%	73.03%	49.36%
Modelos BERTimbau											
BERTimbau+Orig	10007.59 s	75.76%	98.49%	26.92%	1.32%	73.08%	98.68%	74.40%	98.59%	97.32%	87.12%
BERTimbau+Aug	23362.96 s	74.37%	98.61%	25.69%	1.39%	74.31%	98.61%	74.34%	98.61%	97.36%	86.49%
BERTimbau+NM+Orig	1149.96 s	94.66%	86.04%	73.12%	0.34%	26.88%	99.66%	41.87%	92.35%	86.48%	90.35%
BERTimbau+NM+Aug	10530.46 s	75.00%	98.43%	27.81%	1.36%	72.19%	98.64%	73.57%	98.54%	97.23%	86.72%
Modelos LightGBM											
LGBM+LPCA+Orig	15072.14 s	88.82%	92.36%	61.34%	0.65%	38.66%	99.35%	53.87%	95.73%	92.18%	90.59%
LGBM+LPCA+Aug	15088.60 s	82.53%	96.52%	43.77%	0.97%	56.23%	99.03%	66.88%	97.76%	95.80%	89.52%
LGBM+SSPCA+Orig	14800.19 s	87.94%	92.06%	62.49%	0.71%	37.51%	99.29%	52.59%	95.54%	91.85%	90.00%
LGBM+SSPCA+Aug	15115.52 s	80.82%	96.78%	42.32%	1.06%	57.68%	98.94%	67.32%	97.85%	95.96%	88.80%
LGBM+RSPCA+Orig	14882.14 s	87.13%	92.00%	62.88%	0.75%	37.12%	99.25%	52.06%	95.49%	91.75%	89.57%
LGBM+RSPCA+Aug	15141.56 s	81.24%	96.64%	43.25%	1.04%	56.75%	98.96%	66.82%	97.79%	95.85%	88.94%
LGBM+Orig	8351.68 s	73.44%	84.55%	79.51%	1.67%	20.49%	98.33%	32.04%	90.92%	83.98%	78.99%
LGBM+Aug	8283.07 s	29.56%	98.69%	45.04%	3.73%	54.96%	96.27%	38.44%	97.47%	93.13%	64.12%
LGBM+NM+Orig	8135.51 s	81.29%	9.45%	95.36%	9.69%	4.64%	90.31%	7.78%	17.12%	13.15%	45.37%
LGBM+NM+Aug	8355.94 s	41.93%	68.24%	93.32%	4.41%	6.68%	95.59%	11.53%	79.64%	66.89%	55.09%
Modelos Support Vector Machine											
SVM+LPCA+Orig	7710.06 s	59.99%	98.83%	26.51%	2.15%	73.49%	97.85%	66.06%	98.34%	96.83%	79.41%
SVM+LPCA+Aug	17059.10 s	82.26%	97.16%	38.88%	0.98%	61.12%	99.02%	70.13%	98.08%	96.40%	89.71%
SVM+SSPCA+Orig	7787.99 s	56.02%	98.96%	25.55%	2.35%	74.45%	97.65%	63.93%	98.30%	96.75%	77.49%
SVM+SSPCA+Aug	38209.49 s	64.60%	98.39%	31.52%	1.91%	68.48%	98.09%	66.48%	98.24%	96.65%	81.49%
SVM+RSPCA+Orig	8021.97 s	57.31%	98.96%	25.14%	2.29%	74.86%	97.71%	64.92%	98.33%	96.81%	78.13%
SVM+RSPCA+Aug	38763.30 s	68.23%	98.42%	29.91%	1.72%	70.09%	98.28%	69.14%	98.35%	96.87%	83.32%
SVM+Orig	2523.50 s	14.72%	99.72%	26.25%	4.43%	73.75%	95.57%	24.54%	97.60%	95.34%	57.22%
SVM+Aug	8799.60 s	16.13%	99.30%	44.42%	4.38%	55.58%	95.62%	25.00%	97.43%	95.02%	57.72%
SVM+NM+Orig	117.91 s	80.70%	8.75%	95.42%	10.68%	4.58%	89.32%	8.66%	15.95%	12.45%	44.73%
Modelos de Regressão Logística											
LR+LPCA+Orig	123.65 s	45.86%	99.35%	20.72%	2.87%	79.28%	97.13%	58.10%	98.23%	96.60%	72.60%
LR+LPCA+Aug	170.62 s	84.97%	94.77%	53.15%	0.85%	46.85%	99.15%	60.40%	96.91%	94.27%	89.87%
LR+SSPCA+Orig	126.85 s	47.78%	99.38%	19.43%	2.77%	80.57%	97.23%	59.99%	98.29%	96.72%	73.58%
LR+SSPCA+Aug	173.38 s	80.41%	95.46%	51.03%	1.10%	48.97%	98.90%	60.87%	97.15%	94.68%	87.93%
LR+RSPCA+Orig	125.66 s	47.59%	99.38%	19.36%	2.78%	80.64%	97.22%	59.85%	98.29%	96.72%	73.48%
LR+RSPCA+Aug	173.24 s	80.41%	95.23%	52.25%	1.10%	47.75%	98.90%	59.92%	97.03%	94.47%	87.82%
LR+Orig	42.86 s	12.96%	99.66%	32.49%	4.52%	67.51%	95.48%	21.75%	97.53%	95.20%	56.31%
LR+Aug	50.97 s	25.90%	98.44%	52.63%	3.92%	47.37%	96.08%	33.49%	97.25%	94.71%	62.17%
LR+NM+Orig	44.89 s	79.70%	15.06%	95.16%	6.81%	4.84%	93.19%	9.13%	25.93%	18.38%	47.38%
LR+NM+Aug	1176.62 s	37.84%	92.52%	78.47%	3.51%	21.53%	96.49%	27.45%	94.46%	89.71%	65.18%
Modelos K-Nearest Neighbors											
KNN+LPCA+Orig	1247.66 s	51.00%	97.79%	44.39%	2.64%	55.61%	97.36%	53.20%	97.57%	95.39%	74.40%
KNN+LPCA+Aug	2441.69 s	63.35%	94.71%	60.62%	2.05%	39.38%	97.95%	48.57%	96.30%	93.10%	79.03%
KNN+SSPCA+Orig	1207.34 s	53.56%	98.19%	38.41%	2.50%	61.59%	97.50%	57.29%	97.84%	95.89%	75.87%
KNN+SSPCA+Aug	2433.50 s	62.38%	87.20%	79.10%	2.29%	20.90%	97.71%	31.31%	92.16%	85.93%	74.79%
KNN+RSPCA+Orig	1209.08 s	55.90%	98.39%	34.72%	2.37%	65.28%	97.63%	60.23%	98.01%	96.20%	77.14%
KNN+RSPCA+Aug	2490.29 s	64.86%	95.20%	57.70%	1.96%	42.30%	98.04%	51.21%	96.60%	93.64%	80.03%
KNN+Orig	334.00 s	37.65%	71.80%	93.25%	4.50%	6.75%	95.50%	11.45%	81.97%	70.04%	54.72%
KNN+Aug	645.10 s	45.81%	89.61%	80.70%	3.17%	19.30%	96.83%	27.16%	93.08%	87.36%	67.71%
KNN+NM+Orig	123.87 s	100.00%	0.00%	94.86%	0.00%	5.14%	0.00%	9.78%	0.00%	5.14%	50.00%

O *Energy-based Flow Classifier (EFC)* é o algoritmo foco deste estudo de caso. Por se tratar de um Classificadores de Uma Classe (CIUC), o EFC treina cada modelo utilizando apenas uma classe rotulada. Assim, para cada *pipeline* de pré-processamento, dois modelos foram treinados. 8 *pipelines* de pré-processamento foram utilizados, totalizando 16 treinamentos realizados (Figura 4.10 e Figura 4.11). Destes treinamentos foram obtidos de fato 10 modelos:

- *EFC_eSic+Orig* (Matriz de Confusão - Tabela 5.2)
- *EFC_Ouv+Orig* (Matriz de Confusão - Tabela 5.3);
- *EFC_eSic+LPCA+Orig* (Matriz de Confusão - Tabela 5.4)
- *EFC_Ouv+LPCA+Orig* (Matriz de Confusão - Tabela 5.5);
- *EFC_eSic+SPCA+Orig* (Matriz de Confusão - Tabela 5.6)

- *EFC_Ouv+SPCA+Orig* (Matriz de Confusão - Tabela 5.7);
- *EFC_eSic+RPCA+Orig* (Matriz de Confusão - Tabela 5.8)
- *EFC_Ouv+RPCA+Orig* (Matriz de Confusão - Tabela 5.9);
- *EFC_eSic+Pefc+Orig* (Matriz de Confusão - Tabela 5.10);
- *EFC_Ouv+Pefc+Orig* (Matriz de Confusão - Tabela 5.11).

Tabela 5.2: EFC+TFIDF – Classe e-SIC DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	2963	1141
Ouv-DF	54904	20796

Tabela 5.3: EFC+TFIDF – Classe Ouv-DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	1013	3091
Ouv-DF	5197	70503

Tabela 5.4: EFC+TFIDF+LPCA – Classe e-SIC DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	2656	1448
Ouv-DF	25452	50248

Devido as limitações de memória da GPU, 6 *pipelines* de treinamento não foram concluídas:

- *EFC_eSic+Pefc+LPCA+Orig*;
- *EFC_Ouv+Pefc+LPCA+Orig*;
- *EFC_eSic+Pefc+SPCA+Orig*;
- *EFC_Ouv+Pefc+SPCA+Orig*;
- *EFC_eSic+Pefc+RPCA+Orig*;
- *EFC_Ouv+Pefc+RPCA+Orig*.

Tabela 5.5: EFC+TFIDF+LPCA – Classe Ouv-DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	818	3286
Ouv-DF	11695	64005

Tabela 5.6: EFC+TFIDF+SPCA – Classe e-SIC DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	2365	1739
Ouv-DF	20166	55534

Observou-se que quanto maior o grau de pré-processamento (como remoção de caracteres especiais, pontuação e espaços extras), mais rápido foi o treinamento do modelo EFC, porém, menor foi o desempenho, conforme apresentado na Tabela 5.1. Por exemplo, o modelo *EFC_eSic+LPCA+Orig* foi treinado com a classe e-SIC DF, mais um maior nível de pré-processamento (remoção de *stop words*, limitação às 5000 palavras mais frequentes, remoção de caracteres especiais e pontuação), mais lematização e PCA. Este modelos apresentou tempo de treinamento de 9,99s, mas obteve um *Recall* para a classe e-SIC DF de 64,72%. Por outro lado, o modelo *EFC_eSic+Pefc+Orig*, também treinado apenas com a classe e-SIC DF, porém com menor nível de pré-processamento (somente remoção de *stop words*, limitação às 5000 palavras mais frequentes), apresentou um *Recall* mais elevado para a classe e-SIC DF, 94,05%, mas com um tempo de treinamento significativamente maior de 2157,42s.

Quando treinado com amostras submetidas a menor pré-processamento, o EFC alcançou valores de *Recall* superiores a 90% para a classe utilizada no treinamento, indicando forte capacidade de identificação de amostras desta classe, conforme evidenciado nas Matrizes de Confusão: Tabela 5.2 e Tabela 5.3. Evidenciando também, alta sensibilidade do algoritmo à redução de informação por pré-processamento.

Considerando apenas a classe e-SIC DF, o modelo EFC com melhor desempenho foi o *EFC_eSic+Pefc+Orig*, com *Recall* e-SIC DF de 94,05% e tempo de treinamento de 2157,42s. Contudo, esse modelo apresentou *Recall* Ouv-DF de 34,83%, FDR e-SIC DF de 92,74%, FDR Ouv-DF de 0,92%, Precisão e-SIC DF de 7,26%, Precisão Ouv-DF de 99,08%, *F1-Score* e-SIC DF de 13,47%, *F1-Score* Ouv-DF de 51,54%, Acurácia de 37,88% e AUC-ROC de 64,44%.

Para comparar o desempenho dos demais modelos treinados com os modelos derivados do EFC, foram selecionadas as métricas que melhor caracterizam o "*Problema de Negócio*" a ser resolvido no contexto do Participa DF, conforme detalhado na seção 4.10 *Avaliação do Modelo* do Capítulo 4: *Recall*, FDR, *F1-Score*, Acurácia e AUC-ROC. Dado o contexto

Tabela 5.7: EFC+TFIDF+SPCA – Classe Ouv-DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	996	3108
Ouv-DF	18469	5723

Tabela 5.8: EFC+TFIDF+RPCA – Classe e-SIC DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	2418	1686
Ouv-DF	20834	54866

do estudo, foi dada ênfase especial à rotulagem correta da classe minoritária e-SIC DF, que é inerentemente mais difícil de classificar com alta exatidão. Nesse cenário, os modelos foram considerados de alto desempenho quando apresentaram valores elevados de *Recall* e-SIC DF, *F1-Score* e-SIC DF, Acurácia e AUC-ROC, bem como baixos valores de FDR e-SIC DF.

Os modelos que alcançaram os maiores valores de *Recall* e-SIC DF foram agrupados na Tabela 5.12. Dentre eles, destacaram-se: *KNN+NM+Orig* com 100,00%, *BERTimbau+NM+Orig* com 94,66% e *EFC_eSic+Pefc+Orig* com 94,05%. Em relação ao menor FDR e-SIC DF, os modelos com melhor desempenho (do menor para o maior valor) foram: *LR+RSPCA+Orig* com 19,36%, *LR+SSPCA+Orig* com 19,43% e *LR+LPCA+Orig* com 20,72%. No que se refere à Acurácia, os modelos com melhor desempenho foram: *BERTimbau+Aug* com 97,36%, *BERTimbau+Orig* com 97,32% e *BERTimbau+NM+Aug* com 97,23%. Em relação ao *F1-Score* e-SIC DF os modelos com melhor desempenho foram: *BERTimbau+Orig* com 74,40%, *BERTimbau+Aug* com 74,34% e *BERTimbau+NM+Aug* com 73,57%. Quanto à métrica AUC-ROC, os modelos com maiores valores foram: *LGBM+LPCA+Orig* com 90,59%, *BERTimbau+NM+Orig* com 90,35% e *LGBM+SPCA+Orig* com 90,00%.

No que diz respeito à métrica *AUC-ROC*, ela fornece um resumo da curva ROC, permitindo avaliar o comportamento do modelo sob diferentes limiares de decisão ([37]). A Figura 5.1 apresenta essa análise, exibindo as curvas ROC sobrepostas dos modelos listados na Tabela 5.1. Por meio dessa visualização, é possível comparar o desempenho dos modelos na classificação binária neste contexto e observar se ele se aproxima de um desempenho aleatório (linha pontilhada preta). Observando as curvas de desempenho na Figura 5.1, os modelos *SVM+Aug* (AUC-ROC 44,73%), *LGBM+NM+Orig* (AUC-ROC 45,37%), *LR+NM+Orig* (AUC-ROC 47,38%), *EFC_Ouv+Pefc+Orig* (AUC-ROC 48,53%), *EFC_eSic+Orig* (AUC-ROC 49,83%) e *KNN+NM+Orig* (AUC-ROC 50,00%) foram aqueles cujas curvas ROC mais se aproximaram do desempenho aleatório (linha

Tabela 5.9: EFC+TFIDF+RPCA – Classe Ouv-DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	943	3161
Ouv-DF	18362	57338

Tabela 5.10: EFC+TFIDF+Pefc – Classe e-SIC DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	3860	244
Ouv-DF	49333	26367

pontilhada preta). Contudo, é importante destacar que a AUC-ROC pode ser enganosa para conjuntos de dados altamente desbalanceados, apesar de seu uso disseminado na avaliação de classificadores binários.

Ao comparar conjuntamente as métricas relevantes para este estudo de caso, os dez modelos que simultaneamente apresentaram os menores valores de FDR e-SIC DF e os maiores valores de *Recall* e-SIC DF, Acurácia, AUC-ROC e *F1-Score* e-SIC DF foram agrupados e ranqueados por esta última métrica na Tabela 5.13. Ressalta-se que, para facilitar o agrupamento dos modelos, o *F1-Score* da classe e-SIC DF foi utilizado, uma vez que ele corresponde à média harmônica entre *Recall* e-SIC DF e Precisão e-SIC DF, sendo a Precisão e-SIC DF o complementar do FDR e-SIC DF. Dessa forma, o *F1-Score* e-SIC DF sintetiza o equilíbrio entre alcançar alta Precisão (baixo FDR) e alto *Recall* simultaneamente, para uma rotulagem eficaz da classe minoritária. Portanto os modelos que simultaneamente alcançaram os melhores valores de *F1-Score* e-SIC DF, *Recall* e-SIC DF, FDR e-SIC DF, Acurácia e AUC-ROC foram: *BERTimbau+Orig*, *BERTimbau+Aug*, *BERTimbau+NM+Aug* e *SVM+LPCA+Aug* (Tabela 5.13).

Ainda nesse contexto, ao excluir os modelos baseados em *transformers* e aprendizado profundo, o modelo *SVM+LPCA+Aug* apresentou o melhor desempenho geral entre os classificadores clássicos. Ele obteve o maior *F1-Score* e-SIC DF dentre todos os classificadores clássicos (70,13%), além de uma combinação particularmente favorável de *Recall* e-SIC DF (82,26%), FDR e-SIC DF (38,88%), Acurácia (96,40%) e AUC-ROC (89,71%). As duas últimas métricas figuram entre as mais elevadas quando considerados apenas os modelos clássicos. A Tabela 5.14 apresenta os hiperparâmetros de treino dos modelos citados na Tabela 5.13.

Tabela 5.11: EFC+TFIDF+Pefc – Classe Ouv-DF – Sem texto Sintético

Observações	Previsões do Modelo	
	e-SIC DF	Ouv-DF
e-SIC DF	130	3974
Ouv-DF	4623	71077

Tabela 5.12: Top 10 modelos por Recall no e-SIC DF

Algoritmos Pipelines	Recall		FDR		F1-Score		Acurácia	ROC AUC	Tempo de Execução
	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF			
KNN+NM+Orig	100.00%	0.00%	94.86%	0.00%	9.78%	0.00%	5.14%	50.00%	123.87 s
BERTimbau+NM+Orig	94.66%	86.04%	73.12%	0.34%	41.87%	92.35%	86.48%	90.35%	1149.96 s
EFCeSicPefc+Orig	94.05%	34.83%	92.74%	0.92%	13.47%	51.54%	37.88%	64.44%	2157.42 s
LGBM+LPCA+Orig	88.82%	92.36%	61.34%	0.65%	53.87%	95.73%	92.18%	90.59%	15072.14 s
LGBM+SSPCA+Orig	87.94%	92.06%	62.49%	0.71%	52.59%	95.54%	91.85%	90.00%	14800.19 s
LGBM+RSPCA+Orig	87.13%	92.00%	62.88%	0.75%	52.06%	95.49%	91.75%	89.57%	14882.14 s
LR+LPCA+Aug	84.97%	94.77%	53.15%	0.85%	60.40%	96.91%	94.27%	89.87%	170.62 s
LGBM+LPCA+Aug	82.53%	96.52%	43.77%	0.97%	66.88%	97.76%	95.80%	89.52%	15088.60 s
SVM+LPCA+Aug	82.26%	97.16%	38.88%	0.98%	70.13%	98.08%	96.40%	89.71%	17059.10 s
LGBM+NM+Orig	81.29%	9.45%	95.36%	9.69%	8.78%	17.12%	13.15%	45.37%	8135.51 s

5.2 Análises

Este trabalho aplicou o algoritmo EFC para classificar os textos contidos em demandas dos cidadãos submetidas ao Participa DF, e comparou seu desempenho com outras famílias de modelos. O estudo investigou técnicas de Aprendizagem de Máquina (AM) voltadas ao aprimoramento da interação com a população do Distrito Federal, por meio do aumento da produtividade na prestação dos serviços públicos do Participa DF [7], [8], [3], [4].

As demandas recebidas por esse canal de participação constituem um conjunto de dados desbalanceado, com casos especiais pertencentes à classe minoritária — a saber, solicitações de acesso à informação —, as quais exigem atenção ampliada e devem ser classificadas com elevada precisão. Entretanto, classes minoritárias, em geral, são rotuladas de forma inadequada por modelos de Aprendizagem de Máquina, apesar de conjuntos de dados desbalanceados serem comuns em cenários do mundo real [13].

Neste estudo, o algoritmo EFC, ao treinar um modelo exclusivamente com a classe e-SIC DF e com pré-processamento reduzido, gerou o modelo *EFC_eSic+Pefc+Orig*, que se posicionou como o terceiro melhor na métrica de *Recall* para e-SIC DF (Tabela 5.12). Esse modelo apresentou tempo de treinamento de 2157.42 s, *Recall* Ouv-DF de 34.83%, FDR e-SIC DF de 92.74%, FDR Ouv-DF de 0.92%, Precisão e-SIC DF de 7.26%, Precisão Ouv-DF de 99.08%, *F1-Score* e-SIC DF de 13.47%, *F1-Score* Ouv-DF de 51.54%, Acurácia de 37.88% e AUC-ROC de 64.44%.

Embora o EFC tenha produzido um modelo capaz de identificar de forma eficaz a classe minoritária e-SIC DF, o modelo *EFC_eSic+Pefc+Orig* apresentou uma FDR elevada (92.74%), isto é, inferiu muitos falsos positivos (de um total de 53.193 inferências para a classe e-SIC DF, 49333 eram falsos positivos e somente 3860 eram verdadeiros positivos Tabela 5.10) e implicaria um alto volume de reclassificações e trabalho para os servidores

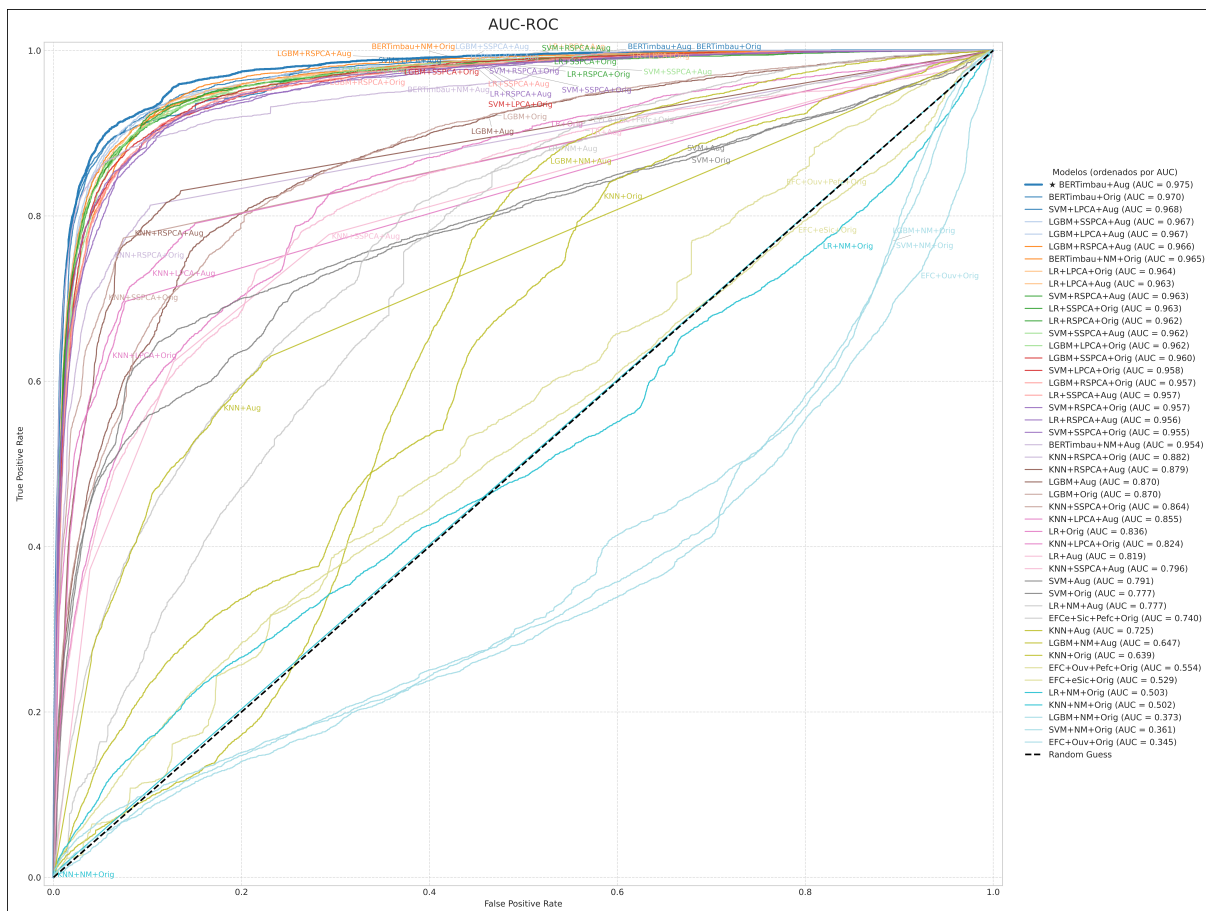


Figura 5.1: Curvas ROC

da CGDF. Dessa forma, os modelos derivados do algoritmo BERTimbau, listados no topo da Tabela 5.13, seriam os mais adequados neste contexto.

Contudo, fatores adicionais também devem ser considerados na escolha de um modelo para a solução do "*Problema de Negócio*", como as restrições do ambiente de produção do Participa DF. Além das limitações de infraestrutura de treinamento descritas no Capítulo 4, o ambiente de produção apresenta também restrições operacionais para realizar a classificação automática de solicitações direcionadas ao e-SIC DF ou à Ouv-DF, como limitações de memória e armazenamento. Por se tratar de um serviço governamental *on-line*, a classificação deve ocorrer com tempo de resposta quase instantâneo, a fim de evitar a introdução de atrasos perceptíveis aos usuários, o que pode resultar em abandono do serviço e redução do engajamento cidadão. O portal recebe aproximadamente 30.000 registros por mês — cerca de 1.000 por dia —, impondo requisitos rigorosos de latência, responsividade e estabilidade operacional na execução desse serviço crítico.

Embora os três melhores modelos da Tabela 5.13 tenham apresentado desempenho superior nas métricas-alvo, eles pertencem à família BERTimbau e, portanto, demandam

Tabela 5.13: Top 10 Modelos ranqueados pelos melhores valores de $F1-Score$ (e-SIC DF), Recall (e-SIC DF), FDR (e-SIC DF), Acurácia e AUC-ROC

Algoritmos Selecionados	F1-Score		Recall		FDR		Acurácia	ROC AUC	Tempo de Treinamento
	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF	e-SIC DF	Ouv-DF			
Modelos BERTimbau									
BERTimbau+Orig	74.40%	98.59%	75.76%	98.49%	26.92%	1.32%	97.32%	87.12%	10007.60 s
BERTimbau+Aug	74.34%	98.61%	74.37%	98.61%	25.69%	1.39%	97.36%	86.49%	23362.96 s
BERTimbau+NM+Aug	73.57%	98.54%	75.00%	98.43%	27.81%	1.36%	97.23%	86.72%	10530.46 s
Modelos SVM									
SVM+LPCA+Aug	70.13%	98.08%	82.26%	97.16%	38.88%	0.98%	96.40%	89.71%	17059.10 s
Modelos LightGBM									
LGBM+SSPCA+Aug	67.32%	97.85%	80.82%	96.78%	42.32%	1.06%	95.96%	88.80%	15115.51 s
LGBM+LPCA+Aug	66.88%	97.76%	82.53%	96.52%	43.77%	0.97%	95.80%	89.52%	15088.60 s
LGBM+RSPCA+Aug	66.82%	97.79%	81.24%	96.64%	43.25%	1.04%	95.85%	88.94%	15141.56 s
Modelos de Regressão Logística									
LR+SSPCA+Aug	60.87%	97.15%	80.41%	95.46%	51.03%	1.10%	94.68%	87.93%	173.37 s
LR+LPCA+Aug	60.40%	96.91%	84.97%	94.77%	53.15%	0.85%	94.27%	89.87%	170.62 s
LR+RSPCA+Aug	59.92%	97.03%	80.41%	95.23%	52.25%	1.10%	94.47%	87.82%	173.24 s

alto Custo Computacional durante a inferência, resultando em maior latência do que modelos clássicos, além de maiores requisitos de armazenamento ([35] e [43]). Esses fatores tornariam, em princípio, inviável sua implantação nos servidores governamentais do Participa DF, os quais dispõem de recursos limitados.

Em contraste, modelos SVM apresentam latência de inferência substancialmente menor, operando de forma eficiente e requerendo apenas alguns megabytes de armazenamento ([35] e [43]). Essas características, em geral, permitem que o sistema suporte altos volumes de solicitações simultâneas com tempos de resposta compatíveis com aplicações interativas de uso contínuo. Assim, pelos motivos expostos, o modelo *SVM+LPCA+Aug* ($F1-Score$ e-SIC DF 70,13%, $Recall$ e-SIC DF 82,26%, FDR e-SIC DF 38.88%, Acurácia 96,40%, AUC-ROC 89,71%) emerge como a alternativa tecnicamente mais viável para o ambiente de produção do Participa DF.

Os hiperparâmetros do modelo *SVM+LPCA+Aug* foram os seguintes:

- *pca__n_components* = 500,
- *svm__C* = 100,
- *svm__gamma* = 1,
- *svm__kernel* = rbf.

Por fim, cabe ressaltar que não foi possível concluir o treinamento dos modelos *KNN+NM+Aug* e *SVM+NM+Aug*, juntamente dos modelos EFC anteriormente citados (*EFC_eSic+Pefc+LPCA+Orig*, *EFC_Ouv+Pefc+LPCA+Orig*, *EFC_eSic+Pefc+SPCA+Orig*, *EFC_Ouv+Pefc+SPCA+Orig*, *EFC_eSic+Pefc+RPCA+Orig*, *EFC_Ouv+Pefc+RPCA+Orig*), devido as limitações de memória da GPU usada para treinamento.

Tabela 5.14: Resumo dos hiperparâmetros utilizados no treinamento dos modelos da Tabela 5.13

Modelo	Hiperparâmetros
Modelos BERTimbau	
BERTimbau+Orig	QLoRA4b(nf4,dq,fp16); LoRA(r16,a32,d0.1,qkv); L128, ep(10), lr(5e-5) bs(16); opt(paged_adamw_8bit); ES2, seed(1917)
BERTimbau+Aug	QLoRA4b(nf4,dq,fp16); LoRA(r16,a32,d0.1,qkv); L128, ep(10) lr(5e-5) bs(16); opt=paged_adamw_8bit; ES2, seed(1917)
BERTimbau+NM+Aug	TF-IDF(5k)+NMv2(k3); QLoRA4b(nf4,dq,fp16); LoRA(r16,a32,d0.1,qkv); L128, ep(10), lr(5e-5), bs(16); opt=paged_adamw_8bit; pad8; ES2, seed(1917)
Modelos SVM	
SVM+LPCA+Aug	PCA(500)+SVM(rbf, $\gamma=1$)
Modelos LightGBM	
LGBM+SSPCA+Aug	PCA(500)+LGBM(lr 0.1, est 500)
LGBM+LPCA+Aug	PCA(500)+LGBM(lr 0.1, est 500)
LGBM+RSPCA+Aug	PCA(500)+LGBM(lr 0.1, est 500)
Modelos de Regressão Logística	
LR+SSPCA+Aug	PCA(500)+LR(saga, it 10000)
LR+LPCA+Aug	PCA(500)+LR(saga, it 15000)
LR+RSPCA+Aug	PCA(500)+LR(saga, it 5000)

Síglas: PCA = Análise de Componentes Principais; QLoRA4b = QLoRA quantização em 4 bits; nf4 = NormalFloat4; dq = *double quantization*; fp16 = precisão float16; LoRA(r,a,d,qkv) = *Low-Rank Adaptation* com rank r , escala α , dropout d aplicado a query/key/value; L128 = comprimento máximo de sequência 128; ep = épocas; lr = taxa de aprendizado; bs = *batch size*; opt = otimizador; paged_adamw_8bit = AdamW paginado em 8 bits (bitsandbytes); ES2 = *early stopping* com paciência 2; seed = semente aleatória; Aug = dados com aumento (*data augmentation*); TF-IDF(5k) = TF-IDF com 5000 atributos; NMv2(k3) = NM versão 2 com $k=3$ vizinhos; pad8 = *padding* para múltiplo de 8; rbf = kernel radial; γ = parâmetro do kernel; est = número de estimadores (árvores); LR(saga,it) = Regressão Logística com solver SAGA.

Capítulo 6

Considerações Finais

Como desdobramento deste estudo, deu-se origem a um artigo científico baseado nos resultados apresentados nesta dissertação, o qual foi submetido à avaliação de um congresso de reconhecida importância acadêmica e referência na área de estudo.

6.1 Conclusões

A hipótese apresentada inicialmente no Capítulo 1 foi:

"A técnica de Aprendizagem de Máquina Energy-based Flow Classifier ([19], [16], [17]) apresenta uma assertividade maior para a rotulação de exemplares minoritários, classes de baixa representatividade no conjunto de dados, ou seja, desbalanceadas, na métrica Recall da classe minoritária."

Revisitando a Tabela 5.12, verificamos que dentre os 10 modelos que apresentaram os melhores *Recall* para a classe minoritária e-SIC DF, dos 46 modelos treinados, o EFC gerou o terceiro modelo com o melhor *Recall* e-SIC DF. Sendo somente superado pelo modelo *KNN+NM+Orig* (*Recall* e-SIC DF 100%) e o modelo *BERTimbau+NM+Orig* (*Recall* e-SIC DF 94,66%). Contudo, apesar do modelo *KNN+NM+Orig* ter apresentado o melhor *Recall* e-SIC DF (100%), ele obteve um AUC-ROC de 50.00% e Acurácia de 5.14%, o que equivale a um classificador aleatório. Portanto, o modelo *KNN+NM+Orig* deve ser desconsiderado na análise dos melhores *Recall* e-SIC DF. Assim, o EFC apresentou o segundo modelo com o melhor *Recall* e-SIC DF dentre os 46 modelos treinados, sendo superado somente pelo modelo *BERTimbau+NM+Orig* (*Recall* e-SIC DF 94,66%). E o EFC alcançou o melhor *Recall* e-SIC DF dentre os modelos clássicos de Aprendizagem de Máquina (AM).

Logo, a hipótese apresentada inicialmente foi parcialmente confirmada. O EFC mostrou-se promissor para detecção de amostras da classe minoritária em tarefas de Classificação

de Textos Desbalanceados (CTD), e portanto sendo viável a sua aplicação em um novo domínio Processamento de Linguagem Natural (PLN).

Contudo, ele não se apresentou como a melhor opção para o presente estudo de caso, considerando os modelos que simultaneamente alcançaram os melhores valores de *Recall* e-SIC DF, FDR e-SIC DF, Acurácia e AUC-ROC, e os maiores *F1-Scores* e-SIC DF Tabela 5.13.

Outro fato a ser destacado observando a Tabela 5.13, que apresenta os modelos que melhor atenderiam ao presente estudo de caso, é que a maioria dos modelos desta tabela que apresentaram os melhores desempenhos usaram algum pré-processamento dos textos de entrada, algum tipo de redimensionamento (Subamostragem por NearMiss (NM) ou sobreamostragem por textos sintéticos - *Aug*), sendo que a maioria usou sobreamostragem por textos sintéticos (*Aug*).

6.2 Trabalhos Futuros

Em pesquisas futuras, serão exploradas outras abordagens para aprimorar a classificação de textos em cenários desbalanceados, tais como: testar variações do hiperparâmetro *cutoff* (limiar estatístico) do EFC para reduzir Falsos Positivos sem impactar tanto o *Recall*; comparar o EFC com outros Classificadores de Uma Classe (CIUC) ([13]); o uso de cascatas de modelos (*model cascades*), que têm demonstrado resultados promissores [13]; métodos alternativos de vetorização, como o *Term Frequency - Inverse Gravity Moment (TF-IGM)* com *Relative Imbalance Ratio (RIR)* [18] ou o *CluWords*, que combina métodos baseados em frequência (por exemplo, *Term Frequency-Inverse Document Frequency*) com informações semânticas provenientes de *word embeddings*, agrupando palavras semanticamente relacionadas e aplicando esquemas de ponderação adaptados ao domínio ([36]); a utilização de modelos de aprendizado profundo destilados para classificação de textos, como o DistilBERTimbau ([39]); e o emprego de outras métricas de avaliação para modelos de classificação com dados altamente desbalanceados, como a Área sob a Curva Precisão-Revocação (*Area Under the Precision-Recall Curve - AUPRC*), a qual na avaliação de modelos treinados com dados muito desbalanceados é considerada mais eficaz para avaliar o desempenho destes classificadores binários do que a métrica AUC-ROC, uma vez que a métrica AUC-ROC falha em registrar os escores de precisão e a quantidade de falsos positivos, dados que a AUPRC evidencia claramente ([37]).

Na avaliação de modelos com dados muito desbalanceados, a métrica AUPRC supera a AUC. Isso ocorre porque a AUC falha em registrar os escores de precisão e a quantidade de falsos positivos, dados que a AUPRC evidencia claramente

Referências

- [1] Martínez-Plumed, Fernando, Lidia Contreras-Ochando, Cèsar Ferri, José Hernández-Orallo, Meelis Kull, Nicolas Lachiche, María José Ramírez-Quintana e Peter Flach: *Crisp-dm twenty years later: From data mining processes to data science trajectories*. IEEE Transactions on Knowledge and Data Engineering, 33(8):3048–3061, 2021. xii, 6, 14, 15, 16, 17, 18, 33
- [2] Campos, Anna Maria: *Accountability: quando poderemos traduzi-la para o português?* Revista de Administração Pública, 24:30 a 50, jun. 1990. <https://periodicos.fgv.br/rap/article/view/9049>. 1
- [3] Souza, J. J.: *A necessidade de regulação do uso da inteligência artificial nas ações de controle do tribunal de contas da união*. Dissertação de mestrado, Centro Universitário Internacional – UNINTER, Curitiba, 2021. <https://repositorio.uninter.com/handle/1/1229>. 1, 3, 5, 8, 11, 56
- [4] Bitencourt, Caroline Müller e Luisa Helena Nicknig Martins: *A inteligência artificial nos órgãos constitucionais de controle de contas da administração pública brasileira*. Revista de Investigações Constitucionais, 10(3):e253, 2023. <https://revistas.ufpr.br/rinc/article/view/e253>. 1, 3, 5, 8, 11, 56
- [5] Grin, E. J.: *Governança participativa, eficácia gerencial e legitimidade política nos governos locais*. Em *XIII Congreso Internacional del CLAD sobre la Reforma del Estado y de la Administración Pública*, Buenos Aires, novembro 2008. 1
- [6] Rover, A. J., F. Galindo e O. Mezzaroba: *Direito, governança e tecnologia: princípios, políticas e normas do Brasil e da Espanha*. Conceito Editorial, Santa Catarina, Brazil, 2014. 1
- [7] Valle, V. L. López e R. M. Cabral: *Administração pública digital e a implementação dos objetivos do desenvolvimento sustentável*. Revista Eletrônica Direito e Política, 17(1):187–225, 2022. <https://periodicos.univali.br/index.php/rdp/article/view/18656>. 1, 5, 8, 11, 56
- [8] Desordi, D. e C. D. Bona: *A inteligência artificial e a eficiência na administração pública*. Revista de Direito, 12(2):1–22, 2020. <https://periodicos.ufv.br/revistadir/article/view/9112>. 1, 5, 8, 11, 56
- [9] Governo do Brasil: *Estratégia de governo digital 2020-2022*, 2020. https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/decreto/d10332.htm, Accessed: 2025-06-28. 2, 3

- [10] Noguti, Mariana Y., Eduardo Vellasques e Luiz S. Oliveira: *Legal document classification: An application to law area prediction of petitions to public prosecution service*. Em *2020 International Joint Conference on Neural Networks (IJCNN)*, páginas 1–8, 2020. 2, 8, 15, 22, 33
- [11] Cordeiro, Fábio, Ricardo Rabelo e Raimundo Moura: *Classification of irregularity communications in public ombudsmen using supervised learning algorithms*. Em *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional*, páginas 704–715, Porto Alegre, RS, Brasil, 2022. SBC. <https://sol.sbc.org.br/index.php/eniac/article/view/22825>. 2, 4, 9, 12, 15, 16, 24, 25, 28, 29, 33, 37, 43
- [12] Bollotti, Joelson Júnior e Marcos Wachowicz: *A aplicação da inteligência artificial pela administração pública diante do princípio da eficiência*. REVISTA DA AGU, 23(4), dez. 2024. <https://revistaagu.agu.gov.br/index.php/AGU/article/view/3429>. 3, 5
- [13] Tsai, Chih Fong e Wei Chao Lin: *Feature selection and ensemble learning techniques in one-class classifiers: An empirical study of two-class imbalanced datasets*. IEEE Access, 9:13717–13726, 2021. 3, 4, 5, 10, 15, 16, 29, 38, 56, 61
- [14] Mifsud, Ruth, Lipika Deka e Indrani Lahiri: *An optimised bert pretraining approach for identification of targeted offensive language: Data imbalance and potential solutions*. Em *2023 4th International Conference on Computing and Communication Systems (I3CS)*, páginas 1–8, 2023. 3
- [15] Akkaradamrongrat, Suphamongkol, Pornpimon Kachamas e Sukree Sinthupinyo: *Text generation for imbalanced text classification*. Em *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, páginas 181–186, 2019. 3
- [16] Pontes, Camila F. T., Manuela M. C. de Souza, João J. C. Gondim, Matt Bishop e Marcelo Antonio Marotta: *A new method for flow-based network intrusion detection using the inverse potts model*. IEEE Transactions on Network and Service Management, 18(2):1125–1136, 2021. 3, 4, 5, 10, 11, 15, 16, 37, 45, 60
- [17] DeAlmeida, Jonathan M., Camila F. T. Pontes, Luiz A. DaSilva, Cristiano B. Both, João J. C. Gondim, Celia G. Ralha e Marcelo A. Marotta: *Abnormal behavior detection based on traffic pattern categorization in mobile networks*. IEEE Transactions on Network and Service Management, 18(4):4213–4224, 2021. 3, 4, 5, 10, 60
- [18] Okkalioglu, Murat: *Tf-igm revisited: Imbalance text classification with relative imbalance ratio*. Expert Systems with Applications, 217:119578, 2023, ISSN 0957-4174. <https://www.sciencedirect.com/science/article/pii/S0957417423000799>. 4, 29, 61
- [19] Pontes, Camila Ferreira Thé: *Um novo método baseado no modelo de potts para detecção de intrusão em rede*, 2020. <https://bdm.unb.br/handle/10483/27586>. 4, 5, 10, 26, 60

- [20] Mariano, Ari Melo e Maíra Rocha Santos: *Revisão da literatura: Apresentação de uma abordagem integradora*. Em *XXVI Congresso Internacional AEDEM / 2017 AEDEM International Conference – Economy, Business and Uncertainty: Ideas for a European and Mediterranean Industrial Policy*, Reggio Calabria, Itália, setembro 2017. 7
- [21] Souza, Raul Carvalho de: *Uma comparação entre métodos e classificadores em documentos jurídicos de atividades processuais repetitivas na pgdf*. Dissertação de mestrado profissional, Universidade de Brasília, Brasília, 2021. 8, 15, 16, 22, 24, 29, 33
- [22] Dos Santos, Dário P., João Paulo J. Da Costa, Daniel A. Da Silva, Fábio Mendonça, Carlos Veiga e Rafael T. De Sousa: *Multi-class text classification based in oversampling for highly imbalanced dataset*. Em *2023 International Conference on Machine Learning and Applications (ICMLA)*, páginas 752–755, 2023. 11, 15, 16, 22, 23, 25, 29, 33
- [23] Sehgal, Shruti, Harpreet Singh, Mohit Agarwal, V. Bhasker e Shantanu: *Data analysis using principal component analysis*. Em *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, páginas 45–48, 2014. 11, 25, 41, 42
- [24] Reusens, Manon, Alexander Stevens, Jonathan Tonglet, Johannes De Smedt, Wouter Verbeke, Seppe vanden Broucke e Bart Baesens: *Evaluating text classification: A benchmark study*. Expert Systems with Applications, 254:124302, 2024, ISSN 0957-4174. <https://www.sciencedirect.com/science/article/pii/S0957417424011680>. 12, 15, 16, 22
- [25] Joshi, Deepali, Aryan Shinde, Shreya Das, Om Deokar, Dipasha Shetiya e Simran Jagtap: *Text data augmentation*. Em *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, páginas 392–396, 2023. 12, 15, 16, 25, 28, 43
- [26] Sokolova, Marina e Guy Lapalme: *A systematic analysis of performance measures for classification tasks*. Information Processing & Management, 45(4):427–437, 2009, ISSN 0306-4573. <https://www.sciencedirect.com/science/article/pii/S0306457309000259>. 13, 15, 16, 21
- [27] Cunha, Washington, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha e Marcos André Gonçalves: *On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study*. Inf. Process. Manage., 58(3), maio 2021, ISSN 0306-4573. <https://doi.org/10.1016/j.ipm.2020.102481>. 13, 15, 16, 21, 23
- [28] Wirth, Rüdiger e Jochen Hipp: *Crisp-dm: Towards a standard process model for data mining*. Em *Proceedings of the 4th International Conference on the Practical*

- Applications of Knowledge Discovery and Data Mining*, páginas 29–39. Citeseer, 2000. <http://cs.unibo.it/~daniilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf>. 19, 21
- [29] Nadkarni, Prakash M., Lucila Ohno-Machado e Wendy W. Chapman: *Natural language processing: an introduction*. Journal of the American Medical Informatics Association : JAMIA, 18(5):544–551, 2011. <https://doi.org/10.1136/amiajnl-2011-000464>. 21, 29
- [30] Chen, Long, Liangxiao Jiang e Chaoqun Li: *Using modified term frequency to improve term weighting for text classification*. Engineering Applications of Artificial Intelligence, 101:104215, 2021, ISSN 0952-1976. <https://www.sciencedirect.com/science/article/pii/S0952197621000622>. 21
- [31] Uysal, Alper Kursat e Serkan Gunal: *The impact of preprocessing on text classification*. Information Processing Management, 50(1):104–112, 2014, ISSN 0306-4573. <https://www.sciencedirect.com/science/article/pii/S0306457313000964>. 22, 23
- [32] Singh, Jasmeet e Vishal Gupta: *Text stemming: Approaches, applications, and challenges*. ACM Comput. Surv., 49(3), setembro 2016, ISSN 0360-0300. <https://doi.org/10.1145/2975608>. 23
- [33] Oprea, Simona Vasilica, Adela Bâra e Marian Pompiliu Cristescu: *A comprehensive analysis of text similarity metrics and vectorization techniques for content-based product recommendation*. IEEE Access, 14:58670–58689, 2026. 23
- [34] Patil, Rajvardhan, Sorio Boit, Venkat Gudivada e Jagadeesh Nandigam: *A survey of text representation and embedding techniques in nlp*. IEEE Access, 11:36120–36146, 2023. 23
- [35] De Santis, Enrico, Alessio Martino, Francesca Ronci e Antonello Rizzi: *From bag-of-words to transformers: A comparative study for text classification in healthcare discussions in social media*. IEEE Transactions on Emerging Topics in Computational Intelligence, 9(1):1063–1077, 2025. 24, 58
- [36] Viegas, Felipe, Leonardo Rocha e Marcos Gonçalves: *On the role of semantic word clusters — cluwords — in natural language processing (nlp) tasks*. Em *Anais do XXXVII Concurso de Teses e Dissertações*, páginas 38–47, Porto Alegre, RS, Brasil, 2024. SBC. <https://sol.sbc.org.br/index.php/ctd/article/view/29267>. 24, 61
- [37] Altalhan, Manahel, Abdulmohsen Algarni e Monia Turki-Hadj Alouane: *Imbalanced data problem in machine learning: A review*. IEEE Access, 13:13686–13699, 2025. 24, 54, 61
- [38] Zhu, Rui, Yiwen Guo e Jing Hao Xue: *Adjusting the imbalance ratio by the dimensionality of imbalanced data*. Pattern Recognition Letters, 133, março 2020. 25

- [39] Souza, F.C., R.F. Nogueira e R.A. Lotufo: *Bert models for brazilian portuguese: Pre-training, evaluation and tokenization analysis*. Applied Soft Computing, 149:110901, 2023, ISSN 1568-4946. <https://www.sciencedirect.com/science/article/pii/S1568494623009195>. 28, 43, 45, 61
- [40] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye e Tie Yan Liu: *Lightgbm: a highly efficient gradient boosting decision tree*. Em *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, página 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc., ISBN 9781510860964. 28
- [41] Opitz, Juri: *A closer look at classification evaluation metrics and a critical reflection of common evaluation practice*. Transactions of the Association for Computational Linguistics, 12:820–836, junho 2024, ISSN 2307-387X. https://doi.org/10.1162/tac1_a_00675. 30, 49
- [42] Silva Barbon, Rafael e Ademar Takeo Akabane: *Towards transfer learning techniques—bert, distilbert, bertimbau, and distilbertimbau for automatic text classification from different languages: A case study*. Sensors, 22(21), 2022, ISSN 1424-8220. <https://www.mdpi.com/1424-8220/22/21/8184>. 43
- [43] Garcia K, Shiguihara P, Berton L: *Breaking news: Unveiling a new dataset for portuguese news classification and comparative analysis of approaches*. PloS one, 19(1), 2024. 58