INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CAMPUS CAÇADOR
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

RICARDO AUGUSTO FRANCO

DESENVOLVIMENTO DE *GUARDRAILS* PARA CONTROLE E MARCAÇÃO DE MENSAGENS NA INTERAÇÃO DE USUÁRIOS EM SISTEMAS BASEADOS EM GRANDES MODELOS DE LINGUAGEM

Caçador, SC
2025

RICARDO AUGUSTO FRANCO

DESENVOLVIMENTO DE *GUARDRAILS* PARA CONTROLE E MARCAÇÃO DE MENSAGENS NA INTERAÇÃO DE USUÁRIOS EM SISTEMAS BASEADOS EM GRANDES MODELOS DE LINGUAGEM

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador:
Prof. Eli Lopes da Silva, Dr.

Coorientador:
Prof. Cristiano Mesquita Garcia, Me.

Caçador, SC

2025

Franco, Ricardo Augusto.
F825d Desenvolvimento de guardrails para controle e marcação de mensagens na interação de usuários em sistemas baseados em grandes modelos de linguagem / Ricardo Augusto Franco ; orientador: Eli Lopes da Silva, coorientador: Cristiano Mesquita Garcia. -- 2025.
66 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Caçador, 2025.
Inclui bibliografias.

1. Guardrails. 2. Aprendizado de máquina. 3. Grandes modelos de linguagem. 4. Inteligência artificial. I. Silva, Eli Lopes da. II. Garcia, Cristiano Mesquita. III. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina – Graduação em Sistemas de Informação. IV. Título.

CDD 600

Ficha catalográfica elaborada pela Bibliotecária
Janice Moser Corrêa – CRB-14/1865

RICARDO AUGUSTO FRANCO

DESENVOLVIMENTO DE *GUARDRAILS* PARA CONTROLE E MARCAÇÃO DE MENSAGENS NA INTERAÇÃO DE USUÁRIOS EM SISTEMAS BASEADOS EM GRANDES MODELOS DE LINGUAGEM

Este trabalho, requisito parcial para obtenção do título de Bacharel em Sistemas de Informação do Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina, foi avaliado e aprovado pela banca examinadora.

Caçador, SC, 10 de julho de 2025.

gov.br Documento assinado digitalmente
ELI LOPES DA SILVA
Data: 12/08/2025 10:31:32-0300
Verifique em https://validar.iti.gov.br

Eli Lopes da Silva, Dr.
Orientador
Instituto Federal de Santa Catarina (IFSC)

gov.br Documento assinado digitalmente
CRISTIANO MESQUITA GARCIA
Data: 12/08/2025 10:41:07-0300
Verifique em https://validar.iti.gov.br

Cristiano Mesquita Garcia, Me.
Coorientador
Instituto Federal de Santa Catarina (IFSC)

PAULO ROBERTO CORDOVA: 00830004963
Digitally signed by PAULO ROBERTO CORDOVA: 00830004963
DN: CN=PAULO ROBERTO CORDOVA: 00830004963, OU=IFSC - Instituto Federal de Santa Catarina, O=ICPEdu, C=BR
Reason: I am approving this document
Location: your signing location here
Date: 2025.08.12 13:52:41-03'00'
Foxit PhantomPDF Version: 10.1.1

Paulo Roberto Córdova, Dr
Avaliador
Instituto Federal de Santa Catarina (IFSC)

gov.br Documento assinado digitalmente
JEAN PAUL BARDDAL
Data: 12/08/2025 10:52:03-0300
Verifique em https://validar.iti.gov.br

Jean Paul Barddal, Dr
Avaliador
Pontifícia Universidade Católica do Paraná (PUCPR)

AGRADECIMENTOS

Como admirador da obra de Gabriel García Márquez, sempre considerei Cem Anos de Solidão o melhor livro já escrito, com frases como: “A solidão é minha companheira mais fiel. Ela nunca me abandonou, mesmo nos momentos mais difíceis”. Mas, aprendi que as conexões se valorizam pela solidão.

Agradeço ao meu pai, Agostinho, por ter sido a primeira pessoa a me apresentar ao amor pela leitura e tudo o que isso trouxe à minha vida. Seu incentivo e a frase “não importa como a gente esteja, sempre dou um jeito” para comprar livros, despertaram minha paixão pelo conhecimento.

Ao pensar sobre amizades, por mais que meus laços sejam poucos, contei com amizades especiais: Rafael, Vitor, Gabriel, Lucas e Cris, que, com muita paciência, me escutaram, trouxeram conforto e apoiaram nos momentos difíceis.

Aos meus orientadores Eli e Cristiano e ao coordenador Eduardo, agradeço por serem exemplos admiráveis, pela paciência diante das minhas figurinhas de gatos, crises eventuais durante a produção deste trabalho, além dos comentários impróprios sobre a cidade de Caçador e seus elementos.

Também agradeço à minha gata Hortelã, que trouxe frescor aos meus dias bons e, principalmente, aos ruins. Vinda de uma cidade com poeira das metalúrgicas, ela trouxe leveza e calma à minha rotina, ouvindo minhas reclamações como uma brisa suave.

Agradeço aos integrantes da ONG Atitude Infinita, em especial Thaysa, Yan e Felipe, por me acolherem e guiarem nos momentos mais desafiadores.

Sou profundamente grato aos meus modelos profissionais, Thiago e João. Obrigado por abrirem caminhos que agora consigo trilhar, mesmo isso, na prática, significando que me tratoraram do Twitter para a vida profissional.

Aos colegas da Monest, obrigado por transformarem dias intensos em momentos divertidos, por aturarem minhas piadas de gosto duvidoso, perguntas sem sentido e minhas exaltações repentinas, sendo minhas principais companhias no desenvolvimento deste trabalho.

Por fim, lembrando o que Gabo escreveu, “a solidão é muito bonita quando se tem alguém para contá-la”, agradeço sinceramente a todos que dividiram, dividem ou dividirão algum laço comigo. Que possamos sempre aproveitar a vida intensamente e viver até morrer ou morrer de tanto viver.

“AI is not just a technology, it's a responsibility”

(Fei-Fei Li, 2024)

RESUMO

As soluções baseadas em inteligência artificial estão cada vez mais presentes, impactando diversos setores e gerando avanços significativos em várias áreas. A adoção crescente desses modelos requer a implementação de mecanismos que garantam a qualidade e a segurança das interações, especialmente em contextos sensíveis como o ambiente de cobrança, onde é essencial distinguir mensagens que contribuem para o avanço da negociação daquelas que podem comprometer o processo. Este trabalho tem como objetivo o desenvolvimento de *guardrails* para marcação de mensagens enviadas por usuários durante interações em sistemas de cobrança baseados em grandes modelos de linguagem (LLMs). Para isso, propõe-se o treinamento e a avaliação de diferentes modelos de aprendizado de máquina, utilizando técnicas de processamento de linguagem natural para analisar as mensagens. Os modelos são utilizados para classificar as interações como funcionais, quando favorecem o progresso da recuperação de crédito, ou disfuncionais, quando não contribuem para o avanço, podendo incluir respostas vagas, negativas infundadas, uso de palavrões, ironias ou tentativas de fraude. A implementação desses *guardrails* visa permitir respostas mais ágeis pelas equipes de cobrança e apoiar a construção de conjuntos de dados mais qualificados para o aprimoramento contínuo dos sistemas.

Palavras-Chave: *guardrails*; aprendizado de máquina; grandes modelos de linguagem; inteligência artificial.

ABSTRACT

Artificial intelligence-based solutions are increasingly present across various sectors, driving significant advances in multiple areas. The growing adoption of such models demands the implementation of mechanisms that ensure the quality and security of interactions, especially in sensitive contexts such as debt collection, where it is essential to distinguish between messages that contribute to negotiation progress and those that may hinder the process. This study aims to develop guardrails for the control and labeling of messages sent by users during interactions in debt collection systems powered by large language models (LLMs). To this end, the training and evaluation of different machine learning models are proposed, using natural language processing techniques to analyze user messages. The models are employed to classify interactions as functional, when they support the progress of credit recovery, or dysfunctional, when they do not, including vague responses, unfounded refusals, profanity, sarcasm, or attempted fraud. The implementation of these guardrails seeks to enable faster responses by collection teams and to support the construction of more qualified datasets for the continuous improvement of such systems.

Keywords: guardrails; machine learning; large language models; artificial intelligence.

LISTA DE ILUSTRAÇÕES

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

BERT	*Bidirectional Encoder Representations from Transformers*
BoW	*Bag-of-Words*
CRISP-DM	*Cross-Industry Standard Process for Data Mining*
DSR	*Design Science Research*
IA	Inteligência Artificial
LLMs	*Large Language Models*
Linear SVC	Máquina de Vetor de Suporte Linear
MSE	Erro Quadrático Médio
NLI	Inferência Natural de Linguagem
NLP	*Natural Language Processing*
NLTK	*Natural Language Toolkit*
RNNs	*Recurrent Neural Networks*
ROC AUC	*Receiver Operating Characteristic – Area Under the Curve*
SMO	*Sequential Minimal Optimization*
SVM	*Support Vector Machines*
TF-IDF	*Term Frequency–Inverse Document Frequency*
VADER	*Valence Aware Dictionary and sEntiment Reasoner*
XNLI	*Cross-lingual Natural Language Inference*

SUMÁRIO

1 INTRODUÇÃO

Atualmente, a comunicação por mensagens instantâneas e de alta disponibilidade atua em diversas esferas do nosso cotidiano. Ferramentas como e-mail, SMS, WhatsApp e Telegram têm se tornado essenciais para o fluxo rápido de informações, tanto no ambiente corporativo quanto no pessoal. Esses recursos permitem uma troca de informações instantânea e eficiente, possibilitando maior conectividade e produtividade. Com a popularização e a confiabilidade dessas ferramentas, elas se consolidaram como métodos de comunicação oficiais em diversos setores. Segundo uma pesquisa conduzida pelo Boston Consulting Group em parceria com a Meta, 85% dos brasileiros preferem interagir com empresas por meio de mensagens (Frenay *et al*., 2024).

Além disso, empresas do setor financeiro vêm adotando soluções digitais para estabelecer uma comunicação mais eficaz e direta com seus clientes, com envio de lembretes sobre vencimentos, propostas de renegociação e notificações de ofertas de maneira ágil e econômica. Como objeto de estudo, este trabalho aborda a empresa Monest Cobranças, uma empresa especializada em soluções para recuperação de crédito. Fundada em 2018 e pioneira na utilização de *Large Language Models* (LLMs) para a automação de cobranças no idioma português, a empresa conta atualmente com 53 colaboradores (dados de dezembro de 2024). Em 2024, registrou um crescimento de 130,77% em relação ao período anterior à adoção dos LLMs. Esse desempenho ressalta sua capacidade de inovar, integrando tecnologias avançadas que aprimoram a eficiência operacional e o relacionamento com seus clientes e respectivos devedores.

A implementação de sistemas de Inteligência Artificial (IA) em ambientes corporativos envolve custos significativos que vão além do investimento inicial em tecnologia. Esses custos incluem despesas contínuas com infraestrutura de processamento e armazenamento de dados, consumo de energia, atualizações de *software* e manutenção de *hardware*, além da necessidade de profissionais especializados para desenvolver, treinar e aperfeiçoar os modelos de IA. Portanto, é essencial que as empresas gerenciem esses custos de maneira eficiente para garantir a sustentabilidade financeira e operacional de suas iniciativas tecnológicas.

No contexto de empresas que utilizam modelos de linguagem para interações com devedores, como a empresa estudada, interações maliciosas podem

representar um desafio significativo. Essas interações, que incluem tentativas de manipulação do sistema, uso indevido dos recursos ou disseminação de conteúdos inadequados, podem sobrecarregar os sistemas de IA, resultando em consumo excessivo de recursos computacionais e acarretando o aumento dos custos operacionais. Monitorar e identificar essas interações maliciosas é fundamental para evitar comprometer eficiência do sistema. Ao implementar mecanismos de identificação e classificação a comportamentos suspeitos, a empresa pode reduzir o desperdício de recursos, melhorar a segurança dos sistemas e proporcionar uma experiência mais segura e satisfatória aos devedores legítimos.

Além das ferramentas convencionais de comunicação, como e-mails e SMS, soluções baseadas em LLMs, como o ChatGPT(OpenAI, 2022), estão emergindo como um novo paradigma de interação. Empresas como a estudada nesta pesquisa têm explorado essas novas tecnologias para potencializar a comunicação e o relacionamento com seus clientes. Esses modelos oferecem uma abordagem personalizada e natural que complementa os canais tradicionais, trazendo mais flexibilidade e eficiência ao processo de comunicação.

Neste contexto, destaca-se a ampla adesão do WhatsApp nos dispositivos móveis, plataforma presente em 99% dos celulares brasileiros, de acordo com Guanaes (2024). Esse dado reforça a relevância de ferramentas digitais no cotidiano da sociedade e evidencia o potencial de integração entre tecnologias emergentes, como os modelos de linguagem, e plataformas de comunicação amplamente adotadas. Assim, a convergência entre inovação tecnológica e acessibilidade digital permite às empresas ampliarem as suas estratégias de relacionamento, promovendo experiências mais enriquecedoras e eficientes para os seus usuários.

Além de sua capacidade de compreender e gerar textos, as LLMs possibilitam um atendimento personalizado e ampliado, ajustado às demandas específicas de cada usuário. Essa abordagem promove interações eficazes e objetivas, mantendo o aspecto natural da comunicação, essencial para estabelecer confiança e engajamento. A versatilidade dessas ferramentas não se limita a um único setor; sua capacidade de adaptação tem despertado grande interesse em áreas como o setor de cobrança, que busca estratégias inovadoras para otimizar a relação com os devedores e alcançar melhores resultados na recuperação de crédito.

Os modelos de aprendizado de máquina desempenham um papel fundamental no contexto de interações entre devedores e IA. Em especial, técnicas

de aprendizado não supervisionado, como *autoencoders* e outros modelos, são capazes de aprender representações eficientes dos dados, capturando padrões importantes e detectando anomalias sem a necessidade de rótulos. Conforme discutido por Goodfellow, Bengio e Courville (2016), os *autoencoders* são capazes de modelar a estrutura intrínseca dos dados, permitindo a extração de características relevantes que podem ser utilizadas para monitorar e melhorar sistemas baseados em LLMs. Além disso, modelos supervisionados, como as *Support Vector Machines* (SVM) em português, máquina de vetores de suporte, apesar de requererem dados rotulados, também contribuem significativamente ao classificar e prever comportamentos com alta precisão, desempenhando um papel crucial no aprimoramento de sistemas inteligentes.

Ao implementar modelos de aprendizado de máquina no formato de *guardrails*, é possível identificar comportamentos inesperados ou não conformes em um sistema de IA, garantindo que as interações com os clientes atendam aos padrões éticos e de qualidade estabelecidos. Conforme demonstrado por Xu *et al*. (2017), o uso de *autoencoders* permite a detecção eficaz de anomalias em textos curtos, identificando desvios significativos dos padrões aprendidos pelo modelo. Essa abordagem contribui para a transparência das operações e para a efetividade da comunicação entre humanos e máquinas, alinhando-se aos objetivos de promover interações responsáveis e eficientes na empresa em questão.

Investir em soluções que permitam o monitoramento e a classificação de interações que não acrescentam valor ao negócio é, portanto, uma estratégia que traz benefícios econômicos e operacionais. Essa abordagem garante que os sistemas de IA operem de forma eficiente e segura, maximiza o retorno sobre o investimento em tecnologia e contribui para a sustentabilidade e o sucesso da empresa no longo prazo. A categorização proativa de interações que não agregam ao negócio não apenas reduz custos operacionais associados ao consumo excessivo de recursos computacionais, mas também previne potenciais danos, como ataques cibernéticos ou exploração de vulnerabilidades que poderiam levar a perdas financeiras ou de reputação.

Diante desse cenário, o presente trabalho visa explorar a importância do monitoramento de interações indesejadas em sistemas de IA e como essa prática pode resultar em economias significativas para a empresa. Por meio da implementação de modelos de aprendizado de máquina, busca-se identificar

interações que não geram valor ao negócio, otimizando o uso de recursos e promovendo um ambiente seguro e eficiente para as operações empresariais. Essa integração entre tecnologia e processos de negócio reforça a importância de soluções digitais como um fator-chave para impulsionar a eficiência organizacional e o sucesso em interações de alto impacto para o negócio.

1.1 Justificativa

A presente pesquisa fundamenta-se na crescente relevância da comunicação digital no setor de cobrança, onde ferramentas como e-mail, SMS e mensageiros instantâneos transformaram significativamente as interações entre empresas e clientes. A adoção de *chatbots* baseados em LLMs, como o ChatGPT, apresenta uma oportunidade única para aprimorar o atendimento ao cliente. No entanto, implementar essas tecnologias traz desafios críticos relacionados à qualidade, ética das interações e custos operacionais dos sistemas de IA. Interações maliciosas podem sobrecarregar os sistemas, aumentando o consumo de recursos computacionais e os custos para a empresa. Portanto, é essencial desenvolver mecanismos que garantam a conformidade e a eficácia das comunicações, prevenindo mal-entendidos e assegurando uma experiência positiva para o cliente e segura para a organização (Limna *et al.*, 2023), além de otimizar os recursos financeiros envolvidos.

Diante desse cenário, o desenvolvimento de processos de garantia de qualidade e controle de interações utilizando aprendizado de máquina surge como uma demanda estratégica. Esses métodos não apenas buscam otimizar o desempenho dos sistemas de inteligência artificial, mas também desempenham um papel fundamental na detecção e marcação de interações disfuncionais. Ao identificar rapidamente essas interações, é possível reduzir o consumo desnecessário de recursos computacionais, promovendo economias significativas para a empresa. O objetivo deste trabalho é desenvolver e validar modelos de aprendizado de máquina para analisar mensagens de devedores e identificar, de forma eficaz, possíveis interações disfuncionais, quando não contribuem para o avanço, podendo incluir respostas vagas, uso de palavrões, ironias ou tentativas de fraude. Assim, pretende-se aprimorar as operações de cobrança, garantir maior eficiência nos processos, diminuir custos, proteger a integridade das interações e

fortalecer a confiança dos clientes na empresa.

1.2 Definição do Problema

Com a crescente relevância da comunicação digital no setor de cobrança, onde agilidade e eficiência são determinantes para o sucesso das operações, a disseminação de ferramentas digitais e o uso de sistemas inteligentes, em especial as LLMs, têm transformado significativamente como as empresas interagem com seus devedores. Nesse contexto, a Monest Cobranças, que já atendeu mais de 1,5 milhão de conversas desde a adoção dos LLMs, apresenta um cenário propício para investigar soluções capazes de detectar interações improdutivas ou maliciosas (como tentativas de fraude, *spam* ou abusos). O objetivo é prevenir o consumo excessivo de recursos computacionais e minimizar os impactos financeiros decorrentes dessas interações. Assim, a questão central que orienta esta pesquisa é: Como modelos de aprendizado de máquina podem contribuir para a identificação e prevenção de interações improdutivas ou maliciosas nas conversas entre devedores e LLMs?

1.3 Objetivos

Nesta sessão será apresentado os objetivos que guiarão o desenvolvimento desta pesquisa.

1.3.1 Objetivo Geral

O objetivo geral desta pesquisa é desenvolver e validar modelos de aprendizado de máquina para a análise de mensagens enviadas por devedores durante interações de cobrança com sistemas de inteligência artificial via WhatsApp, com o intuito de identificar interações improdutivas. A proposta visa classificar automaticamente as mensagens como funcionais, quando contribuem para o avanço da negociação, ou disfuncionais, quando dificultam o processo de recuperação de crédito, incluindo comunicações vagas, ofensivas ou que demonstrem má-fé.

1.3.2 Objetivos Específicos

1. Implementar modelos de aprendizado de máquina para identificar, de forma eficaz, interações que não agregam valor ao processo de cobrança.
2. Definir e aplicar métricas de desempenho para avaliar a qualidade dos modelos desenvolvidos.
3. Validar a eficácia dos modelos na detecção de mensagens disfuncionais, considerando seu impacto na marcação e classificação de conversas em sistemas de cobrança digital.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os fundamentos teóricos que sustentam a implementação de modelos de aprendizado de máquina desenvolvidos nesta pesquisa para a análise de mensagens de devedores. Busca-se, por meio deste, esclarecer os principais conceitos e termos técnicos que embasam o projeto, fornecendo uma base sólida para a compreensão das escolhas realizadas ao longo de sua implementação. Ademais, este capítulo tem a finalidade de situar o desenvolvimento do trabalho no contexto dos avanços recentes nas áreas de inteligência artificial, modelos de linguagem e processamento de texto.

2.1 Cross-Industry Standard Process for Data Mining

O *Cross-Industry Standard Process for Data Mining* (CRISP-DM) é um modelo de processo amplamente aceito que oferece uma abordagem estruturada para projetos de mineração de dados. Desenvolvido em 1996 por um consórcio de empresas, incluindo SPSS, NCR Corporation e Daimler-Benz, tornou-se o padrão na indústria de análise de dados (Chapman *et al*., 2000) É composto por seis etapas:

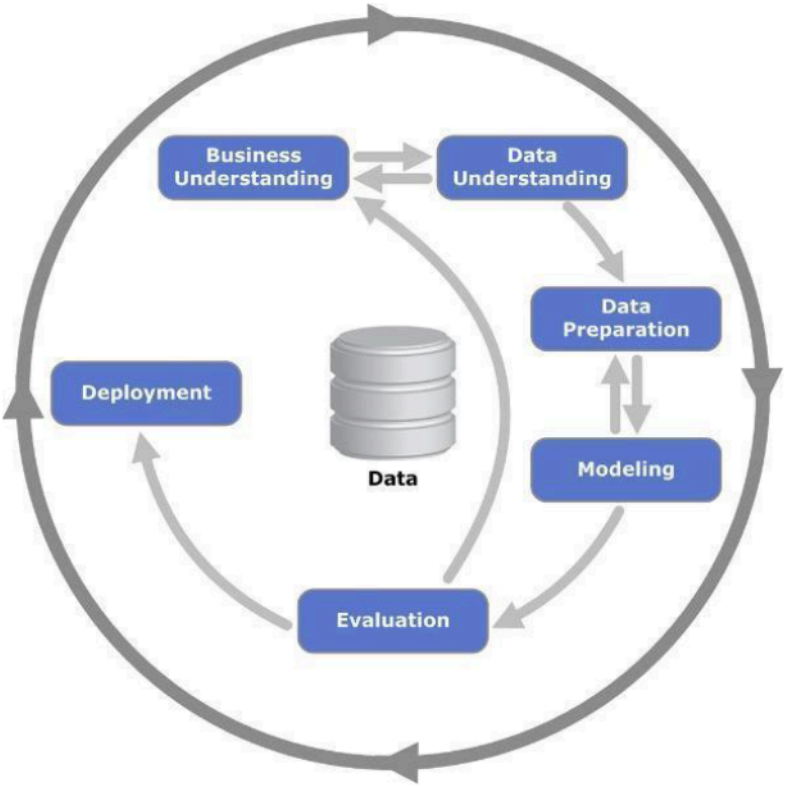
1. **Compreensão do Negócio:** Essa etapa é dedicada a entender os objetivos e requisitos do projeto a partir de uma perspectiva de negócios. Envolve a definição clara do problema e a formulação de perguntas de mineração de dados que possam abordar efetivamente os desafios identificados.
2. **Entendimento dos Dados:** Consiste na coleta inicial e análise exploratória dos dados para compreender suas características, identificar problemas de qualidade e extrair *insights* preliminares que guiarão as próximas etapas.
3. **Preparação dos dados:** inclui todas as atividades necessárias para construir o conjunto de dados final a partir dos dados brutos. As tarefas envolvem limpeza, transformação, integração e criação de variáveis relevantes, garantindo que os dados estejam prontos para a modelagem.
4. **Modelagem:** Etapa em que técnicas específicas de modelagem são selecionadas, aplicadas e avaliadas. Inclui a escolha dos algoritmos, a definição de parâmetros e critérios de teste, bem como a construção dos modelos preditivos.
5. **Avaliação:** Após a construção dos modelos, é realizada uma análise para

verificar se eles atendem aos objetivos do projeto e resolvem os problemas de negócios. Caso necessário, ajustes podem ser realizados ou até mesmo o modelo pode ser reformulado.

6. **Implementação:** Na última etapa, o modelo aprovado é implementado no ambiente operacional. Isso pode incluir sua integração em sistemas de tomada de decisão, apresentação dos resultados para os *stakeholders* ou documentação detalhada das lições aprendidas durante o projeto.

A Figura 1 representa as fases do CRISP-DM.

Figura 1 — Fases do CRISP-DM.



Nota: em ordem - Compreensão do Negócio (*Business Understanding*), Entendimento dos dados (*Data Understanding*), Preparação dos Dados (*Data Preparation*), Modelagem (*Modeling*), Avaliação (*Evaluation*), Implementação (*Deployment*)
Fonte: Jensen (2016).

2.2 Processamento de Linguagem Natural

A área da *Natural Language Processing* (NLP) ou Processamento de Linguagem Natural, investiga a interação entre computadores e a linguagem humana, visando capacitar as máquinas a compreenderem, interpretarem e gerarem texto ou fala de forma que seja natural e útil para os usuários (Goodfellow; Bengio; Courville, 2016). A NLP abrange uma variedade de tarefas, como tradução

automática, análise de sentimentos, resposta automática a perguntas e geração de conteúdo textual.

A introdução de arquiteturas baseadas em *deep learning*, uma subárea do aprendizado de máquina que se baseia em redes neurais artificiais com múltiplas camadas para aprender representações complexas dos dados, especialmente os modelos *Transformer*, trouxe uma revolução para o campo do NLP. Propostos por Vaswani *et al*. (2023), os *Transformers* são modelos de aprendizado de máquina que utilizam exclusivamente o mecanismo de atenção para processar sequências de dados. Diferentemente dos modelos anteriores, como as *Recurrent Neural Networks* (RNNs), eles permitem o processamento paralelo dos dados e são mais eficientes ao capturar dependências de longo prazo. Esse avanço resultou em melhorias substanciais em diversas tarefas de NLP, estabelecendo novos marcos na capacidade das máquinas de compreender e gerar linguagem natural.

O pré-processamento de texto desempenha um papel fundamental na modelagem de linguagem e na análise de dados textuais, sendo uma etapa essencial da fase de preparação dos dados no contexto da metodologia CRISP-DM. Ele visa transformar dados linguísticos brutos em um formato estruturado e limpo, adequado para análise ou processamento posterior por algoritmos de NLP. Essa etapa é crucial para reduzir a complexidade do texto, eliminar ruídos e realçar padrões que podem ser explorados por modelos computacionais. Esta etapa pode ser composta de normalização de texto, remoção de ruídos, substituição de emojis, substituição de entidades, remoção de *stopwords* e criação de *embeddings*.

A normalização de texto consiste em padronizar a forma como os dados textuais aparecem, reduzindo variações que não agregam significado. Isso inclui converter todos os caracteres para caixa baixa, remover acentos, pontuações desnecessárias e espaços em branco duplicados. Também é comum expandir abreviações e aplicar técnicas de lematização ou *stemming*, visando a redução das palavras às suas formas canônicas (Jurafsky; Martin, 2008). Esse processo é essencial para diminuir a variabilidade linguística e melhorar o desempenho dos algoritmos de Processamento de Linguagem Natural.

A remoção de ruídos consiste na eliminação de elementos irrelevantes ou redundantes do texto, como caracteres especiais, URLs, códigos HTML e outros símbolos não linguísticos. Esses elementos podem atrapalhar o aprendizado dos modelos ao introduzir padrões espúrios. De acordo com Uysal e Gunal (2014), a

limpeza adequada desses ruídos melhora a qualidade dos dados e pode aumentar significativamente a acurácia em tarefas de classificação textual.

Emojis e símbolos são comumente utilizados em comunicações informais, especialmente em plataformas digitais e redes sociais. No entanto, sua presença pode representar um desafio para os sistemas de NLP, que geralmente são treinados para processar texto escrito convencionalmente. A substituição de *emojis* ou símbolos por descrições textuais consiste em traduzir esses elementos gráficos em palavras que capturem seu significado ou intenção emocional. Por exemplo, um emoji de sorriso pode ser substituído por “sorriso” ou “felicidade”, enquanto um símbolo de coração pode ser transformado em “amor”' ou “apreço”. Esse processo facilita a integração de informações emocionais e contexto semântico ao texto, melhorando a compreensão por parte dos algoritmos de NLP. Além disso, segundo Palomino e Aider (2022), essa substituição auxilia na manutenção da coesão textual e na precisão das análises realizadas, como a detecção de sentimentos e a categorização de conteúdo.

A substituição de entidades envolve a substituição de nomes próprios, locais, organizações ou outras entidades específicas por marcadores genéricos, ou identificadores padronizados. Esse processo é fundamental para preservar a privacidade e a confidencialidade dos dados, além de reduzir a variabilidade das entidades nos textos, facilitando a generalização dos modelos de NLP. Por exemplo, nomes próprios podem ser substituídos por “<PESSOA>” e organizações por “<ORGANIZACAO>”. Essa abordagem, como demonstram Camacho-Collados e Pilehvar (2018), reduz a dimensionalidade e mitiga vieses, promovendo modelos mais generalizáveis e justos.

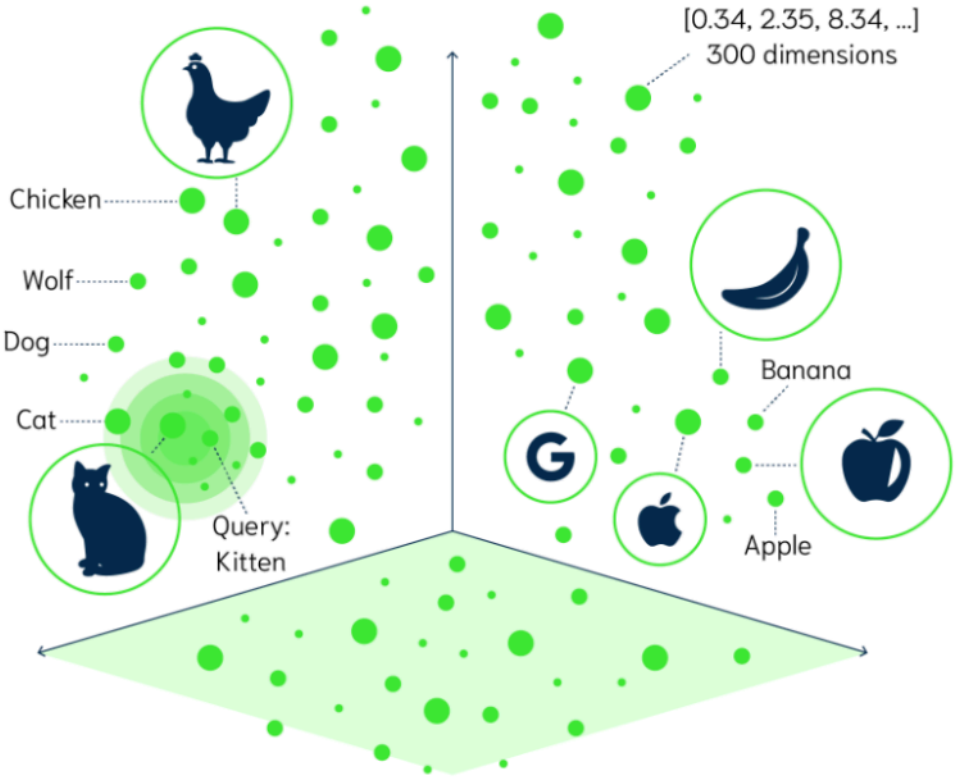
Stopwords são palavras de alta frequência que geralmente não carregam significado semântico relevante para determinadas análises, como artigos, preposições, conjunções e pronomes (por exemplo, “e”, “a”, “o”, “de”, “em”). A identificação e remoção dessas palavras são etapas fundamentais no pré-processamento de textos em NLP, pois contribuem para a redução da dimensionalidade dos dados e melhoram a eficiência dos algoritmos de processamento. Ao eliminar *stopwords*, concentra-se a análise textual nas palavras que realmente contribuem para a compreensão do conteúdo e dos temas abordados no texto. A remoção dessas palavras contribui para a redução da dimensionalidade e do ruído textual. Segundo Baeza-Yates e Ribeiro-Neto (1999), essa prática é

amplamente utilizada para aumentar a eficiência de algoritmos de classificação e recuperação de informação.

E*mbeddings* são representações vetoriais de palavras, frases, parágrafos ou até mesmo documentos inteiros. Essas representações visam capturar aspectos como significado, semântica e sintaxe dos elementos linguísticos, além das relações e contextos em que aparecem.

De maneira geral, *embeddings* são estruturados de forma que elementos linguísticos com significados ou contextos semelhantes fiquem próximos uns dos outros em um espaço vetorial, ou seja, apresentem menor distância entre seus vetores. Por outro lado, *embeddings* com significados distintos tendem a estar mais afastados. Por exemplo, no espaço vetorial, o *embedding* da palavra *kitten* (gatinho, em inglês) estará mais próximo de *cat* (gato, em inglês) do que de *dog* (cachorro, em inglês), conforme ilustrado na Figura 2.

Figura 2 — Representação Vetorial



Fonte: Ham (2022).

Essas representações vetoriais são especialmente úteis em tarefas como a recuperação de informações, pois permitem identificar relações semânticas mesmo

quando não há correspondência exata de palavras-chave. Por exemplo, uma busca por “carro elétrico compacto” pode trazer resultados relacionados a “veículo pequeno movido a eletricidade”, destacando a capacidade dos *embeddings* de compreender o significado contextual por trás das palavras.

2.2.1 Métodos de Vetorização de Texto

Os métodos de vetorização textual são fundamentais no NLP, sendo responsáveis por converter textos em representações numéricas que possam ser processadas por algoritmos de aprendizado de máquina. Essa transformação é crucial, pois permite a análise quantitativa e qualitativa dos conteúdos linguísticos. De forma geral, essas técnicas podem ser classificadas em abordagens tradicionais baseadas na frequência e presença de palavras e métodos modernos que levam em consideração o contexto linguístico

Nas abordagens baseadas diretamente em palavras, os documentos são transformados em vetores que têm dimensões equivalentes ao tamanho do vocabulário disponível. Um exemplo clássico desse grupo é o modelo Bag-of-Words (BoW) é um dos métodos mais tradicionais de vetorização textual, no qual um texto é representado por um vetor de contagem de palavras. Cada posição do vetor corresponde a uma palavra do vocabulário e armazena o número de vezes que essa palavra aparece no documento. Por exemplo, dadas todas as palavras únicas de um corpus, um documento é vetorizado como um histograma das frequências de cada termo.

Esse modelo é simples e tem baixo custo de implementação, tendo sido amplamente utilizado em tarefas de classificação de textos e recuperação de informações. Por outro lado, ao tratar o documento apenas como um “saco” de palavras, o BoW desconsidera completamente a ordem das palavras e as dependências entre elas, ignorando o contexto em que surgem. Assim, informações semânticas importantes são perdidas, o método não distingue frases com ordem invertida nem reconhece que termos diferentes possam ter significado similar. Consequentemente, o BoW mostra-se incapaz de capturar nuances mais profundas de significado no texto (Jurafsky; Martin, 2008). Além disso, a dimensionalidade do vetor resultante pode ser muito alta (uma dimensão por palavra distinta) e a maioria das posições tende a ser zero para um dado documento, resultando em vetores

esparsos.

A técnica *Term Frequency–Inverse Document Frequency* (TF-IDF) é uma extensão do BoW que procura atribuir maior peso às palavras mais informativas de cada documento. Em vez de apenas contar ocorrências, o TF-IDF multiplica a frequência de cada termo no documento (TF) por um fator inversamente proporcional ao número de documentos em que o termo aparece (IDF). Dessa forma, termos muito frequentes em todo o corpus (por exemplo, “de”, “e”, “o”) recebem pesos baixos, enquanto termos raros e potencialmente mais relevantes têm peso alto. Essa ponderação realça os conteúdos distintivos de cada documento e atenua a influência de palavras comuns. O TF-IDF tem ampla aplicação em sistemas de busca e mineração de textos, melhorando a identificação de documentos relevantes por destacar termos-chave de cada item. Contudo, trata-se ainda de um modelo baseado apenas na contagem de palavras individuais, sem compreensão do significado contextual. Assim, tal como o BoW, o TF-IDF falha em capturar as relações semânticas ou o contexto em que as palavras ocorrem (Manning; Schütze, 1999).

As abordagens baseadas em modelos de linguagem representam um avanço significativo na forma como textos são processados computacionalmente. Esses modelos utilizam redes neurais profundas, geralmente treinadas sobre grandes volumes de dados textuais, para gerar representações vetoriais contextualizadas do conteúdo linguístico. Diferentemente dos métodos tradicionais baseados exclusivamente na contagem de palavras, como o modelo BoW ou o TF-IDF, os modelos de linguagem são capazes de capturar relações de dependência entre palavras, mesmo quando distantes entre si, e ajustar a representação de cada termo conforme o seu contexto específico. Essa capacidade resulta em representações mais precisas, que lidam melhor com ambiguidades semânticas e expressões polissêmicas (Manning; Raghavan; Schütze, 2008).

Os modelos mais recentes, chamados language models, deixaram de tratar palavras de forma isolada e passaram a considerar o encadeamento completo da linguagem natural. Essa evolução se deu de forma progressiva: inicialmente com *embeddings* como *word2vec* (Mikolov *et al*., 2013), que atribuíam vetores fixos a palavras com base em ocorrência, até arquiteturas mais sofisticadas que empregam mecanismos de atenção para analisar a estrutura completa das sentenças. Nesse novo contexto, destaca-se o modelo *Bidirectional Encoder Representations from*

Transformers (BERT).

O BERT, desenvolvido por Devlin *et al*. (2019), representa um marco na geração de representações semânticas profundas. Sua principal inovação reside na capacidade de capturar, de forma bidirecional, o contexto completo em que uma palavra está inserida, permitindo compreender o significado dinâmico de termos de acordo com seu uso na frase. Isso é viabilizado pela arquitetura de Transformers, proposta por Vaswani *et al*. (2017), que se fundamenta em mecanismos de *self-attention* para modelar interações entre todos os tokens de uma sequência simultaneamente.

A vetorização de mensagens por meio do BERT se dá de maneira estruturada e precisa. A entrada textual é tokenizada, e um marcador especial [CLS] é adicionado no início da sequência. Ao passar pelas camadas do transformador, cada token recebe uma representação vetorial contextualizada. O vetor associado ao token [CLS] ao final do processamento é frequentemente utilizado como um resumo semântico da sentença, funcionando como *embedding* fixo da mensagem (Devlin *et al*., 2019). Esse vetor é amplamente utilizado em tarefas como classificação de texto, análise de sentimentos e avaliação de similaridade semântica entre frases.

Além da extração do [CLS], também é possível aplicar técnicas de *pooling*, como a média (*average pooling*) ou o máximo (*max pooling*) sobre os *embeddings* de todos os tokens da sequência. Diferentemente das médias de vetores estáticos, como os produzidos por modelos como *word2vec*, esses vetores representam o conteúdo levando em conta o contexto aprendido durante o pré-treinamento, resultando em representações muito mais ricas semanticamente.

Mesmo que o BERT não tenha sido originalmente treinado com o objetivo específico de gerar *embeddings* para comparação direta entre sentenças, sua arquitetura é suficientemente robusta para desempenhar essa função com eficácia. A utilização do vetor [CLS] ou de um vetor gerado por *pooling* permite representar sentenças em um espaço vetorial denso e semântico, útil para sistemas de agrupamento textual, análise de intenção, recuperação de informação e outras aplicações avançadas em processamento de linguagem natural.

Ao considerar o contexto bidirecional, o BERT supera as limitações de modelos anteriores como TF-IDF e BoW, que, apesar de seus méritos históricos, não contemplam a complexidade contextual da linguagem. A geração de *embeddings* densos e altamente informativos torna o BERT uma ferramenta rica para a

vetorização de mensagens.

2.3 Métodos de Aprendizado de Máquina

No domínio da Aprendizagem de Máquina, os métodos costumam ser organizados em dois paradigmas centrais: aprendizado supervisionado e não supervisionado. Enquanto o primeiro se baseia em conjuntos de dados rotulados, orientando a construção de modelos capazes de generalizar a partir de pares entrada-saída previamente conhecidos, o segundo explora a estrutura subjacente de dados não rotulados, identificando padrões, agrupamentos ou representações latentes de forma autônoma.

A distinção entre essas abordagens reside, portanto, na disponibilidade de informações de referência (rótulos) durante o treinamento e na natureza dos problemas que visam resolver. Compreender ambos os paradigmas é essencial, pois eles oferecem estratégias complementares para extrair conhecimento a partir de dados e serão detalhados nas seções subsequentes deste trabalho, evidenciando suas aplicações, vantagens e limitações nos contextos analisados

2.3.1 Não Supervisionado

Dentro do escopo dos métodos não supervisionados, esta subseção concentra-se em técnicas voltadas à detecção de padrões atípicos e à extração de representações latentes, tarefas cruciais quando rótulos não estão disponíveis. Em particular, serão explorados o *One-ClassSVM*, que modela a fronteira de decisão ao redor da distribuição “normal” dos dados para isolar anomalias (Schölkopf *et al*., 2001), e os *autoencoders*, redes neurais capazes de aprender uma codificação comprimida e reconstruir a entrada original, sinalizando desvios significativos por meio do erro de reconstrução (Hinton; Salakhutdinov, 2006).

O *OneClassSVM* é uma técnica que detecta anomalias aprendendo a representar apenas a classe “normal” dos dados. Diferentemente do SVM tradicional, que classifica dados em duas ou mais categorias (Cortes; Vapnik, 1995), o *OneClassSVM* delimita uma fronteira ao redor dos dados de treinamento de uma única classe, rejeitando pontos que estejam fora dessa região. Para isso, os dados são mapeados para um espaço de alta dimensionalidade por meio de um *kernel* (como o RBF), e parâmetros como *nu*, que controla a taxa de *outliers* e *gamma*, que

influencia a definição da fronteira, são ajustados. Assim, mensagens que não se encaixem no padrão aprendido são identificadas como anomalias (Schölkopf *et al*., 2001).

Autoencoders são redes neurais empregadas em aprendizado não supervisionado que buscam aprender representações comprimidas dos dados capazes de reconstruir a entrada original com mínima perda de informação (Hinton; Salakhutdinov, 2006). Ao reduzirem a dimensionalidade, esses modelos conseguem captar estruturas relevantes nos dados sem depender de rótulos, tornando-se úteis em tarefas de extração de características e redução de ruído.

Em termos arquiteturais, um *autoencoder* divide-se em duas partes complementares: o codificador, responsável por transformar o vetor de entrada em um espaço latente de menor dimensão, e o decodificador, encarregado de reverter essa compactação e aproximar-se da entrada original. O treinamento procede pela minimização de uma função de perda, comumente o erro quadrático médio, entre a amostra reconstruída e a observada, otimizando os pesos por retro propagação (Goodfellow; Bengio; Courville, 2016).

Ao impor um gargalo (um número restrito de neurônios na camada latente), força-se o modelo a reter apenas as características estatisticamente mais salientes dos dados, deixando de memorizar detalhes irrelevantes. Uma consequência direta é que, quando treinado exclusivamente em exemplos “normais”, o *autoencoder* tende a gerar erros de reconstrução significativamente maiores para padrões que se desviam do comportamento aprendido, característica explorada em detecção de anomalias (Sakurada; Yairi, 2014).

Entre os principais hiper parâmetros destacam-se: dimensão do espaço latente, profundidade e largura da rede, funções de ativação, taxa de aprendizado e número de épocas.

2.3.2 Supervisionado

No campo do aprendizado supervisionado, serão examinados quatro algoritmos de classificação que se complementam em termos de interpretabilidade, robustez e desempenho preditivo. As Árvores de Decisão estruturam o processo de inferência em uma hierarquia de regras simples, oferecendo transparência na identificação de relações entre atributos e classes (Breiman *et al*., 1984). A Máquina

de Vetor de Suporte Linear (Linear SVC) projeta os dados em um espaço de alta dimensionalidade para encontrar o hiperplano que maximiza a margem entre categorias, sendo particularmente eficaz em problemas de alta dimensão e classes bem separáveis (Cortes; Vapnik, 1995). A regressão logística modela diretamente a probabilidade de pertencimento a uma classe por meio da função sigmoide, destacando-se pela simplicidade matemática e pela facilidade de interpretação dos coeficientes (Hosmer; Lemeshow, 2000). Por fim, a floresta aleatória agrega múltiplas árvores de decisão construídas a partir de amostras e subconjuntos aleatórios de atributos, resultando em uma combinação de modelos que reduz variância e melhora a generalização (Breiman, 2001). Esses modelos ilustram estratégias distintas de aprendizado supervisionado que serão detalhadas nas seções subsequentes.

As árvores de decisão são algoritmos de aprendizado supervisionado amplamente utilizados para tarefas de classificação e regressão, conhecidas pela sua simplicidade e interpretabilidade. Uma árvore de decisão organiza o processo de tomada de decisão em uma estrutura hierárquica semelhante a um fluxograma, onde nós internos representam atributos (características) dos dados, arestas (galhos) correspondem a testes ou regras de decisão sobre esses atributos, e nós folha representam as classes de saída ou valores estimados (Breiman *et al.*, 1984).

Durante o treinamento, o algoritmo particiona recursivamente o conjunto de dados de treinamento em subconjuntos cada vez mais homogêneos em relação ao rótulo de classe, escolhendo em cada passo um atributo e um ponto de corte que melhor separe os dados de classes diferentes. Para medir a “qualidade” de uma possível divisão dos dados, são utilizadas métricas como ganho de informação ou índice de Gini (medida de impureza, indicando o grau de heterogeneidade de uma amostra, quanto menor o valor, maior a pureza do nó), para selecionar o atributo que produz os grupos mais puros em termos de classe alvo (Breiman *et al.*, 1984).

O crescimento da árvore continua até que algum critério de parada seja atingido, por exemplo, até que todos os nós folha sejam suficientemente puros (contenham majoritariamente amostras de uma única classe) ou até alcançar um limite pré-definido de profundidade da árvore, ajustado antes de iniciar o treinamento. Sem restrições, uma árvore pode se tornar muito complexa e ajustar-se perfeitamente aos dados de treinamento, perdendo capacidade de generalização. Por isso, hiper parâmetros como a profundidade máxima da árvore, o número

mínimo de amostras requerido para subdividir um nó interno, ou o número mínimo de amostras em cada folha são ajustados para limitar o crescimento da árvore. A seleção desses parâmetros pode ser feita por validação cruzada, buscando-se o melhor equilíbrio entre desempenho no treino e validação, de modo a obter um modelo que generalize bem para novos dados.

As SVM são algoritmos de aprendizado supervisionado amplamente aplicados a problemas de classificação, pois buscam construir um hiperplano que maximize a margem entre as classes (Cortes; Vapnik, 1995). Quando utilizadas em sua forma linear, estabelecem uma fronteira de decisão reta no espaço de características, separando duas classes de modo que os pontos de dados mais próximos fiquem o mais afastados possível dessa fronteira. Ao maximizar tal margem, a SVM tende a melhorar a capacidade de generalização do modelo, evitando que pequenas variações nos dados levem a erros de classificação.

Quando os dados não são perfeitamente separáveis, recorre-se à estratégia de margem flexível (*soft margin*), que introduz variáveis de folga e um hiper parâmetro de penalização (C). Esse parâmetro controla o equilíbrio entre largura da margem e erros de treinamento: valores menores de C ampliam a margem, mas toleram mais erros, ao passo que valores maiores priorizam a correção das classificações, podendo reduzir a margem e aumentar o risco de sobreajuste(Cortes; Vapnik, 1995).

Diferentemente de modelos probabilísticos, a SVM devolve apenas a classe prevista; entretanto, o valor da função de decisão (distância ao hiperplano) pode ser interpretado como medida de confiança ou posteriormente calibrado para probabilidades. O treinamento, por sua vez, envolve a resolução de um problema de otimização quadrática sujeito às restrições impostas pelos vetores de suporte, o que pode ser custoso em bases muito grandes. Implementações eficientes, como o *Sequential Minimal Optimization* (SMO), tornam viável seu uso em escala.

A regressão logística é um método estatístico e de aprendizado supervisionado amplamente empregado em problemas de classificação binária, apreciado por sua simplicidade, interpretabilidade e sólida fundamentação teórica em estatística (Hosmer; Lemeshow, 2000).

Como modelo linear generalizado, ela se destina a situações em que a variável dependente assume apenas dois estados possíveis. A estratégia central consiste em combinar linearmente as variáveis de entrada e, em seguida, converter

esse resultado em uma probabilidade por meio da função logística (ou sigmoide), garantindo que o valor obtido esteja sempre entre zero e um (Hosmer; Lemeshow, 2000). Essa transformação equivale a modelar o *logit*, isto é, o logaritmo da razão entre as chances de ocorrência ou não de um evento, como função linear dos preditores (Menard, 1995).

Durante o treinamento, os coeficientes do modelo são ajustados via máxima verossimilhança, procedimento que busca encontrar os parâmetros que tornam os rótulos observados prováveis. Na prática, isso implica minimizar a perda logística por métodos numéricos iterativos, tais como gradiente descendente ou algoritmos quasi-Newton (Hosmer; Lemeshow, 2000).

Uma das maiores virtudes da regressão logística reside na interpretação direta de seus coeficientes, cada peso indica quanto uma variável preditora altera o *logit* da classe positiva, mantendo as demais constantes. Consequentemente, o exponencial de um coeficiente representa o fator pelo qual as chances de pertencer à classe de interesse se modificam quando a variável correspondente aumenta uma unidade (Menard, 1995). Entretanto, a presença de multicolinearidade entre variáveis explicativas pode inflar a variância das estimativas e distorcer essa leitura, exigindo atenção na seleção de atributos (Hosmer; Lemeshow, 2000).

Em cenários de alta dimensionalidade, por exemplo, textos vetorizados com milhares de termos, o modelo se beneficia de técnicas de regularização, que adicionam penalidades aos coeficientes para reduzir o sobreajuste e melhorar a generalização. Penalizações do tipo L2 ou L1 são particularmente úteis para esse fim (Ng, 2004).

A floresta aleatória (do inglês *Random Forest*) é um método de aprendizado em conjunto introduzido por Breiman (2001) que constrói múltiplas árvores de decisão e combina seus resultados para tarefas de classificação ou regressão. A ideia central é gerar uma coleção de árvores “fracas”, porém pouco correlacionadas entre si, e agregá-las para obter um modelo mais forte e robusto (Breiman, 2001).

Para treinar cada árvore, o algoritmo aplica *Bootstrap Aggregating* (*bagging*), amostras de treinamento ligeiramente diferentes são geradas por amostragem com reposição do conjunto original, e cada árvore é ajustada em uma dessas amostras (Breiman, 1996). Além disso, a cada nó de decisão, o algoritmo impõe uma aleatoriedade adicional, em vez de avaliar todos os atributos, seleciona-se aleatoriamente um subconjunto de características e busca-se entre elas o melhor

ponto de corte (Ho, 1995). Essa técnica, conhecida como *random subspace*, promove maior diversidade entre as árvores, pois cada uma explora combinações distintas de variáveis.

A combinação de *bagging* e *random subspace* gera árvores com erros desacoplados. Ao final, suas previsões são consolidadas por voto majoritário (classificação) ou por média (regressão), resultando em uma predição final que geralmente supera a de qualquer árvore individual, graças ao cancelamento de erros não correlacionados e à redução da variância do modelo. Cada árvore pode ter profundidade máxima ou número mínimo de amostras por folha definidos por hiper parâmetros, limites que evitam que as árvores individuais se ajustem demais aos dados. Em conjunto, uma floresta suficientemente diversa tende a não sobreajustar mesmo sem poda, pois o mecanismo de agregação confere regularização natural (Breiman, 2001).

Embora apresente excelente desempenho preditivo, a floresta aleatória perde em interpretabilidade: enquanto uma única árvore produz regras claras, o conjunto de dezenas ou centenas de árvores é difícil de analisar diretamente. Ainda assim, é possível extrair medidas agregadas, para identificar quais variáveis mais contribuem para a previsão (Breiman, 2001).

3 METODOLOGIA

Neste capitulo serão introduzidas as metodologias de pesquisa e projeto que serão utilizadas na construção deste trabalho.

3.1 Metodologia de Pesquisa

A metodologia de pesquisa deste trabalho apoia-se nos princípios do *Design Science Research* (DSR) para orientar de forma sistemática a investigação sobre detecção de interações mal-intencionadas em sistemas de cobrança digital. Inicialmente, caracteriza-se como pesquisa aplicada (Gil, 2002), exploratória e descritiva, pois busca resolver um problema concreto na Monest Cobranças e, ao mesmo tempo, obter maior familiaridade com o fenômeno estudado (Marconi; Lakatos, 2017).

A seguir, apresentam-se as etapas principais com base na metodologia DSR (Dresch; Lacerda; Antunes Júnior, 2015).

1. **Identificação e compreensão do problema**: delimitação do desafio de monitorar e mitigar mensagens mal-intencionadas no sistema de cobrança digital, com imersão no processo da Monest Cobranças e levantamento dos requisitos funcionais e de desempenho necessários para o artefato.
2. **Fundamentação teórica e revisão sistemática da literatura**: levantamento de estudos sobre detecção de anomalias, segurança de sistemas, *autoencoders* e modelos de linguagem, formando a base de conhecimento que orienta o *design* do artefato.
3. **Proposição e desenvolvimento do artefato de pesquisa**: construção iterativa de modelos de aprendizado de máquina para detecção de padrões atípicos, atendendo aos requisitos de eficiência e baixo índice de falsos positivos.
4. **Avaliação analítica e experimental do artefato em ambiente real**: realização de testes com dados históricos e simulações controladas, verificando métricas de acurácia, bem como a percepção dos especialistas da empresa sobre as conversas classificadas.
5. **Análise dos resultados e extração de lições aprendidas**:

interpretação dos indicadores de desempenho do artefato, identificação de limitações, proposta de refinamentos e discussão sobre a contribuição metodológica e prática do estudo, alinhada ao objetivo de gerar conhecimento generalizável e aplicações futuras.

Essa estrutura, pautada no DSR, assegura equilíbrio entre rigor científico e relevância prática, fornecendo direção clara para o desenvolvimento e validação dos modelos de aprendizado de máquina aplicados ao problema proposto.

3.2 Metodologia de Projeto

Para alcançar os objetivos propostos neste estudo, será adotada uma abordagem metodológica estruturada fundamentada no CRISP-DM, reconhecido por sua clareza e eficiência na gestão e condução de projetos analíticos, integrada à flexibilidade característica das metodologias ágeis. Essa integração visa promover ciclos iterativos de desenvolvimento e adaptação constante, contemplando etapas interdependentes que buscam o desenvolvimento de modelos para a detecção de interações disfuncionais em sistemas de cobrança digital via WhatsApp.

Inspirando-se nos princípios ágeis descritos por Schwaber e Sutherland (2020), o trabalho será conduzido por meio de entregas incrementais, permitindo ajustes contínuos ao longo do processo e garantindo alinhamento com os objetivos finais. A seguir, detalha-se cada fase do CRISP-DM utilizada neste projeto.

Durante a etapa inicial de entendimento de negócio, foram definidas duas classes de mensagens para análise no contexto da recuperação de crédito: as mensagens funcionais, que contribuem diretamente para o objetivo do processo de cobrança, e as mensagens disfuncionais, que não contribuem ou até mesmo prejudicam o fluxo eficiente da comunicação e negociação. Como objetivo principal do trabalho a geração de modelos de aprendizado de máquina capazes de classificar se uma mensagem é funcional ou disfuncional.

Durante o entendimento dos dados, as mensagens dos devedores foram coletadas de 15 mil conversas provenientes de uma empresa especializada em recuperação de crédito focada em produtos de cartão de crédito, empréstimo e financiamento pessoal. Os dados contemplam dívidas com atraso entre 16 e 180 dias. Das conversas coletadas, 10 mil geraram pelo menos um acordo de pagamento, enquanto as demais não resultaram em acordos. Durante o processo de

coleta, duas conversas (uma de cada grupo) continham mensagens que não puderam ser processadas utilizando os padrões de codificação UTF-8 e ISO/IEC 8859-1.

Para visualizar o comprimento das mensagens em palavras, foi aplicada uma transformação logarítmica nos comprimentos das mensagens, definida matematicamente como log(x + 1), sendo x a quantidade de palavras da mensagem, visando normalizar as distribuições (Fielding, 2000). Essa técnica é frequentemente utilizada para corrigir distribuições assimétricas positivas, típicas em comunicações digitais, onde há concentração de valores baixos e caudas longas com valores extremos (Jurafsky; Martin, 2025). A transformação logarítmica tem a propriedade de comprimir a escala dos valores maiores, mantendo a diferenciação dos valores menores (Tabachnick; Fidell, 2019).

Também nesta pesquisa foram implementadas três abordagens distintas de análise de sentimentos utilizando bibliotecas *Python*. A primeira metodologia empregou o VADER (*Valence Aware Dictionary and sEntiment Reasoner*), uma ferramenta baseada em modelo léxico (Palomino; Aider, 2022) da biblioteca *vaderSentiment* que analisa textos por dicionários pré-definidos, gerando scores positivo, negativo e composto (entre -1 e +1). A segunda abordagem utilizou o modelo BERT multilíngue (*nlptown/bert-base-multilingual-uncased-sentiment*) via biblioteca *transformers* do Hugging Face, aplicando aprendizado profundo supervisionado para classificar mensagens em categorias discretas baseadas em sistema de estrelas (1-2 = negativo, 3 = neutro, 4-5 = positivo). A terceira metodologia implementou classificação *zero-shot* com o modelo *MoritzLaurer/mDeBERTa-v3-base-mnli-xnli*, que utiliza inferência natural de linguagem (NLI) treinada em tarefas XNLI (Cross-lingual Natural Language Inference) para classificar textos sem necessidade de treinamento específico em dados de sentimento, avaliando a probabilidade de cada mensagem pertencer às categorias "*positive*", "*neutral*" ou "*negative*".

Inicialmente, durante a preparação dos dados foram realizadas as seguintes operações:

- Remoção de *stopwords*, combinando listas do *Natural Language Toolkit* (NLTK) e do *spaCy* para garantir uma cobertura abrangente de termos comuns em português.
- Lemmatização com o modelo *pt_core_news_lg* do *spaCy*, visando obter a

forma básica das palavras e reduzir a variabilidade dos termos.

- *Stemming* utilizando o algoritmo RSLP do NLTK para reduzir as palavras ao seu radical.

Foram construídos conjuntos de dados baseados nas técnicas de TF-IDF e BoW, aplicando seleção de *features* com o método Chi-quadrado (x^2) para manter as características mais relevantes. O teste do Chi-quadrado é uma técnica estatística utilizada para avaliar a independência entre duas variáveis categóricas, auxiliando na seleção das *features* mais discriminativas para o modelo preditivo (Agresti, 2007).

Além disso, foram geradas representações vetoriais usando *embeddings* BERT com o modelo "*nlptown/bert-base-multilingual-uncased-sentiment*", produzindo vetores densos que capturam relações contextuais profundas entre as palavras. Cabe destacar uma limitação enfrentada na utilização de diferentes modelos de linguagem, devido a uma vulnerabilidade crítica identificada na função *torch.load*, exigindo a atualização do *Torch* para pelo menos a versão 2.6.1 que não estava disponível para mitigar riscos de segurança (NVD, 2025).

Resultando nos seguintes conjuntos de dados:

BoW (Todos sem pontuações):

- Lemmatizado;
- Lemmatizado sem *stopwords*;
- *Stemming*;
- *Stemming* sem *stopwords*.

TF-IDF (Todos sem pontuações):

- Lemmatizado;
- Lemmatizado sem *stopwords*;
- *Stemming*;
- *Stemming* sem *stopwords*.

BERT:

- Conteúdo original.

Para a modelagem, treinamento e avaliação dos modelos, o conjunto de dados foi dividido em duas partes, utilizando a técnica de *hold-out*: 80% para treino e 20% para validação. A divisão foi feita de forma estratificada, considerando não apenas o rótulo da classe (contexto, funcional ou disfuncional), mas também o grupo

associado a cada mensagem. Essa estratégia assegura que a proporção de mensagens funcionais e disfuncionais de cada grupo seja preservada em ambas as partições, evitando que todas as mensagens de um grupo fiquem concentradas em apenas uma das partes o que poderia introduzir viés ou sobreajuste por memorização de padrões específicos.

Além dos modelos supervisionados tradicionais, também foi aplicada uma abordagem de detecção de anomalias focada nas representações densas (*embeddings*). A motivação é que mensagens disfuncionais podem ser tratadas como anomalias ou eventos raros fora do padrão normal de comunicação. Duas técnicas foram empregadas nesse sentido.

Para avaliar quantitativamente o desempenho dos modelos de classificação, foram empregadas métricas de classificação baseadas na matriz de confusão, incluindo Acurácia, F1 e F1 Ponderada, sendo que a acurácia é a proporção de predições corretas do modelo em relação ao total de exemplos avaliados. Em um problema binário, a acurácia pode ser calculada como:

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

Onde TP (verdadeiros positivos) é o número de positivos verdadeiros (mensagens corretamente classificadas como funcionais), TN (verdadeiros negativos) é o número de negativos verdadeiros (disfuncionais corretamente classificados), FP (falsos positivos) é o número de mensagens disfuncionais incorretamente classificadas como funcionais e FN (falsos negativos) são mensagens funcionais classificadas como disfuncionais.

Essa métrica indica a taxa global de acerto do modelo. Porém, vale ressaltar que a acurácia por si pode ser enganosa em cenários de classes desbalanceadas. Portanto, outras métricas que consideram especificamente os acertos por classe são necessárias para uma avaliação mais confiável.

A precisão mede a exatidão das predições positivas do modelo, ou seja, dentre todas as instâncias que o modelo predisse como pertencentes a uma classe (por exemplo, predisse "funcional"), quantas de fato eram daquela classe. Especificamente em relação à classe funcional (positiva), a precisão é calculada como:

$$Precisão = \frac{TP}{TP+FP},$$

Representando a proporção de verdadeiros positivos em relação a todos os

positivos preditos. Uma alta precisão significa que poucas mensagens marcadas pelo modelo como funcionais eram, na verdade, disfuncionais, em outras palavras, que o modelo comete poucos erros de falso positivo. No contexto de recuperação de crédito, uma alta precisão para a classe funcional assegura que quando o modelo diz que uma mensagem está no contexto da cobrança, há grande confiança de que realmente esteja, evitando classificar indevidamente mensagens inadequadas como úteis ao processo.

A revocação, também chamada de Sensibilidade, é a capacidade do modelo de recuperar os exemplos verdadeiros positivos. Ou seja, dentre todas as instâncias que deveriam ser identificadas como de uma certa classe (por exemplo, todas as mensagens funcionais existentes no conjunto), quantas o modelo de fato conseguiu identificar corretamente. Matematicamente, para a classe funcional, a revocação é dada por:

$$Revocação = \frac{TP}{TP+FN},$$

Que representa a proporção de verdadeiros positivos em relação ao total de positivos reais. Um recall alto indica que o modelo consegue cobrir a maioria dos exemplos da classe positiva, cometendo poucos erros de falso negativo. No nosso problema, uma alta revocação para funcionais significa que a maioria das mensagens relevantes ao contexto de cobrança foi reconhecida como tal pelo modelo, ele não deixa muitas mensagens funcionais escaparem sem identificação. Por outro lado, um recall baixo apontaria que o modelo está deixando passar muitos casos que deveria pegar (por exemplo, classificando mensagens funcionais como disfuncionais, o que poderia interromper fluxos válidos de cobrança).

A Medida F1 é a métrica que resume precisão e revocação em um único valor, calculando a média harmônica entre os dois. A fórmula do F1 é dada por:

$$F1 = 2 \cdot \frac{Precisão \times Revocação}{Precisão + Revocação},$$

Que produz um valor alto somente quando ambas precisão e revocação são altas. O F1 é especialmente útil em casos de classes desbalanceadas, pois penaliza situações em que um dos componentes (precisão ou revocação) está baixo. Por exemplo, se um modelo obtém precisão=0,9 e revocação=0,1 para a classe positiva, a média aritmética seria 0,5, mas o F1 resultará em aproximadamente 0,18, refletindo melhor o desempenho insatisfatório nesse caso extremo.

Como estamos tratando de duas classes (funcional vs disfuncional), é

possível calcular as métricas acima para cada classe separadamente e depois combiná-las para ter uma visão geral do desempenho. Focamos em duas variantes de agregação: macro média e média ponderada.

A média macro calcula a métrica independentemente para cada classe e então toma a média aritmética desses valores. Ou seja, precisão macro seria a média da precisão da classe funcional e da precisão da classe disfuncional, revocação macro a média das revocações de cada classe e F1 macro a média dos F1 de cada classe. Todas as classes têm igual peso nessa média, não importando seu tamanho. Esse indicador macro valoriza o desempenho em ambas as classes de forma equilibrada. No nosso caso, como há forte desequilíbrio (bem mais mensagens funcionais que disfuncionais), a métrica macro fornece uma avaliação justa, demonstrando se o modelo consegue performar bem também na classe minoritária. Um F1-macro alto significa que o modelo está apresentando bons resultados tanto para funcionais quanto para disfuncionais em média.

Já a média ponderada calcula a métrica para cada classe, porém faz uma média ponderada pelo suporte (número de exemplos) de cada classe. Assim, classes com mais instâncias influenciam mais no valor final. No caso binário, a média ponderada de precisão, recall ou F1 tende a ficar mais próxima do valor da classe majoritária, já que esta domina em número de exemplos. A vantagem da métrica ponderada é que ela reflete o desempenho geral do modelo considerando a proporção real de cada classe, ou seja, é uma medida do desempenho global do modelo. Por exemplo, se 95% dos dados são da classe funcional, o F1 ponderado estará muito próximo do F1 dessa classe, mas ainda será penalizado pelos erros na classe minoritária proporcionalmente.

Para avaliar o *autoencoder*, foi calculado duas medidas principais de discrepância entre a entrada e a reconstrução da rede: a similaridade de cosseno e o erro quadrático médio de reconstrução. A similaridade de cosseno mede o quão alinhados estão o vetor original x e o vetor reconstruído $\hat{x}$ no espaço de características, sendo calculada como:

$$cosine\ (x,\ \hat{x})\ =\ \frac{x \cdot \hat{x}}{||x||\ ||\hat{x}||},$$

Onde $x \cdot \hat{x}$ é o produto interno entre os vetores e $|x|$ denota a norma euclidiana, essa métrica varia de -1 à 1, onde os valores próximos de 1 indicam que

a reconstrução mantém alta semelhança direcional com o original. Já o erro quadrático médio (MSE) é computado como:

$$MSE(x, \hat{x}) = \frac{1}{d}\Sigma_{j=1}^{d}(x_j - \hat{x}_j)^2,$$

Onde d é a dimensionalidade (768), x_j e $\hat{x}_j$ são os componentes de cada vetor. o MSE quantifica a magnitude do erro de reconstrução em valores absolutos. Intuitivamente, espera-se que para mensagens funcionais (semelhantes às do treino) o *autoencoder* reconstrua bem (alta similaridade de cosseno, baixo erro), enquanto para mensagens disfuncionais (diferentes do padrão aprendido) a reconstrução tenda a ser pior (menos similaridade, maior erro).

Para avaliar cada *One-ClassSVM*, calculamos o escore de decisão do modelo para todas as amostras de validação. O escore de decisão do *One-ClassSVM* é um valor real onde valores positivos (no contexto padrão do *scikit-learn*) indicam que a amostra é considerada similar aos dados de treino (*inliers*) e valores negativos indicam anomalias (*outliers*) conforme o modelo. No caso do modelo treinado com contexto=*True* (funcionais), esperamos que ele produza escores positivos para mensagens funcionais e escores negativos para disfuncionais; já o modelo treinado em disfuncionais invertidamente trataria mensagens funcionais como anomalias. Para quantificar o quão bem cada modelo separa as duas classes, utilizamos a métrica ROC AUC (Receiver Operating Characteristic - Area Under the Curve). Calculamos a AUC tomando as classes verdadeiras (funcional ou disfuncional) e os escores do *One-ClassSVM* como entrada. No caso do modelo treinado em funcionais, consideramos a classe positiva como "funcional" (esperando escores maiores para estas) e no caso do modelo treinado em disfuncionais invertidos o sinal dos escores (multiplicando por -1) e consideramos "disfuncional" como classe positiva, de forma que em ambos os cenários um valor de AUC alto (próximo de 1.0) indica boa separação entre as duas classes pelo modelo de anomalia.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Durante a classificação inicial, a tarefa mostrou-se especialmente complexa devido à natureza volátil da língua portuguesa, onde expressões coloquiais, gírias regionais e diferentes formas de negação (como "não posso", "não consigo", "não agora") são comuns em diálogos legítimos de negociação de dívidas. Portanto, essas expressões não podem ser consideradas ruído apenas pela ausência explícita de positividade. Essa particularidade linguística demandou uma estratégia inicial cautelosa, conectando diretamente ao processo de rotulamento.

Para garantir a confiabilidade dos dados rotulados, as mensagens foram inicialmente analisadas manualmente para compreensão do contexto e posteriormente classificadas automaticamente por meio da API da OpenAI, especificamente como "tue" (mensagens funcionais, alinhadas ao fluxo de cobrança) ou "false" (mensagens disfuncionais, não alinhadas ao fluxo de cobrança). Esse processo automatizado foi realizado utilizando o modelo gpt-4o-mini-2024-07-18, com temperatura definida em 0 para reduzir ao máximo a aleatoriedade das classificações e assegurar resultados consistentes.

A implementação técnica adotou requisições assíncronas gerenciadas por um semáforo, limitando a simultaneidade a 75 requisições para garantir eficiência e estabilidade do sistema. O método few-shot foi utilizado, apresentando-se ao modelo exemplos claros e pré-rotulados para direcionar adequadamente a classificação das mensagens, conforme detalhado no *prompt* apresentado no Quadro 1.

Quadro 1 – Instruções de Classificação

Você receberá uma única mensagem trocada entre uma empresa de cobrança e um cliente. Sua tarefa é atribuir exatamente um destes dois rótulos:

- true → se a mensagem pertencer ao fluxo de recuperação de crédito (cobrança/negociação), incluindo saudações simples que apareçam isoladas, pois serão consideradas parte do diálogo.
- false → se a mensagem for fora de contexto (insultos, spam, papo irrelevante, pedidos sem relação com a dívida).

DEFINIÇÕES E EXEMPLOS

1. true
Qualquer resposta que demonstre engajamento na cobrança ou negociação da dívida, mesmo que seja apenas uma saudação simples e isolada. Exemplos:
- "oi"
- "olá"
- "bom dia"
- "Não posso pagar agora, mas posso em 5 dias"
- "Posso parcelar em 12x?"
- "Já paguei hoje."
- "Preciso que seja parcelado."
- "Em 12x."
- "Sim, estou de acordo."
- "Me manda o boleto."
- "Pode"
- "ok"
- "👍"

Justificativas pessoais ("estou desempregado", "não tenho dinheiro") e emojis de confirmação (👍, ✅) ou negação (❌) também contam como true se estiverem vinculados à cobrança ou negociação.

2. false
Qualquer mensagem sem relação alguma com cobrança:
- Insultos ("vá se ferrar", "palhaçada")
- Papo irrelevante ("kkkk", "vou ao mercado")
- Pedidos de promoção ou ofertas alheias ("quero desconto na loja", "me manda cupom")
- Reclamações ou dúvidas que não mencionem dívida, pagamento nem justificativa pessoal ligada à cobrança.
- Qualquer conteúdo que não avance nem responda à negociação de dívida.

INSTRUÇÕES
Analise todo o conteúdo da mensagem.
Escolha apenas um rótulo: true ou false.
Retorne somente o rótulo, sem explicações nem texto adicional.

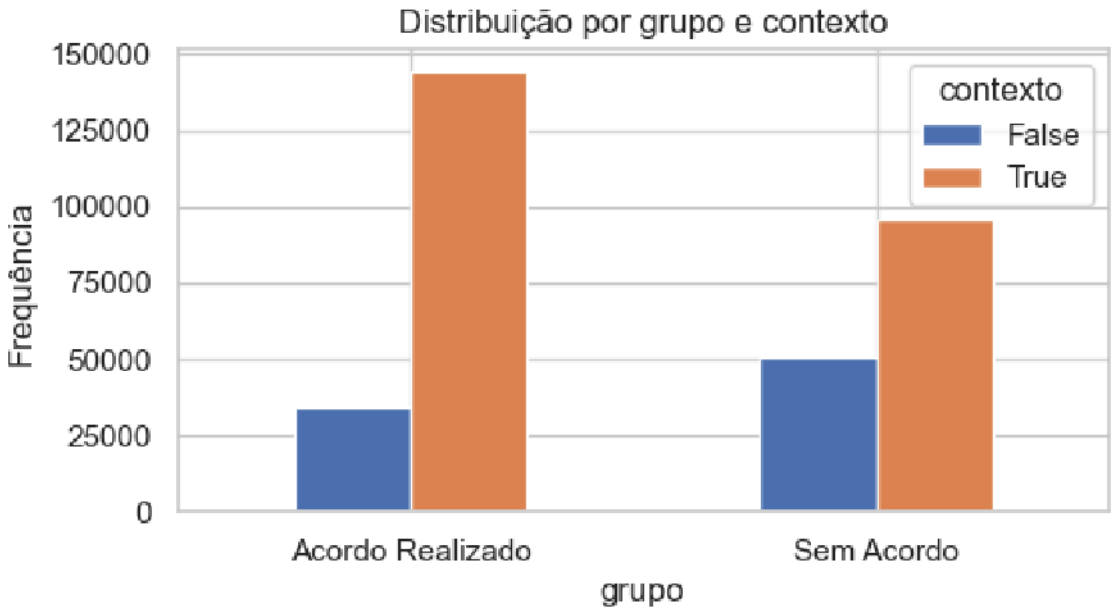
Mensagem: {}

Fonte: elaborado pelo autor (2025).

Ao analisar a distribuição das mensagens por grupo e contexto com mensagens funcionais representadas por *True* e disfuncionais por *False*, observa-se que ambos os grupos possuem uma predominância de mensagens funcionais. No entanto, a proporção de mensagens disfuncionais é significativamente maior no grupo "Sem Acordo", o que sugere uma associação entre a presença de mensagens

disfuncionais e a não realização de acordos, descrito na Figura 3.

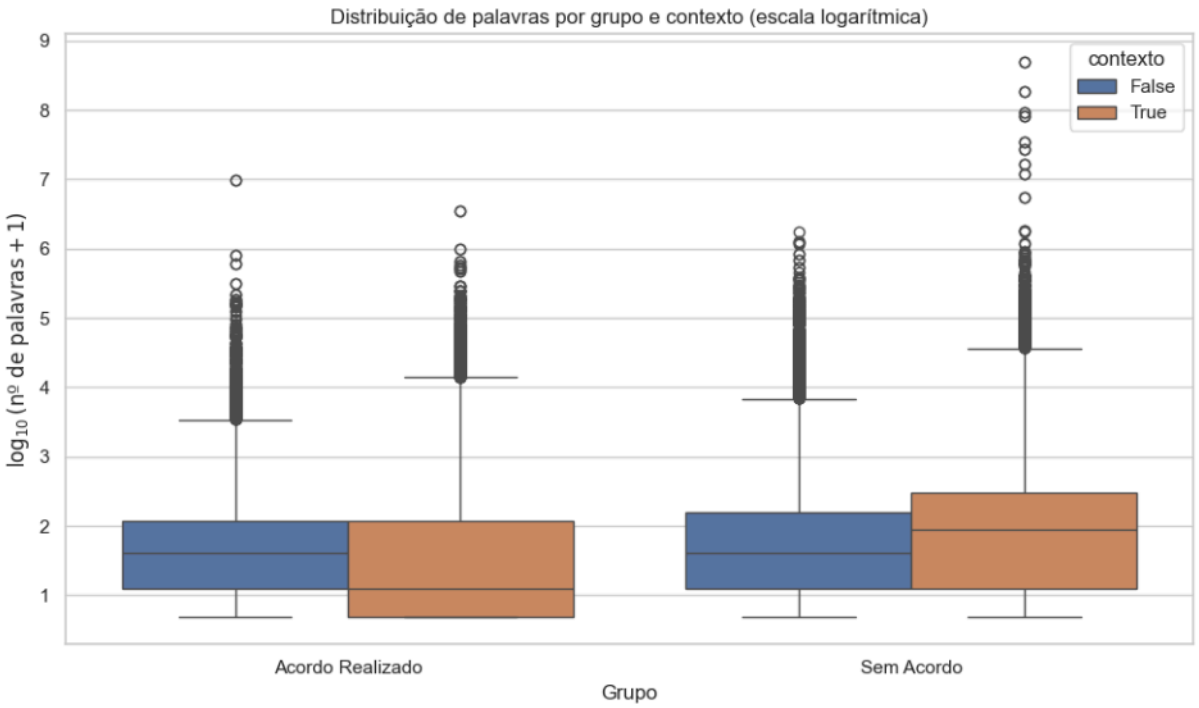
Figura 3 — Distribuição por grupo e contexto



Fonte: elaborado pelo autor (2025).

Após aplicar a transformação logarítmica, é possível visualizar que as mensagens ficam concentradas entre 1 à 3 palavras na Figura 4.

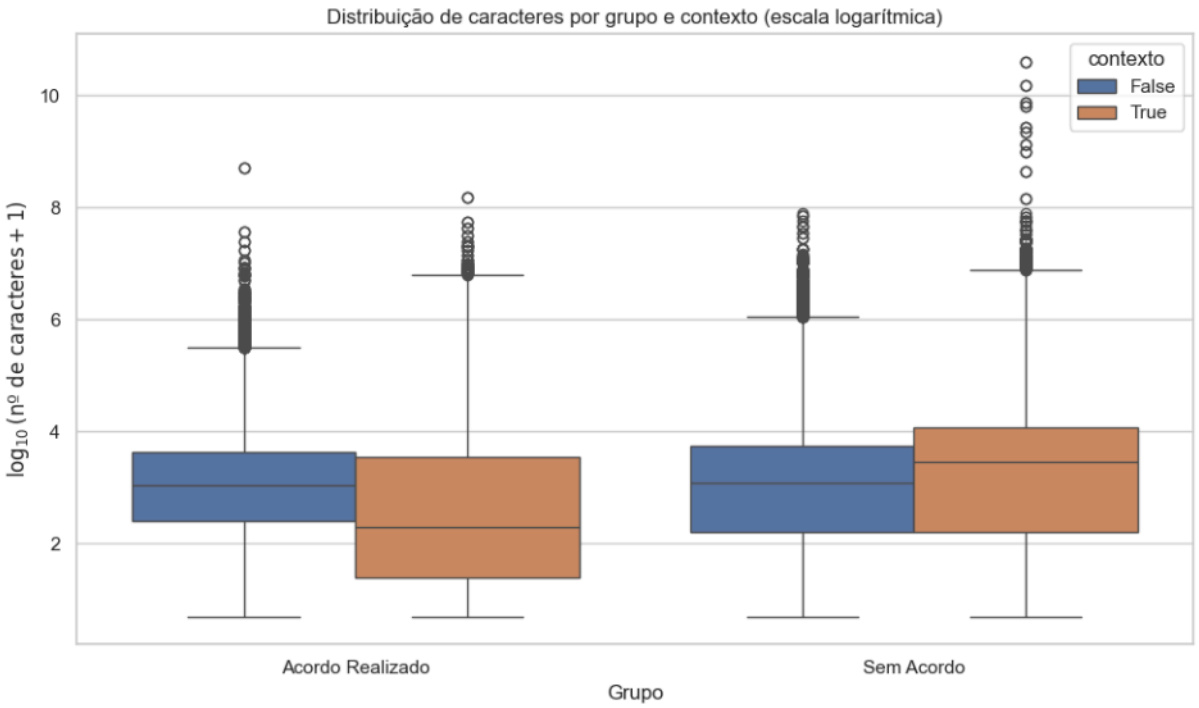
Figura 4 — Distribuição de Palavras



Fonte: elaborado pelo autor (2025).

E ao analisar a quantidade de caracteres por mensagens, após ter aplicado a transformação logarítmica, ficando concentradas entre 2 à 4 caracteres, demonstrado na Figura 5.

Figura 5 — Distribuição de Caracteres



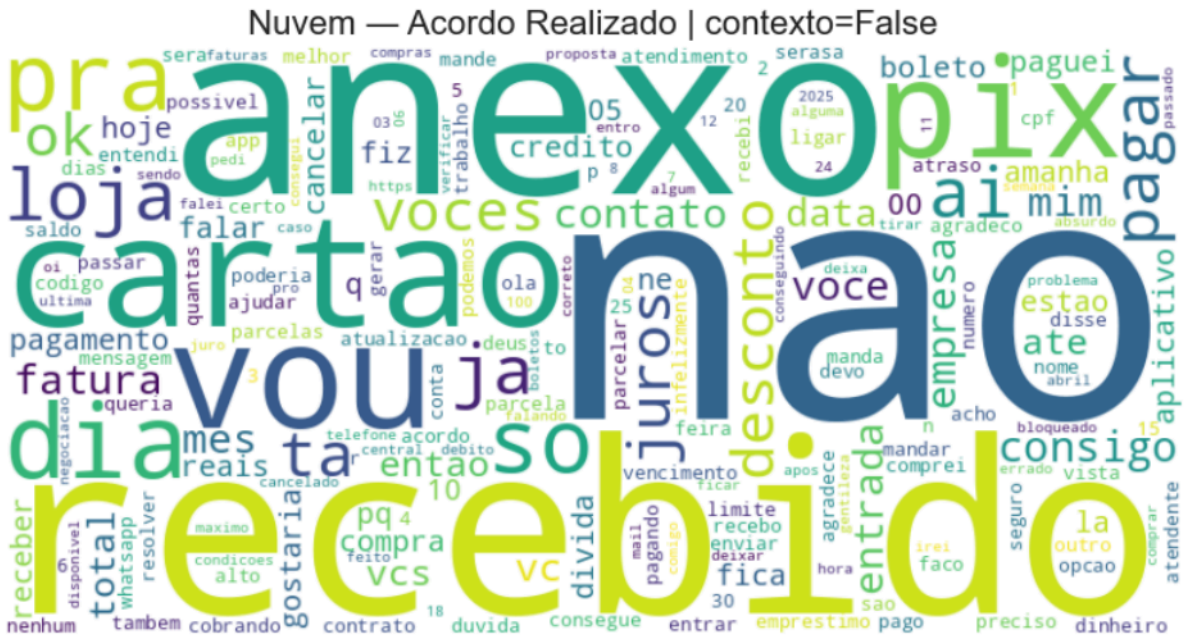
Fonte: elaborado pelo autor (2025).

Adicionalmente, para explorar visualmente os termos mais frequentes utilizados nas mensagens, foi empregada a técnica de nuvem de palavras. As nuvens de palavras foram geradas separadamente para cada combinação dos grupos e contextos: grupo "Acordo Realizado" com contexto *True* e *False*, e grupo "Sem Acordo" também com contexto *True* e *False*, permitindo identificar e comparar os termos mais recorrentes em cada contexto específico, nas Figuras 6 e 7 é possível ver as nuvens de palavras do grupo “Acordo Realizado” com ambos os contextos.

Figura 6 — Acordo Realizado no Contexto

Fonte: elaborado pelo autor (2025).

Figura 7 — Acordo Realizado Fora de Contexto

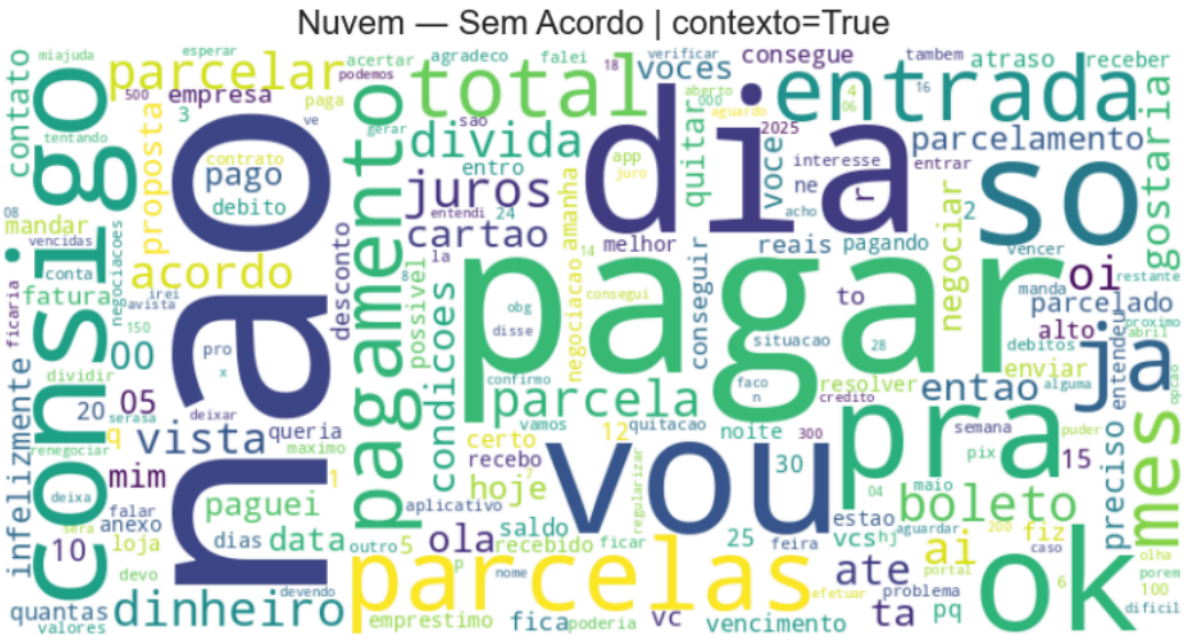


Fonte: elaborado pelo autor (2025).

As nuvens de palavras para mensagens funcionais e disfuncionais do grupo Acordo Realizado são bem parecidas, denotando pouca diferença de vocabulário entre os 2 contextos. Abaixo, nas Figuras 8 e 9, as mensagens do grupo “Sem

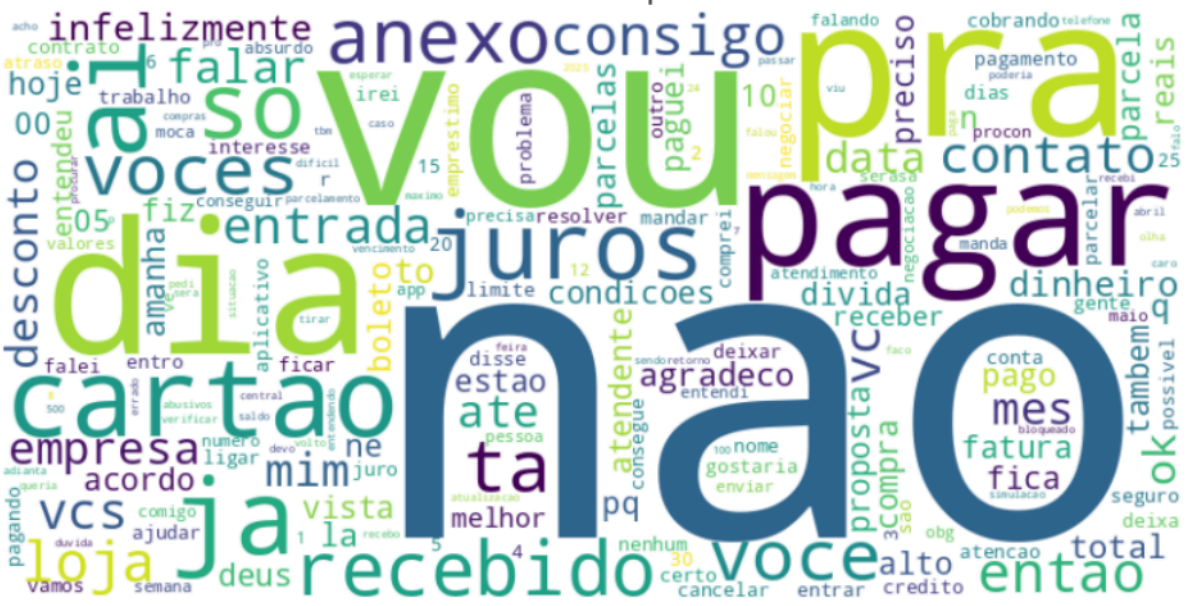
Acordo” com ambos os contextos.

Figura 8 — Sem Acordo no Contexto



Fonte: elaborado pelo autor (2025).

Figura 9 — Sem Acordo Fora de Contexto



Fonte: elaborado pelo autor (2025).

As nuvens de palavras para mensagens funcionais e disfuncionais do grupo “Sem Acordo” são bem parecidas, denotando pouca diferença de vocabulário entre os 2 contextos.

Para avaliar o comportamento emocional nas interações de cobrança, buscou-se identificar se mensagens classificadas como sem acordo ou disfuncionais possuíam uma carga emocional mais negativa em comparação àquelas com acordo e funcionais. Para alcançar esse objetivo, foram aplicadas três metodologias distintas de análise de sentimentos. A primeira abordagem utilizou o modelo VADER, que forneceu scores médios compostos para cada cenário de interação. A segunda metodologia empregou o modelo BERT multilíngue, categorizando individualmente cada mensagem como negativa, neutra ou positiva. Por fim, foi aplicada uma abordagem *zero-shot* baseada em inferência textual para classificar as mensagens conforme suas nuances emocionais. Os resultados dessas análises estão detalhados nas Tabelas 1, 2 e 3.

Tabela 1 – Análise VADER

Grupo	Contexto	Comprimento Médio (Tokens)	Positivo	Negativo	Composto
Acordo Realizado	*False*	6,5120	0,0198	0,0134	0,0034
Acordo Realizado	*True*	5,2314	0,0756	0,0075	0,0214
Sem Acordo	*False*	7,7224	0,0153	0,0165	-0,0045
Sem Acordo	*True*	9,1744	0,0536	0,0156	-0,0002

Fonte: dados da pesquisa.

Tabela 2 – Análise BERT

Grupo	Contexto	Negativo	Neutro	Positivo
Acordo Realizado	*False*	11.448	4.856	4.480
Acordo Realizado	*True*	23.536	41.229	11.681
Sem Acordo	*False*	20.542	3.615	4.974
Sem Acordo	*True*	24.894	16.959	7.919

Fonte: dados da pesquisa.

Tabela 3 – Análise *Zero-Shot Classification*

Grupo	Contexto	Negativo	Neutro	Positivo
Acordo Realizado	False	12.489	1.698	6.597
Acordo Realizado	True	39.374	1.120	35.952
Sem Acordo	False	17.086	4.051	7.994
Sem Acordo	True	26.103	1.849	21.900

Fonte: dados da pesquisa.

Os resultados das três metodologias revelaram uma convergência significativa, fortalecendo a confiança nos achados e confirmando a hipótese inicial. Nas conversas em que o acordo foi realizado, o modelo VADER registrou scores compostos levemente positivos (0,0034 e 0,0214), o modelo BERT apontou uma predominância clara de mensagens neutras (41.229 mensagens no cenário funcional), e a análise *zero-shot* indicou uma distribuição emocional mais equilibrada, porém com leve superioridade de mensagens positivas (35.952 positivas contra 39.374 negativas). Por outro lado, nas conversas sem acordo, observou-se um evidente aumento da carga emocional negativa em todas as técnicas utilizadas: o VADER apresentou scores médios próximos de zero ou negativos (-0,0045 e -0,0002), o BERT revelou aumento das mensagens classificadas como negativas, e a abordagem *zero-shot* confirmou esse padrão, registrando expressiva predominância de mensagens negativas (17.086 e 26.103). Essas constatações reforçam que mensagens sem acordo e disfuncionais possuem, de fato, um tom emocional significativamente mais negativo quando comparadas àquelas com acordo e funcionais.

Foi treinado um classificador do tipo árvore de decisão (implementação do *scikit-learn*). Optou-se por limitar a profundidade máxima da árvore (*max_depth*=25) e exigir um mínimo de 20 amostras por folha terminal (*min_samples_leaf*=20). Esses hiper parâmetros atuam como forma de regularização, prevenindo que a árvore fique excessivamente complexa e sobreajuste nos dados de treino. Além disso, usou-se *class_weight*="*balanced*" para compensar eventuais desequilíbrios de classe. Após o treinamento, a árvore de decisão foi salva para posterior análise.

Em seguida, foi treinado um modelo do tipo floresta aleatória. No

experimento, utilizou-se uma floresta com *n_estimators*=300 árvores, empregando todos os núcleos de processamento disponíveis (*n_jobs*=-1 para agilizar o treinamento). O parâmetro de balanceamento de classe também foi ativado (*class_weight*="*balanced*"), assegurando que cada árvore receba amostras reponderadas de modo a tratar ambas as classes com equidade. O modelo treinado foi então salvo.

Também treinamos um classificador linear de regressão logística multinomial. Configuramos o modelo com *solver* SAGAs (*solver*="*saga*") apropriado para conjuntos esparsos grandes, permitimos até 2000 iterações para a convergência (*max_iter*=2000), e utilizamos todos os núcleos (*n_jobs*=-1). Assim como nos anteriores, aplicou-se *class_weight*="*balanced*". O modelo resultante foi salvo.

O quarto modelo supervisionado foi um SVM linear (implementado como *LinearSVC* do *scikit-learn*). O parâmetro de regularização foi mantido em C = 1.0 (padrão), e novamente *class_weight*="*balanced*" foi definido para lidar com o desbalanceamento de classes, o modelo foi treinado e salvo.

Cada um desses modelos foi treinado separadamente em cada representação de atributos descrita anteriormente. Ao final do treinamento, para cada modelo e para cada tipo de representação, foram obtidas as predições no conjunto de validação e calculadas métricas de desempenho.

Além dos modelos supervisionados tradicionais, também foi aplicada uma abordagem de detecção de anomalias focada nas representações densas (*embeddings*). A motivação é que mensagens disfuncionais podem ser tratadas como anomalias ou eventos raros fora do padrão normal de comunicação, duas técnicas foram empregadas nesse sentido.

Foi desenvolvida uma pequena rede neural do tipo *autoencoder* para aprender a reconstruir as mensagens funcionais e, assim, detectar indiretamente mensagens disfuncionais. Neste caso, o *autoencoder* foi implementado em *PyTorch* com arquitetura densa totalmente conectada, a entrada é um vetor de dimensão 768 (correspondente ao *embedding* de uma mensagem), em seguida duas camadas ocultas diminuem para 256 neurônios e depois 128 neurônios (este último sendo o gargalo que força a rede a aprender uma codificação compacta dos dados) depois, o *decoder* reconstrói de 128 de volta para 256 e finalmente para 768 neurônios, tentando reproduzir o vetor original. A função de ativação ReLU foi usada nas camadas ocultas.

Para treinar o *autoencoder*, utilizamos somente exemplos positivos (mensagens funcionais) do conjunto de treino, desta forma a rede aprende a codificar e reconstruir apenas o comportamento "normal" das mensagens de cobrança. O treinamento foi feito por 10 épocas, em lotes de tamanho 256, otimizando o erro quadrático médio de reconstrução entre a entrada e a saída. Utilizou-se o otimizador AdamW com taxa de aprendizado 0.001 para atualizar os pesos. Ao final do treinamento, os pesos do *autoencoder* foram salvos.

Após treinado, o *autoencoder* foi avaliado usando dados de validação para verificar sua capacidade de distinguir anomalias. Para isso, foi passado para o modelo apenas mensagens disfuncionais presentes no conjunto de validação que seriam considerados anomalias esperadas.

Observando os modelos que melhor performaram em cada vetorização, com o intuito de encontrar o melhor modelo, no quadro 2 foi anotado qual modelo teve a melhor desempenho para cada uma das métricas.

Quadro 2 – Modelos que tiveram melhor desempenho por métrica

Representação	Tratamento	Acurácia	F1 Macro	F1 Ponderado	Precisão
BERT	Sem tratamento	Floresta	Linear SVC	Floresta	Floresta
BoW	Lematizado	Floresta	Floresta	Floresta	Floresta
	Lematizado sem *stopwords*	Floresta	Floresta	Floresta	Floresta
	Stemming	Floresta	Floresta	Floresta	Floresta
	Stemming sem *stopwords*	Floresta	Floresta	Floresta	Floresta
TF-IDF	Lematizado	Floresta	Floresta	Floresta	Floresta
	Lematizado sem *stopwords*	Floresta	Floresta	Floresta	Floresta
	Stemming	Floresta	Floresta	Floresta	Floresta
	Stemming sem *stopwords*	Floresta	Floresta	Floresta	Floresta

Fonte: dados da pesquisa.

Os experimentos revelaram diferenças claras de desempenho entre os modelos e representações testados. De modo geral, a floresta aleatória obteve as

melhores pontuações em quase todos os cenários de vetorização, seguida pela regressão logística em segundo lugar e pelo Linear SVC em terceiro. Os resultados das métricas das Florestas podem ser observados na Tabela 4.

Tabela 4 – Resultados das Florestas

Representação	Tratamento	Acurácia	F1 Macro	F1 Ponderado	Precisão
BERT	Sem tratamento	0,8700771763	0,8239456839	0,8630018567	0,8717117949
BoW	Lematizado	0,8114856429	0,752730394	0,8049671619	0,774699511
	Lematizado sem *stopwords*	0,80989672	0,7512766748	0,8036064568	0,7720193179
	Stemming	0,9058279423	0,8820019994	0,9049809381	0,8892890448
	Stemming sem *stopwords*	0,8414765634	0,7951594135	0,8373724422	0,8132446552
TF-IDF	Lematizado	0,8152877085	0,7537542039	0,8071009374	0,7835052923
	Lematizado sem *stopwords*	0,8146067416	0,7530144248	0,806466757	0,7823479909
	Stemming	0,9111905573	0,8881575616	0,9101536836	0,8979211633
	Stemming Sem *stopwords*	0,8491090682	0,8025662674	0,8441099058	0,8271901565

Fonte: dados da pesquisa.

Entre as representações textuais, destacaram-se TF-IDF com *stemming* e BoW com *stemming*, ambas mantendo as *stopwords*, além da abordagem com embeddings de BERT, que apresentou desempenho ligeiramente inferior às melhores combinações tradicionais.

A configuração TF-IDF + *stemming* alimentando uma floresta aleatória obteve aproximadamente 91% de acurácia, F1 macro em torno de 0,888 e F1 ponderado de 0,910, constituindo o melhor resultado geral. Muito próxima veio BoW + *stemming*, também com lloresta aleatória, atingindo cerca de 90% de acurácia e F1 macro de 0,882. A representação BERT alcançou em torno de 87% de acurácia e F1 macro de

0,824; embora robusta, não superou os métodos baseados em frequência de termos. As variações com lematização ou com remoção de *stopwords* mostraram desempenho inferior: a lematização reduziu a acurácia para a faixa de 81%–82% e remover *stopwords* causou nova queda de alguns pontos percentuais. Isso indica que, para este conjunto de dados, o agrupamento agressivo de variantes por *stemming* foi mais eficaz e que certas *stopwords* continham sinais úteis ao classificador.

Os resultados sugerem que técnicas clássicas de vetorização, quando combinadas como a floresta aleatória, continuam competitivas mesmo diante de modelos de linguagem avançados. A pequena diferença entre acurácia e F1 ponderado nas melhores configurações sugere equilíbrio razoável entre as classes, embora o F1 macro um pouco menor aponte espaço para melhorar a classe minoritária. A adoção de *stemming* foi decisiva para elevar a performance, enquanto a lematização não trouxe ganhos e, neste domínio, a remoção de *stopwords* mostrou-se contraproducente. Apesar de não ter liderado, o BERT manteve desempenho elevado e poderia superar os métodos tradicionais caso fosse afinado com mais dados ou épocas, exigindo, porém, maior custo computacional.

Enquanto para os resultados dos modelos não supervisionados, temos as seguintes métricas:

- Modelo: *Autoencoder*
    - Média da similaridade de cosseno (contexto=False): 0,92825526.
    - Desvio-padrão da similaridade de cosseno (contexto=*False*): 0,04520174861.
    - Média do erro quadrático médio (MSE) (contexto=*False*): 0,1482542008.
    - Desvio-padrão do MSE (contexto=*False*): 0,1157932132.
    - Número de amostras (contexto=*False*) avaliadas: 9985.

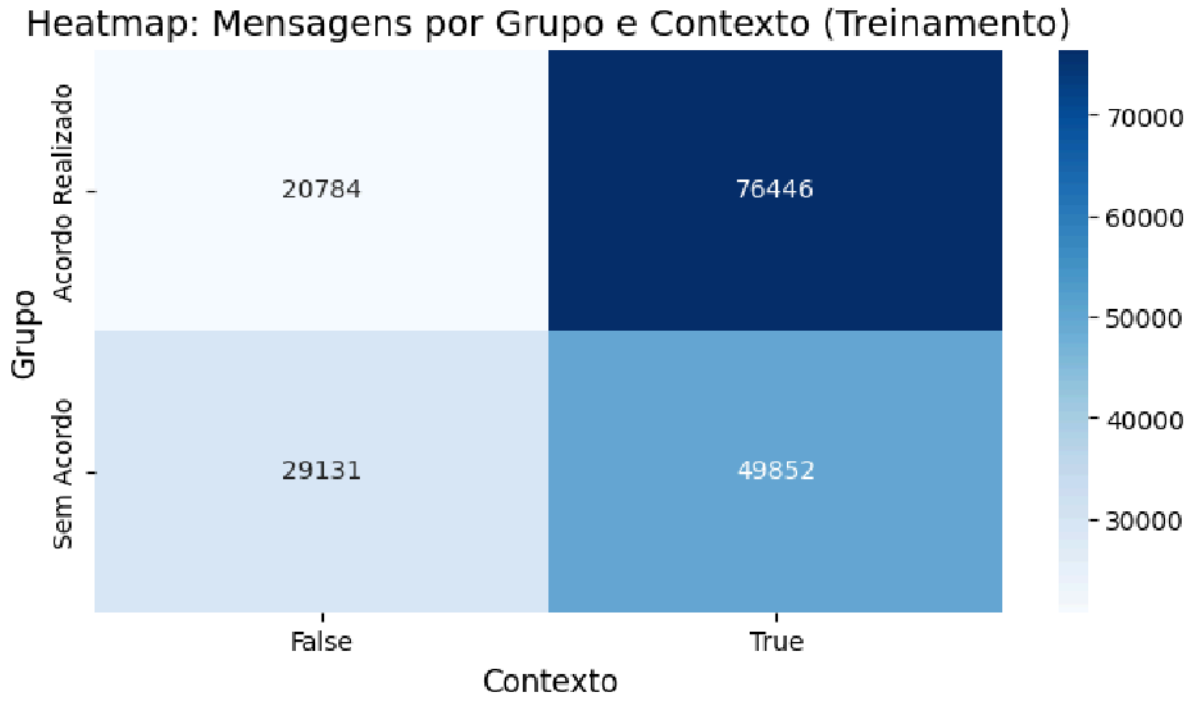
- Modelo: *One-ClassSVM*
    - AUC-ROC treinado em (contexto=*True*): 0,6735829049.
    - AUC-ROC treinado em (contexto=*False*): 0,6113447439.

Esses resultados adicionais complementam a análise introduzindo métricas de detecção de anomalias aplicadas à representação BERT. Para o *autoencoder*, a

média da similaridade de cosseno (~0,93) indica que, mesmo ao reconstruir mensagens disfuncionais, a rede manteve alta coerência direcional entre entrada e saída, o erro quadrático médio (~0,148) mostra discrepância detectavel, porém baixa, em 9985 amostras. Já o *One-ClassSVM* treinado em exemplos funcionais apresentou ROC AUC de ~0,674, sugerindo capacidade moderada de distinguir anomalias; a versão treinada com exemplos disfuncionais atingiu ROC AUC de ~0,611, indicando separação mais fraca.

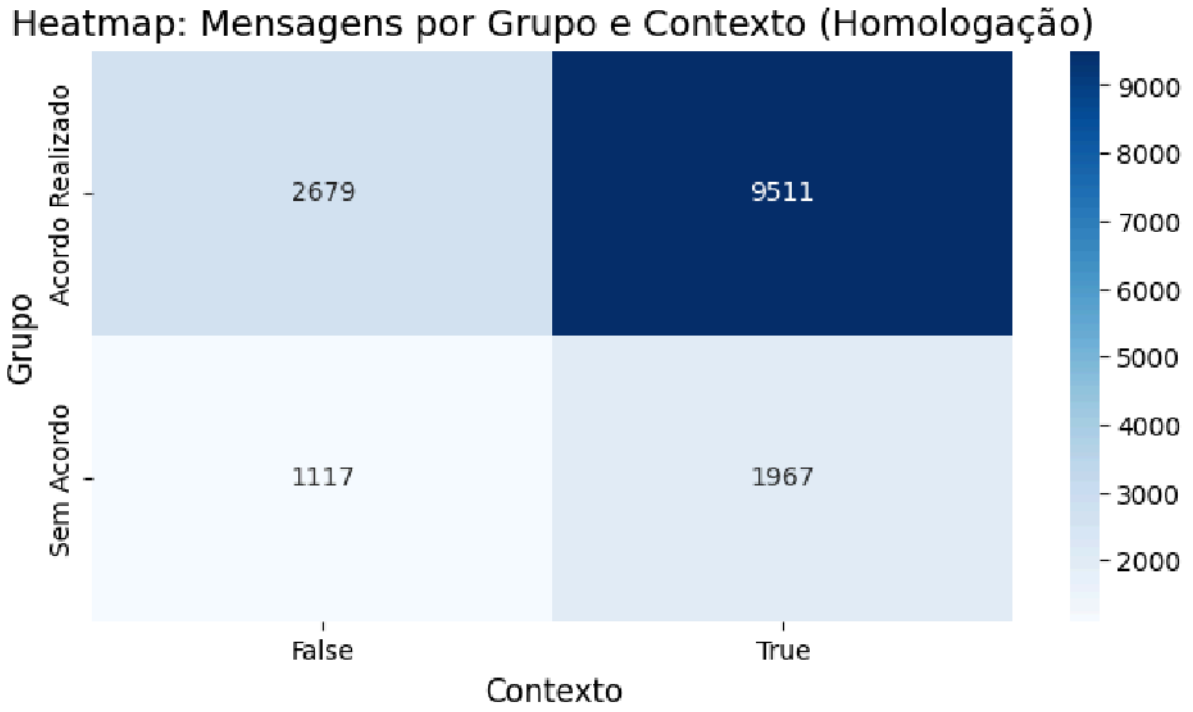
Para a homologação, foi coletada conversas de 1143 sessões únicas, sendo 978 do grupo “Acordo Realizado” e 165 “sem acordo”, após separar em mensagens, resultando no total de 15274 mensagens, 12190 do grupo “Acordo Realizado” (9511 funcionais e 2679 disfuncionais) e 3084 do grupo “Sem acordo” (1967 funcionais e 1117 disfuncionais), representados nas Figuras 10 e 11.

Figura 10 — Mapa de calor das mensagens do treinamento



Fonte: elaborado pelo autor (2025).

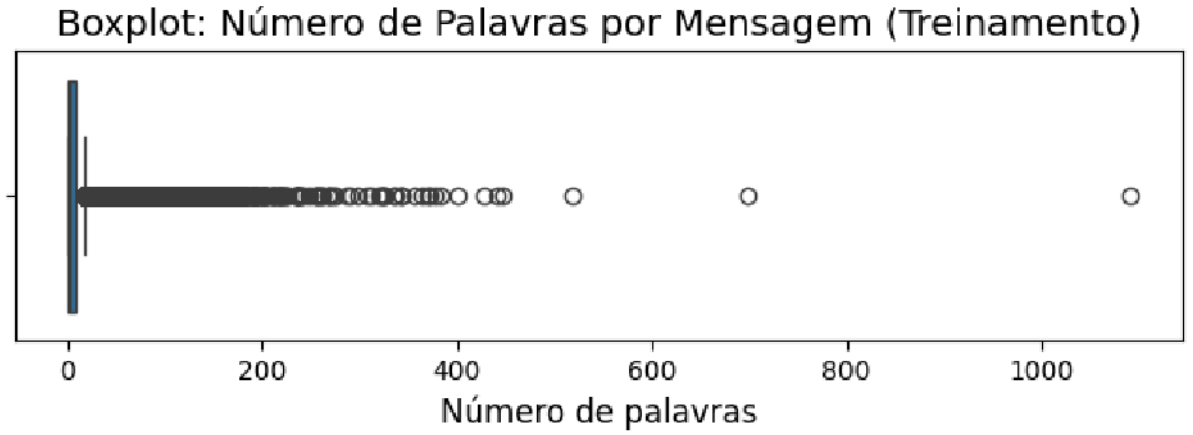
Figura 11 — Mapa de calor das mensagens de homologação



Fonte: elaborado pelo autor (2025).

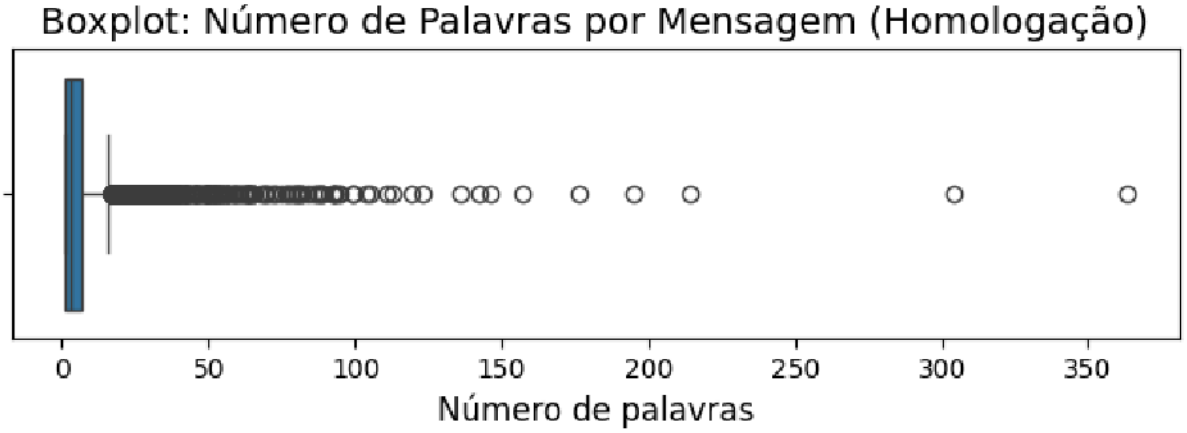
A fim de comparação, nas Figuras 12 e 13, gráficos do número de palavras por mensagem no treino e na homologação.

Figura 12 — Palavras por mensagem no treinamento



Fonte: elaborado pelo autor (2025).

Figura 13 — Palavras por mensagem na homologação



Fonte: elaborado pelo autor (2025).

Denotando a mesma tendência de acúmulo de número de palavras no limite baixo com *outliers* acima de 50 palavras por mensagem.

Após realizar os mesmos tratamentos nas mensagens de homologação, aplicado nos textos treinados, foi comparado os modelos que tiveram as melhores métricas em cada caso, visto no Quadro 3:

Quadro 3 – Modelos que tiveram melhor desempenho por métrica na homologação

Representação	Tratamento	Acurácia	F1 Macro	F1 Ponderado	Precisão
BERT	Sem tratamento	Floresta	Regressão Logarítmica	Floresta	Regressão Logarítmica
BoW	Lematizado	Floresta	Floresta	Floresta	Floresta
	Lematizado sem *stopwords*	Floresta	Floresta	Floresta	Floresta
	Stemming	Floresta	Floresta	Floresta	Floresta
	Stemming sem *stopwords*	Floresta	Floresta	Floresta	Floresta
TF-IDF	Lematizado	Floresta	Floresta	Floresta	Floresta
	Lematizado sem *stopwords*	Floresta	Floresta	Floresta	Floresta
	Stemming	Floresta	Floresta	Floresta	Floresta
	Stemming sem *stopwords*	Floresta	Floresta	Floresta	Floresta

Fonte: dados da pesquisa.

Assim como no treino, a floresta aleatória obteve as melhores pontuações em quase todos os cenários de vetorização, seguida pela regressão logística em segundo lugar e pelo Linear SVC em terceiro. Analisando as métricas das florestas em homologação, temos os resultados na Tabela 5.

Tabela 5 – Resultados das florestas homologação

Representação	Tratamento	Acurácia	F1 Macro	F1 Ponderado	Precisão
BERT	Sem tratamento	0,6467199162	0,49959486	0,6360612548	0,5007336025
BoW	Lematizado	0,835472044	0,7664125918	0,8302915457	0,786299265
	Lematizado sem *stopwords*	0,8349482781	0,7662822584	0,8299967606	0,7850856472
	Stemming	0,9207804112	0,8921467278	0,9200964045	0,8992262893
	Stemming sem *stopwords*	0,8633625769	0,8093655132	0,8603933695	0,8244517059
TF-IDF	Lematizado	0,8442451224	0,7749142201	0,8377430501	0,8033898665
	Lematizado sem *stopwords*	0,8433285321	0,7739690763	0,836942642	0,8015601483
	Stemming	0,9230718869	0,894693613	0,9221878619	0,9043096437
	Stemming sem *stopwords*	0,8719392432	0,8186077736	0,8680755772	0,8410238021

Fonte: dados da pesquisa.

Entre as representações textuais, destacaram-se o TF-IDF com *stemming* e o BoW com *stemming*, ambos mantendo as *stopwords*. Em seguida, obteve-se bom desempenho com o TF-IDF com *stemming* e remoção de *stopwords*. Já a abordagem baseada em BERT, embora tenha apresentado desempenho ligeiramente inferior às melhores combinações tradicionais durante o treinamento, sofreu uma queda significativa nos resultados ao ser avaliada nos dados de homologação.

Enquanto para os modelos não supervisionados, os resultados foram proporcionalmente piores em comparação com os supervisionados:

- Modelo: *Autoencoder*
    - Média da similaridade de cosseno (contexto=*False*): 0,9663506746292114.
    - Desvio-padrão da similaridade de cosseno (contexto=*False*):

0,036752939224243164.

- Modelo: One-ClassSVM
    - ROC AUC treinado em (contexto=*True*): 0,4887165367530426
    - ROC AUC treinado em (contexto=*False*): 0,4928501374600165

Os indicadores reportados ROC-AUC de 0,489 para a classe False e 0,493 para a *True*, mostram que o *One-ClassSVM* está pior que um chute ao acaso: em vez de separar padrões normais de anômalos, ele confunde ambos e ainda tende a errar na direção contrária. Na prática, qualquer valor abaixo de 0,50 indica que o classificador inverte a lógica de decisão, minando a utilidade do modelo na fase de homologação.

O quadro se agrava quando olhamos o *autoencoder*: a média da similaridade cosseno para mensagens rotuladas como anômalas (false) é altíssima (0,966) com desvio-padrão estreito, revelando que o modelo reconstrói esses casos quase tão bem quanto os exemplos normais. Se anomalias parecem “normais” para o *autoencoder*, não há distância suficiente para fixar um limiar de corte confiável, e o sistema deixa de sinalizar *outliers* relevantes.

Em conjunto, esses números explicitam *features* incapazes de capturar a diferença semântica entre mensagens funcionais e disfuncionais. Antes de prosseguir, é preciso revisar a representação textual e calibrar novos limiares em um conjunto de validação balanceado, caso contrário, o *pipeline* continuará sendo, pior que jogar cara ou coroa.

5 CONCLUSÃO

O presente trabalho buscou avaliar a eficácia de diferentes abordagens de aprendizado de máquina para identificar mensagens improdutivas em interações digitais de cobrança, testando modelos supervisionados e não supervisionados em diferentes cenários de representação textual. Os resultados obtidos permitiram identificar claramente os pontos fortes e limitações de cada metodologia aplicada, confirmando hipóteses teóricas e fornecendo evidências empíricas sólidas que respaldam as decisões metodológicas tomadas ao longo da pesquisa. A seguir, são detalhados os principais achados obtidos com os modelos supervisionados e não supervisionados, discutindo suas implicações práticas e teóricas no contexto estudado.

5.1 Desempenho dos Modelos de Classificação Supervisionada

Os modelos supervisionados apresentaram desempenhos distintos, com a floresta aleatória destacando-se como o classificador de melhor desempenho em praticamente todos os cenários de vetorização testados. Esse algoritmo de ensemble atingiu aproximadamente 91% de acurácia e F1 ponderado de 0,910 na melhor configuração (TF-IDF com *stemming*, mantendo *stopwords*), superando os demais modelos. Em seguida, apareceu o Bag-of-Words com *stemming* (também usando floresta aleatória), com cerca de 90% de acurácia. Esses resultados sugerem que técnicas clássicas de representação textual, aliadas a métodos de conjunto, permanecem altamente competitivas mesmo frente a modelos de linguagem avançados.

Esse achado está alinhado com a literatura, florestas tendem a melhorar a generalização em relação à modelos individuais ao combinar múltiplas árvores de decisão, reduzindo a variância do modelo conforme exposto na teoria de árvores de decisão e seus aprimoramentos (Breiman *et al*., 1984). Em contrapartida, o classificador árvore de decisão isolado apresentou métricas inferiores, resultado já esperado, dado que modelos de árvore únicos costumam sofrer com menor poder preditivo. Ainda assim, vale ressaltar o benefício da interpretabilidade, árvores de decisão são reconhecidas por sua simplicidade e interpretabilidade, fornecendo regras claras de decisão.

De fato, por se tratar de um modelo interpretável, as regras extraídas pela

árvore podem fornecer *insights* sobre quais fatores linguísticos o modelo utilizou para distinguir mensagens relevantes de irrelevantes. Essa característica facilita a validação e justificativa das decisões do classificador, conforme destacado por Breiman *et al*. (1984), ainda que venha ao custo de desempenho preditivo ligeiramente menor.

Os algoritmos lineares também tiveram atuação sólida. A regressão logística figurou consistentemente em segundo lugar em desempenho, chegando em algumas configurações a resultados próximos aos da floresta aleatória. Esse fato corrobora a noção de que mesmo modelos lineares relativamente simples podem obter desempenho robusto em tarefas de classificação de texto de alta dimensionalidade, atribuindo pesos úteis a cada atributo.

O SVM Linear, por sua vez, obteve desempenho semelhante ao da regressão logística, embora ligeiramente inferior em F1 na média. Ambos os modelos linearmente separadores (logístico e SVM) confirmaram expectativas teóricas de eficácia em dados textuais esparsos (Cortes; Vapnik, 1995), mas a regressão logística possivelmente levou vantagem em nosso caso por conseguir balancear melhor as classes minoritárias (devido ao uso de class weight balanceado e à interpretação probabilística de suas saídas).

A hierarquia de desempenho observada foi, floresta aleatória > regressão logística > SVM Linear > árvore de decisão. Essa ordenação reflete o compromisso entre complexidade do modelo e capacidade preditiva: modelos mais complexos como a floresta (que combina muitas árvores) superaram a árvore única, enquanto modelos lineares bem calibrados quase alcançaram a floresta, beneficiando-se de sua simplicidade e generalização em espaço de alta dimensionalidade.

Outro ponto de análise importante foi a comparação entre as representações textuais tradicionais e a baseada em embeddings BERT. Notou-se que os melhores desempenhos vieram de representações de frequência de termos com pré-processamentos adequados (especialmente o *stemming*), ao passo que a representação densa por BERT, embora robusta, não superou os métodos tradicionais nesse conjunto de dados.

Especificamente, o modelo com vetores BERT atingiu cerca de 87% de acurácia (F1 macro de 0,82), ficando abaixo dos 91% obtidos com TF-IDF + *stemming*. Esse resultado pode parecer contraintuitivo dado o poder semântico de modelos de linguagem; entretanto, ele encontra respaldo na literatura quando se

considera a necessidade de ajuste fino (fine-tuning) de modelos pré-treinados. Conforme os autores do BERT sugerem, sem um re-treinamento específico no domínio, os *embeddings* genéricos podem não capturar de forma ótima as nuances da tarefa.

Nesse caso, o BERT foi usado essencialmente “pré-treinado” e pode não ter se adaptado totalmente aos padrões linguísticos específicos das conversas de cobrança. Além disso, ressalta-se que técnicas clássicas com *stemming* mostraram-se extremamente eficazes, o *stemming* agregou um ganho substancial de performance, enquanto a lematização ou a remoção de *stopwords* não melhoraram os resultados e até os prejudicaram em alguns casos.

Isso indica que, para este domínio, reduzir palavras às suas raízes ajudou o classificador a agrupar variações lexicais úteis, e curiosamente certas *stopwords* carregavam informação de contexto relevante (sua remoção causou leve piora). Em concordância com essas observações empíricas, a literatura reconhece que métodos tradicionais bem ajustados podem superar arquiteturas de última geração, quando os dados de treino são limitados ou quando o modelo avançado não é especificamente otimizado para a tarefa.

Por fim, destaca-se que o desempenho consistente da floresta aleatória em praticamente todos os cenários validou o primeiro objetivo específico do trabalho, demonstrando ser possível identificar de forma eficaz interações que não agregam valor por meio de modelos de aprendizado de máquina supervisionados.

5.2 Desempenho dos Modelos de Classificação Não Supervisionados

Os resultados dos métodos não supervisionados ficaram aquém do esperado e reforçam a necessidade de revisão da representação textual antes de se adotar essas técnicas como *guardrails* autônomos.

Para o *autoencoder*, a média da similaridade cosseno para mensagens rotuladas como anômalas (contexto=*False*) foi de 0,966 praticamente idêntica à obtida para as mensagens funcionais. Isso indica que o modelo reconstruiu exemplos “fora do contexto” quase tão bem quanto os normais, impossibilitando a definição de um limiar de corte confiável para sinalizar *outliers*. Na prática, se anomalias parecem “normais” para a rede, o erro de reconstrução deixa de ser discriminativo; esse comportamento contrasta com os achados de Xu *et al*. (2017),

que relatam diferenças marcantes de reconstrução em conjuntos com ruído semântico controlado. A discrepância sugere que, no nosso domínio ou modelo utilizado para geração dos *embeddings*, os *embeddings* densos alimentados ao *autoencoder* não capturam as sutis distinções semânticas entre mensagens produtivas e improdutivas, como denotado na etapa de entendimento dos dados, mensagens funcionais e disfuncionais, mesmo sendo diferentes, possuem muitas semelhanças semanticamente.

Para o *One-ClassSVM*, a performance foi ainda mais problemática, treinado com exemplos funcionais obteve ROC AUC = 0,489 enquanto o modelo treinado com exemplos disfuncionais alcançou 0,493. Ambos abaixo do ponto de aleatoriedade (0,50). Valores inferiores a 0,50 indicam que o classificador inverteu o critério de decisão, confundindo padrões normais e anômalos. Em termos operacionais, um *guardrail* com ROC AUC < 0,5 mais atrapalha do que ajuda, pois mina a confiança na filtragem automática, isso também pode se dever ao caso de mensagens dentro e fora do contexto, mesmo sendo diferentes, possuem muitas semelhanças semanticamente.

Conjuntamente, esses números evidenciam que a feição atual das representações textuais não descreve o “desvio semântico” entre mensagens funcionais e disfuncionais, condição necessária para técnicas de detecção de anomalias. Para tornar tais abordagens competitivas, devera passar por uma revisão nos itens descritos abaixo.

- Re-engenharia das features: testar vetores TF-IDF, outras *embeddings* pré-treinadas ou *embeddings* ajustados no domínio de cobrança.
- Nova calibração de limiares em conjunto de validação estratificado: usar curva PR ou distribuição empírica do erro de reconstrução em vez da similaridade cosseno bruta, que se mostrou pouco sensível.
- Arquiteturas mais profundas ou sequenciais que enfatizem ordem temporal e dependências de longo alcance.

Enquanto não são revistas essas etapas, os resultados apontam que os modelos supervisionados permanecem a fronteira mais confiável para implementar *guardrails* na Monest Cobranças, ao passo que as abordagens não supervisionadas, nos moldes atuais, equivalem a “jogar cara ou coroa” e frequentemente perder.

REFERÊNCIAS

ACADEMY, D, S. **Deep Learning book**. [2022]. Disponível em: https://www.deeplearningbook.com.br/. Acesso em: 10 dez. 2024.

AGRESTI, Alan. **An introduction to categorical data analysis**. 2. ed. Hoboken: John Wiley & Sons, 2007.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Modern information retrieval**. New York: ACM Press, 1999.

BREIMAN, L. *et al*. **Classification and Regression Trees**. New York: Chapman and Hall, 1984.

BREIMAN, L. Bagging Predictors. **Machine Learning**, Boston: Kluwer Academic Publishers. v. 24, p. 123-140, 1996. DOI 10.1007/BF00058655. Disponível em: https://link.springer.com/article/10.1007/BF00058655. Acesso em: 15 jan. 2025.

BREIMAN, L. Random Forests. **Machine Learning**, Boston: Kluwer Academic Publishers. v. 45, p. 5-32, 2001. DOI 10.1023/A:1010933404324. Disponível em: https://link.springer.com/article/10.1023/A:1010933404324. Acesso em: 17 jan. 2025.

CAMACHO-COLLADOS, José; PILEHVAR, Mohammad Taher. On the role of text preprocessing in neural network architectures: an evaluation study on text categorization and sentiment analysis. *In*: EMNLP WORKSHOP BLACKBOXNLP. 2018, Brussels. **Proceddings [...]**. Brussels: Association for Computational Linguistics, 2018.

CHAPMAN, P. *et al*. **CRISP-DM 1.0**: Step-by-step data mining guide. SPSS inc, v. 9, n. 13, p. 1–73, 2000.

CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. **Machine Learning**, Boston: Kluwer Academic Publishers. v. 20, p. 273-297, 1995. DOI 10.1007/BF00994018. Disponível em: https://link.springer.com/article/10.1007/BF00994018. Acesso em: 15 jan. 2025.

DEVLIN, J. *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2019. Disponível em: https://arxiv.org/abs/1810.04805. Acesso em: 17 dez. 2024.

DRESCH, Aline; LACERDA, Daniel P.; ANTUNES JÚNIOR, José A. V. **Design Science Research**: método de pesquisa para avanço da ciência e tecnologia. Porto Alegre: Bookman, 2015.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000.

FRENAY, L. *et al*. **Messaging as a value driver for brazilian businesses**. [2024].

Disponível em: https://www.bcg.com/publications/2024/brazil-messaging-as-a-value-driver-for-brazilian-businesses. Acesso em: 7 dez. 2024.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo, Brasil: Atlas, 2002.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.

GUANAES, Lucas. O discreto monopólio do meta: um estudo de caso sobre o domínio do Whatsapp no Brasil. *in*: SEMINÁRIO DO LEG,14. 2024, Limeira. **Anais** [...]. Limeira: UNICAMP, 2024.

HAM, L. A gentle introduction to Vector Search. 2022. Disponível em: https://opendatascience.com/a-gentle-introduction-to-vector-search/. Acesso em: 15 dez. 2024.

HINTON, Geoffrey E.; SALAKHUTDINOV, Ruslan R. Reducing the dimensionality of data with neural networks. **Science**, Washington, v. 313, n. 5786, p. 504-507, 2006. DOI 10.1126/science.1127647. Disponível em: https://www.cs.toronto.edu/~hinton/absps/science.pdf. Acesso em 15 jan. 2025.

HO, T. K. Random Decision Forests. *In:* INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION, 3., 1995, Montreal. **Proceddings [...]**. Montreal: IEEE. 2002. p. 278-282.

HOSMER, David W.; LEMESHOW, Stanley. **Applied Logistic Regression**. 2. ed. New York: Wiley-Interscience Publication, 2000.

JENSEN, K. A. **CRISP-DM**: Cross-Industry Standard Process for Data Mining. [2016]. https://sites.google.com/view/kajensen/profile/project-methodology. Acesso em: 15 dez. 2024.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**: an introduction to Speech Recognition, Computational Linguistics and Natural Language Processing. New Jersey: Prentice Hall, 2008.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**: an introduction to Natural Language Processing, Computational Linguistics and Speech Recognition with Language Models. 3. ed. Stanford: Stanford University, 2025. *E-book*.

LIMNA, P. *et al*. The use of ChatGPT in the digital era: perspectives on chatbot implementation. **Journal of Applied Learning and Teaching**, v. 6, n. 1, p. 64–74, 2023.

MANNING, C. D.; SCHÜTZE, H. **Foundations of Statistical Natural Language Processing**. Cambridge: MIT Press, 1999.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich.

Introduction to Information Retrieval. Cambridge: Cambridge University Press, 2008.

MARCONI, M. d. A.; LAKATOS, E. M. **Metodologia científica**. 7. ed. São Paulo: Atlas, 2017.

MENARD, Scott W. **Applied Logistic Regression Analysis**. 2. ed. Thousand Oaks: Sage Publications, 1995.

MIKOLOV, Tomas *et al*. Distributed representations of words and phrases and their compositionality. **arXiv preprint arXiv:1706.03762**. 2013. Disponível em: https://arxiv.org/abs/1310.4546. Acesso em 17 jan. 2025.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **CVE-2025-2998**. 2025. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2025-2998. Acesso em: 01 abr. 2025.

NG, Andrew. Y. Feature selection, L1 vs. L2 regularization, and rotational invariance. *In*: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 21., 2004, Banff. **Proceddings** [...]. New York: Association for Computing Machinery. 2004.

OPENAI. **ChatGPT:** Optimizing Language Models for dialogue. [2022]. Disponível em: https://openai.com/blog/chatgpt. Acesso em: 11 out. 2023.

PALOMINO, Marco A.; AIDER, Farida. Evaluating the effectiveness of Text Pre-Processing in Sentiment Analysis. **Applied Sciences**, Basel, v. 12, n. 17, p. 1-21, 2022. DOI 10.3390/app12178765. Disponível em: https://www.mdpi.com/2076-3417/12/17/8765. Acesso em: 15 jan. 2025.

SAKURADA, Mayu; YAIRI, Takehisa. Anomaly detection using autoencoders with nonlinear dimensionality reduction. *In*: WORKSHOP ON MACHINE LEARNING FOR SENSORY DATA ANALYSIS (MLSDA). 2014, New York. **Proceddings** [...]. New York: ACM. 2014.

SCHÖLKOPF, B. *et al*. Estimating the support of a high-dimensional distribution. **Neural Computation**, MIT Press, v. 13, n. 7, p. 1443–1471, 2001.

SCHWABER, K.; SUTHERLAND, J. **The Scrum guide**: the definitive guide to Scrum: the rules of the game. [2020]. Disponível em: https://scrumguides.org/scrum-guide.html. Acesso em: 19 dez. 2024.

UYSAL, Alper Kürşat; GUNAL, Serkan. The impact of preprocessing on text classification. **Information Processing & Management**, Amsterdam, v. 50, n. 1, p. 104–112, 2014. DOI 10.1016/j.ipm.2013.08.006. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0306457313000964?via%3Dihub. Acesso em 15 jan. 2025.

VASWANI, A. *et al*. Attention is all you need. **arXiv preprint arXiv:1706.03762**. 2023. Disponível em: https://arxiv.org/abs/1706.03762. Acesso em: 18 nov. 2024.

XU, Weidi *et al*. Variational autoencoder for semi-supervised text classification. **arXiv preprint arXiv:1603.02514**. 2016. Disponível em: https://arxiv.org/abs/1603.02514. Acesso em 19 nov 2024.