UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Detecção de cadastro telefônico desatualizado por meio de respostas de SMS transacionais

David Sobrinho Camurça

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

David Sobrinho Camurça

Detecção de cadastro telefônico desatualizado por meio de respostas de SMS transacionais

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Dr. Rafael Geraldeli Rossi

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

Camurça, David Sobrinho
S856m Detecção de cadastro telefônico desatualizado por meio de respostas de SMS transacionais / David Sobrinho Camurça ; orientador Rafael. – São Carlos, 2023.
54 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. LaTeX. 2. abnTeX. 3. Classe USPSC. 4. Editoração de texto. 5. Normalização da documentação. 6. Tese. 7. Dissertação. 8. Documentos (elaboração). 9. Documentos eletrônicos. I. Rossi, Rafael Geraldeli, orient. II. Título.

David Sobrinho Camurça

Detection of outdated phone records through transactional SMS responses

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Dr. Rafael Geraldeli Rossi

Original version

São Carlos

2023

Este trabalho é dedicado em memória ao meu pai, que foi chamado por Deus enquanto eu finalizava os últimos detalhes deste trabalho. Sem ele, jamais teria chegado tão longe. Obrigado por sempre acreditar em mim, por ser um exemplo de trabalhador e por ensinar que nada vem de graça; tudo tem seu preço. Obrigado por investir em mim e dizer: "Vá, meu filho, eu dou um jeito."

RESUMO

Camurça, D. S. **Detecção de cadastro telefônico desatualizado por meio de respostas de SMS transacionais**. 2023. 54p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Este trabalho investiga a aplicação de técnicas de classificação de texto para otimizar a comunicação via SMS entre empresas e clientes, com o objetivo de identificar necessidades de atualização cadastral. Considerando a relevância do SMS em transações empresariais e a importância da manutenção de cadastros atualizados para a conformidade com a LGPD, diferentes modelos, incluindo SVM e BERT, foram explorados para analisar mensagens em linguagem natural. O BERT demonstrou superioridade em todas as métricas avaliadas, estabelecendo-se como referência para tarefas similares. Contudo, aspectos como custo e personalização são fundamentais na seleção do modelo, sendo o Random Forest uma alternativa viável. Este estudo proporciona *insights* significativos para a implementação de soluções eficientes de classificação de texto em SMS, representando um recurso valioso para empresas e pesquisadores.

Palavras-chave: SMS. Classificação de Texto. BERT. Atualização Cadastral. LGPD.

ABSTRACT

Camurça, D. S. **Detection of outdated phone records through transactional SMS responses**. 2023. 54p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

This study investigates the application of text classification techniques to optimize SMS communication between companies and clients, aiming to identify needs for registration updates. Given the significance of SMS in business transactions and the importance of maintaining updated registers for compliance with LGPD, different models, including SVM and BERT, were explored to analyze messages in natural language. BERT demonstrated superiority in all evaluated metrics, establishing itself as a reference for similar tasks. However, aspects such as cost and customization are crucial in model selection, with Random Forest being a viable alternative. This study provides significant insights for implementing efficient text classification solutions in SMS, serving as a valuable resource for companies and researchers.

Keywords: SMS. Text Classification. BERT. Registration Update. LGPD.

LISTA DE FIGURAS

Figura 1 – Pipeline de Classificação de texto. Fonte: Baseado em (NASEEM *et al.*, 2021) 21
Figura 2 – Arquitetura Word2Vec, destacando A (CBOW) e B (Skip-gram). Fonte: (JANG; KIM; KIM, 2019) 25
Figura 3 – Random Forest Voting. Fonte: (YEHOSHUA, 2023) 28
Figura 4 – Logistic Regression. Fonte: (KANADE, 2022) 29
Figura 5 – Processo de pré-treino e *fine-tuning* BERT. Fonte: (DEVLIN *et al.*, 2018) 29
Figura 6 – Esquema de validação. Fonte: (ZHENG, 2015) 31
Figura 7 – Curva ROC. Fonte: (HOO; CANDLISH; TEARE, 2017) 32
Figura 8 – Adoção de IA nas empresas. Fonte: (MCKINSEY, 2022) 33
Figura 9 – Aumento de recursos financeiros para investimento em IA, Fonte: (MCKINSEY, 2022) 34
Figura 10 – Canvas da solução. Fonte: Autor 37
Figura 11 – Swot da solução. Fonte: Autor 41

LISTA DE TABELAS

Tabela 1 – Exemplos de texto desestruturado de baixa qualidade 22
Tabela 2 – BoW - Frequência de palavras nos documentos 25
Tabela 3 – Comparação das Plataformas de Classificação de Textos 38
Tabela 4 – Comparação modelos de classificação de texto para a classe 1 (cadastro desatualizado). 47

SUMÁRIO

1 INTRODUÇÃO 19
1.1 Contextualização 19
1.2 Inovação e Proposta de Solução do Problema 20

2 FUNDAMENTAÇÃO TEÓRICA 21
2.1 Coleta de Dados 21
2.2 Representação de Texto 22
2.2.1 Pré-processamento de Texto 23
2.2.2 Extração de Características 24
2.3 Classificação 27
2.3.1 *Support Vector Machine (SVM)* 27
2.3.2 *Multinomial Naive Bayes (MNB)* 27
2.3.3 *Random Forest* 28
2.3.4 *Logistic Regression* 28
2.3.5 BERT, *fine-tuning* 29
2.4 Avaliação da Performance de Classificação 30

3 PANORAMA DE MERCADO, INVESTIMENTO E RESULTADOS ESPERADOS 33
3.1 Panorama de Mercado e Investimento em Inteligência Artificial . . 33
3.2 Resultados Esperados 35

4 CANVAS, BENCHMARK TECH, SWOT 37
4.1 Canvas 37
4.2 *Benchmark* Tecnológico 38
4.3 SWOT 41

5 MÉTODO DE PESQUISA - CONSTRUÇÃO DO MVP 43
5.1 Coleta e Rotulação de Textos 43
5.2 Pré-processamento de Texto 43
5.3 Balanceamento das Classes 43
5.4 Representação de Texto 44
5.5 Classificação 45
5.6 Avaliação 45

6 DISCUSSÃO E ANÁLISE DE RESULTADOS 47

7 CONCLUSÕES 49

Referências 51

1 INTRODUÇÃO

1.1 Contextualização

O *Short Message Service* (SMS), é uma das formas mais antigas e populares de comunicação móvel, permitindo o envio de mensagens curtas de texto entre dispositivos móveis (BROWN; SHIPMAN; VETTER, 2008). Apesar de terem sido substituídos em grande parte pelos aplicativos de mensagens instantâneas, como o WhatsApp, Telegram e o Messenger, muitas empresas ainda utilizam o SMS como uma forma de comunicação com seus clientes, como confirmações de transações, lembretes, cobranças e envio de códigos de segurança (MKOM, 2022). Além disso, segundo Shakeel, Karim and Khan (2019), o SMS é um canal amplamente utilizado para a coleta de *feedback* dos consumidores para mensuração da qualidade dos serviços oferecidos.

Entretanto, em algumas aplicações, o uso do SMS como canal para comunicação e coleta de informações pode apresentar desafios para as empresas, principalmente quando as respostas dos clientes correspondem a textos escritos em linguagem natural. Um exemplo disso ocorre quando o cliente recebe uma mensagem da empresa, como uma cobrança, e responde à mensagem informando que não é o devedor da cobrança, evidenciando uma situação de cadastro desatualizado (THINKDATA, 2022). Para evitar problemas como esse, é crucial que as empresas sejam diligentes na manutenção dos cadastros de seus clientes, seja por meio da atualização frequente dos dados cadastrais ou através das respostas aos SMS para identificar a necessidade de atualização do cadastro. No entanto, o grande volume de mensagens de texto recebidas e a natureza não estruturada dos dados textuais tornam o processo de classificação e processamento dessas mensagens um desafio adicional para as empresas (ABAYOMI-ALLI *et al.*, 2019).

Existem várias ferramentas disponíveis para classificar textos em geral, Google Cloud NLP, Amazon Comprehend, IBM Watson NLP, Microsoft Azure Text Analytics, NLTK, Hugging Face Transformers e ChatGPT (OpenAI). Cada uma dessas soluções apresenta benefícios e desafios. As plataformas de grandes empresas como Google, Amazon, IBM e Microsoft oferecem recursos avançados e modelos robustos, porém podem ser caras para grandes volumes de dados e, por vezes, menos personalizáveis ou menos adequadas para problemas muito específicos, como a detecção de cadastros telefônicos desatualizados. Por outro lado, bibliotecas como NLTK e Hugging Face Transformers oferecem maior flexibilidade, com a NLTK requerendo maior trabalho de desenvolvimento e a Hugging Face podendo ser excessiva para problemas mais simples ou ainda, requerendo uma grande quantidade de exemplos de treinamento e poder computacional para fazer o ajuste fino (*fine-tuning*) dos modelos. O ChatGPT (OpenAI) pode ser útil para a classificação de respostas de SMS, embora seja menos personalizável e possivelmente mais caro, dependendo

do volume de uso e do modelo sendo utilizado.

É válido ressaltar que muitas dessas soluções pré-prontas e modelos pré-treinados são otimizados para textos em inglês, o que pode resultar em desempenho insatisfatório quando aplicados a textos em português. Esse viés linguístico pode ser particularmente problemático para empresas e aplicações que operam em países de língua portuguesa, pois a acurácia na classificação de textos e a eficácia na detecção de cadastros desatualizados podem ser comprometidas.

1.2 Inovação e Proposta de Solução do Problema

Com a crescente popularidade das mensagens de texto em comunicações transacionais entre empresas e clientes, a classificação manual das mensagens dos clientes pode ser um processo demorado e custoso e propenso a erros. Nesse contexto, a aplicação de técnicas de classificação de textos pode ser uma solução eficiente, permitindo a automação da classificação e análise de grandes volumes de mensagens de texto (AGGARWAL, 2018). É importante destacar que as soluções oferecidas pelas empresas atuais podem enfrentar desafios ao lidar com situações específicas que envolvem a classificação de mensagens de texto enviadas por SMS, como limitações linguísticas, custos, poder computacional e customização limitada.

Neste trabalho de conslusão de curso, será instanciando um processo para classificação automática de textos cujo objetivo é indentificar as melhores técnias em cada etapa do processo para obter um modelo de detecção de cadastros telefônicos desatualizados por meio de respostas de SMS transacionais escritas em português. Adicionalmente, a solução proposta fará uso de bibliotecas gratuitas disponíveis na internet, tornando o projeto não apenas inovador, mas também acessível para empresas que buscam melhorar a qualidade da comunicação com seus clientes sem incorrer em altos custos.

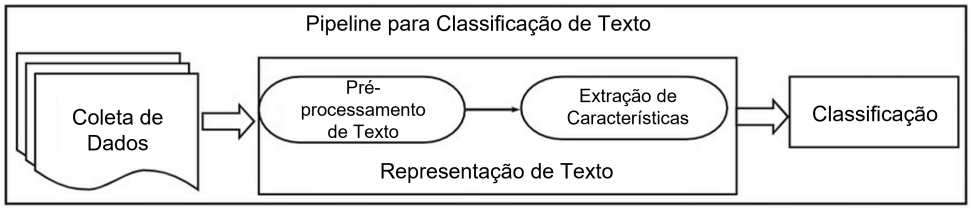
Além disso, manter o cadastro de clientes atualizado é fundamental para garantir a conformidade com a Lei Geral de Proteção de Dados (LGPD), que entrou em vigor em setembro de 2020. Essa lei tem como objetivo proteger a privacidade e os dados pessoais dos cidadãos brasileiros, e impõe às empresas diversas obrigações em relação ao tratamento desses dados, de acordo com o Art. 1° da Lei n° 13.709, de 14 de agosto de 2018 (Brasil, 2018). Portanto, a detecção de cadastros desatualizados por meio de respostas de SMS transacionais pode ser uma ferramenta importante para ajudar as empresas a manter seus registros atualizados e em conformidade com as normas de proteção de dados.

2 FUNDAMENTAÇÃO TEÓRICA

A classificação automática de textos é uma área de estudo que tem como objetivo automatizar a tarefa de atribuir uma ou mais categorias a um texto, com base no seu conteúdo (AGGARWAL, 2018). Segundo Jurafsky and Martin (2009), a classificação automática de texto pode ser definida como um processo que consiste em aprender uma função de mapeamento que associa um vetor de características extraído do texto a uma ou mais classes pré-definidas.

Na Figura 1 é apresentado um pipeline para a classificação de textos, isto é, para construir um modelo de classificação capaz de atribuir automaticamente categorias à textos não estruturados (NASEEM *et al.*, 2021). Nas próximas seções serão apresentadas as fundamentações teóricas a respeito de cada etapa do pipeline, sendo elas: i) *Unstructured, Low Quality Text*, ii) *Text Representation*, iii) *Classification* e iv) *Classification Performance Evaluation* começando pela coleta de dados até as métricas de validação do modelo.

Figura 1 – Pipeline de Classificação de texto. Fonte: Baseado em (NASEEM *et al.*, 2021)



2.1 Coleta de Dados

Antes do pré-processamento, da extração de características, e da classificação em qualquer fluxo de trabalho de análise de texto, a primeira fase envolve a coleta ou seleção de dados existentes. Na maioria das aplicações de classificação de texto, os dados coletados ou selecionados inicialmente são frequentemente não estruturados e em algumas situações, de baixa qualidade. Isso significa que os textos podem ser oriundo de várias fontes e terem diferentes formatos e conteúdos, como publicações em mídias sociais, resenhas de produtos, e-mails, SMSs e documentos eletrônicos em geral.

Como mencionado anteriormente, dependendo da aplicação, textos podem ter ruídos, como erros ortográficos e carácteres que não compõem a gramática da escrita. Para ilustrar, na Tabela 1, são apresentados 10 exemplos de respostas de SMS de baixa qualidade provenientes de diversos serviços de uma empresa varejista. Estes exemplos representam a natureza inicialmente não estruturada e ruidosa dos dados coletados na primeira fase de um fluxo de trabalho de análise de texto. Os textos mostram uma variedade de características

comuns em comunicações informais, como jargão, abreviações, emojis, erros de digitação e uso de caracteres especiais.

Tabela 1 – Exemplos de texto desestruturado de baixa qualidade

ID	Exemplo de texto
1	“que incomodo, nao sou maria ^^”
2	"1234567890"
3	"nota 10 d+!!! #$%^^&*"
4	"decepcionado com produt. n corresponde o anunciado ###"
5	"nova att do app melhorou mt a interface! :) :))))))) (((("
6	"pessimo atendiment... muito lento! @@@@"
7	"parabens para o montador nota 10"
8	"!@ \^&()_+ {}|[]:’<>?,./;"
9	"bemol, pessoa errada, aqui n tem nenhum leandro ^^ %%%"
10	"já falei mil vezes q não sou essa pessoa :(###### 123456"

Esta falta de estrutura e presença de ruído pode dificultar a interpretação precisa das respostas do cliente e representam desafios significativos para a extração de características úteis e a subsequente classificação. No entanto, essa fase inicial de coleta ou seleção de dados é fundamental para definir a qualidade das informações, destacando a importância de estratégias eficientes de coleta e pré-processamento de dados para criar um corpus de texto relevante e representativo.

Para muitas aplicações, é crucial ter rótulos claros nos documentos. Quando esses rótulos não existem, métodos de rotulação humana ou ferramentas de inteligência artificial são necessários para adicionar rótulos aos textos (WANG *et al.*, 2021). A rotulação correta é importante para evitar erros e garantir que os modelos sejam justos e confiáveis (DING *et al.*, 2022). Assim, encontrar formas eficazes e exatas de rotulação faz se necessário para desenvolver sistemas de processamento de linguagem natural e garantir um corpus de texto de qualidade.

2.2 Representação de Texto

Nessa etapa, são aplicadas técnicas para remover caracteres especiais, pontuação e normalizar o texto, garantindo que ele esteja em um formato padronizado e adequado para análise. Em seguida, na etapa de representação o texto é transformado em uma forma estruturada e representável, como vetores numéricos. Essa representação numérica captura as características e relações das palavras, permitindo que os algoritmos de aprendizado

de máquina trabalhem com os dados textuais de forma eficiente. Nesse formato de representação, cada dimensão do vetor corresponde à uma característica do texto e o valor da dimensão é uma mensuração da ocorrência de determinada característica. Vale ressaltar que a representação vetorial é o formato de entrada para a maioria dos algoritmos de aprendizado de máquina tradicionais e estado-da-arte.

2.2.1 Pré-processamento de Texto

O pré-processamento de textos é uma fase que envolve processos como remoção de caracteres especiais, pontuação, normalização de texto, tokenização, remoção de *stopwords*, e simplificação de palavas como a radicalização (*stemming*, em inglês) e lematização (HACOHEN-KERNER; MILLER; YIGAL, 2020). Cada uma dessas etapas desempenha um papel fundamental na transformação de dados brutos em um formato mais adequado para análise, facilitando a extração de *insight* úteis e aprimorando a eficácia dos modelos de aprendizado de máquina.

A remoção de caracteres especiais e pontuação consiste em retirar caracteres como *hashtags*, símbolos de moedas, sinais de pontuação e emojis, pois esses podem não contribuir para o significado do texto, portanto, ser removidos para simplificar a análise. Por exemplo, em uma frase como "Ganhei dinheiro$!!", a frase "Ganhei dinheiro"seria mantido e os caracteres "$!!"seriam removidos. A remoção de caracteres especiais e pontuação permite que o algoritmo se concentre no conteúdo essencial do texto, ou seja, as palavras. Todavia, vale ressaltar que para algumas tarefas, como a análise de sentimentos em redes sociais, pode ser que caracteres de pontuação sejam mantidos, já que formam emojis, os quais podem auxiliar na definição do sentimento de um texto (SHIHA; AYVAZ, 2017)

A normalização de texto consiste em padronizar o texto de entrada, o que geralmente inclui converter todas as letras para minúsculas. Por exemplo, sem a normalização, palavras como "CASA", "Casa"e "casa"seriam tratadas como entidades distintas pelo algoritmo de aprendizado de máquina, o que pode distorcer a análise e a extração de padrões. A normalização garante que essas variantes sejam tratadas como a mesma palavra, e que sejam representadas pelo mesmo atributo.

A tokenização é o processo de dividir um fluxo de dados textuais em palavras, termos, frases, símbolos ou outros elementos significativos chamados tokens. Geralmente é feita uma quebra pelo caractere de espaço para separar os tokens, o que basicamente faz com que palavras simples sejam consideradas como tokens. Porém, há outras formas de tokenizar o texto, como considerando sequencias de caracteres, sequencias de palavras, ou ainda identificar emojis e outros caracteres unidos com palavras e separá-los(MULLEN *et al.*, 2018).

As *stopwords* são palavras comuns que geralmente não contribuem para o significado de uma frase, como "a", "e", "o", "em", "de", entre outras (KHANNA, 2021). Por exemplo,

na frase "O gato está na caixa", as palavras "o", "está", "na"poderiam ser consideradas *stopwords* e serem removidas, resultando na frase "gato caixa". Embora a remoção de *stopwords* possa ser útil em muitos contextos, ela não é sempre necessária. A decisão de remover ou não as *stopwords* depende da tarefa específica que está sendo realizada e do objetivo que se deseja alcançar.

Por fim, pode-se utilizar a simplificação dos termos na etapa de pré-processamento. A simplificação de termo tem como objetivo normalizar palavras com significados semelhantes que possam variar devido a diferentes tempos verbais, gêneros ou números, assegurando que todas representem uma única informação. Estratégias como a radicalização e a lematização podem ser empregadas para alcançar essa uniformidade (CONRADO, 2009).

A radicalização (*stemming*) é um processo que remove os sufixos das palavras, reduzindo-as à sua raiz ou forma base. Por exemplo, “correndo”, “corre” e “corredor” poderiam ser reduzidas ao radical “corr”. Isso pode ajudar a reduzir a complexidade do texto e a consolidar variantes de uma mesma palavra. Porém, vale ressaltar que em algumas situações, palavras com diferentes significados podem ser representados pelo mesmo radical.

De acordo com Balakrishnan and Lloyd-Yemoh (2014) a lematização é um processo que reduz as palavras à sua forma base ou “lemma”, considerando o contexto. Diferente da radicalização, que apenas remove os sufixos das palavras, a lematização considera a morfologia da palavra e transforma verbos para sua forma no infinitivo, enquanto substantivos e adjetivos são convertidos para o masculino singular. Por exemplo, "correndo"se tornaria "correr", e "melhores"poderia ser reduzido a "bom". Essa etapa ajuda a consolidar diferentes formas da mesma palavra.

2.2.2 Extração de Características

Após a conclusão da etapa de pré-processamento, inicia-se a fase de extração de características, onde os dados textuais são transformados em vetores numéricos que podem ser utilizados por algoritmos de aprendizado de máquina. Esta etapa é vital porque os modelos de aprendizado de máquina dependem dessas características numéricas para realizar suas tarefas, seja classificação, agrupamento, regressão, entre outros (AGGARWAL, 2018).

Existem algumas técnicas para a extração de características, e as duas grandes categorias dessas técnicas são a abordagem de “Saco de Palavras” (*Bag of Words - BoW*, em inglês) e a abordagem de *Embeddings*.

A Tabela 2 é uma ilustração da representação por BoW. Neste modelo, cada palavra do documento é transformada em um atributo em um vetor de características multidimensional e esparso. Os valores de cada atributo ou dimensão são determinados pela

frequência do atributo que a dimensão representa. Esses valores podem ser diretamente a frequência do atributo no documento (*term-frequency – TF*) ou uma versão que ponderada a frequência do atributo pelo número documentos em que o atributo aparece (*term-frequency-inverse document frequency – TF-IDF*) (JURAFSKY; MARTIN, 2009).

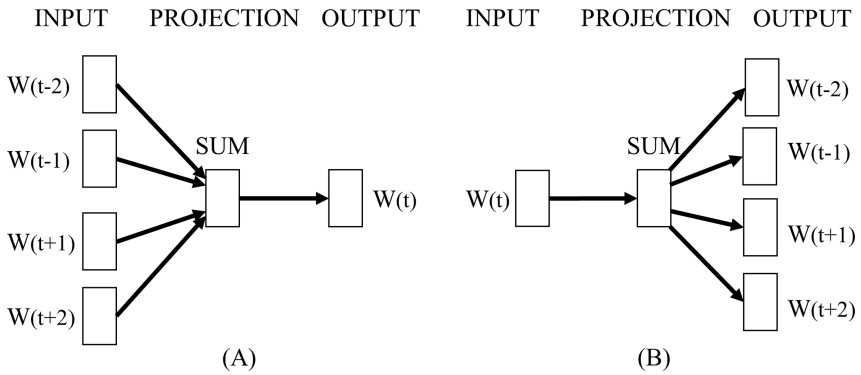
Tabela 2 – BoW - Frequência de palavras nos documentos

documento	**por**	**favor**	**não**	**conheço**	**essa**	**pessoa**	**número**	**errado**	**casa**
doc1	1	1	1	1	1	1	0	0	0
doc2	0	0	1	1	0	0	1	1	0
doc3	0	0	0	0	0	1	0	1	0
doc4	1	0	1	0	0	0	0	0	1
doc5	0	1	0	0	1	0	0	0	1
doc6	1	1	0	0	0	0	1	0	0

Por outro lado, as representações vetoriais contínuas, também conhecidas como *embeddings*, são cruciais para capturar a semântica e as relações intrínsecas entre palavras (MIKOLOV *et al.*, 2013; PENNINGTON; SOCHER; MANNING, 2014). A Figura 2 é um representação da arquitetura da técninca *Word2Vec* onde a mesma pode ser CBOW (*Continuous Bag of Words*) ou *Skip-gram* para gerar esses *embeddings*.

Ambas geram representações a partir do treinamento de uma rede neural. No caso da CBOW, dada uma palavra t, e seu contexto, isso é, palavras que precedem ou sucedem t, o objetivo é utilizar o contexto de t para prever t. Já na abordagem *Skip-gram*, o treinamento consiste em usar a palavra t como entrada e prever o contexto de t (JANG; KIM; KIM, 2019). Após obter os *embeddings* das palavras, é essencial construir uma representação global do documento. Isso é frequentemente alcançado calculando-se a média ou a soma dos *embeddings* de todas as palavras contidas no documento, proporcionando uma representação consolidada do seu conteúdo semântico (PITA; PAPPA, 2018).

Figura 2 – Arquitetura Word2Vec, destacando A (CBOW) e B (Skip-gram). Fonte: (JANG; KIM; KIM, 2019)



Pode-se também utilizar a técnica Doc2Vec, a qual é baseada no Word2Vec para gerar diretamente as *embeddings* de sentenças, parágrafos ou textos completos. A principal diferença das abordagens reside em também aprender *embeddings* que representam os documentos durante o treinamento além das *embeddings* das palavras (LE; MIKOLOV, 2014).

Outra estratégia é recorrer aos Large Language Models (LLMs) (ZHAO *et al.*, 2023), que são modelos de linguagem compostos por redes neurais com muitos parâmetros, treinados em grandes quantidades de texto sem rótulo. Exemplos de LLMs incluem o *Embedding from Language Models* (ELMo), o *Bidirectional Encoder Representations from Transformers* (BERT), o *Generative Pretrained Transformer* (GPT), e o *Pathways Language Model* (PALM). Os LLMs geralmente utilizam redes neurais profundas para aprender as representações, e comumente fazer uso de um tipo de componente denominado *Transformer* (WANG; LI; SMOLA, 2019), o qual é capaz de aprender relações entre palavras e determinar a importância das relações.

O BERT representa uma significativa evolução na modelagem de linguagem natural, construído sobre a arquitetura *Transformer*. Este modelo emprega apenas o componente do codificador da arquitetura *Transformer* e é treinado usando um método conhecido como *Masked Language Model* (MLM). No treinamento MLM, algumas palavras da sentença são ocultadas e o modelo é treinado para predizê-las, utilizando o contexto fornecido pelas palavras restantes. Isso possibilita que o BERT capture contextos de palavras de forma bidirecional, entendendo o relacionamento de uma palavra com todas as outras na sentença, anteriores e subsequentes. Dessa maneira, o BERT é capaz de gerar representações de texto altamente contextualizadas, sendo extremamente eficaz em diversas tarefas de processamento de linguagem natural.

Apesar de outros LLMs maiores e mais recentes ao BERT terem sido lançados nos últimos anos, o BERT continua sendo o estado da arte em algumas aplicações de processamento de linguagem natural e ainda tem sido utilizado em muitas aplicações (LIAO *et al.*, 2023). Muito disso se dá ao fato não só da arquitetura do BERT mas pelo fato de outros modelos terem um número de parâmetros muito maior, o que requer um poder computacional proporcionalmente maior, maior tempo de treinamento, e também um maior tempo de resposta, o que pode inviabilizar o uso de LLMs em algumas aplicações.

Em resumo, as técnicas de extração de características, como BoW e *embeddings*, fornecem métodos robustos para transformar texto em representações numéricas que são úteis para algoritmos de aprendizado de máquina. No entanto, modelos mais avançados como o BERT, que é construído sobre a arquitetura *Transformer*, oferecem representações contextualizadas e bidirecionais do texto. As representações apresentadas aqui servem como entrada para algoritmos de aprendizado de máquina que serão apresentados na próxima seção.

2.3 Classificação

Uma vez gerada a representação da coleção de textos, é possível aplicar algoritmos de aprendizado de máquina para gerar modelos de classificação, ou seja, algoritmos que irão aprender um mapeamento das características dos textos para um conjunto de classes pré-definidas (AGGARWAL, 2018). Em outras palavras, o algoritmo de aprendizado de máquina aprende a identificar padrões e relações nas características extraídas do texto que permitem classificá-lo em uma ou mais classes. Existem vários algoritmos de aprendizado de máquina que podem ser usados para classificação de texto, como Árvores de Decisão, Redes Neurais, Naïve Bayes e SVM (Support Vector Machines) (TAN *et al.*, 2019).

2.3.1 *Support Vector Machine (SVM)*

O SVM busca encontrar o que é conhecido como um "hiperplano de máxima margem"no espaço de características. A fórmula que representa essa operação é:

$$f(x) = \mathbf{w} \cdot \phi(\mathbf{x}) + b \tag{2.1}$$

no qual $\phi(\mathbf{x})$ é uma função de mapeamento para o espaço de características de alta dimensão e $\mathbf{w}$ e b são os parâmetros do modelo. Esta fórmula é central para entender como o SVM maximiza a margem entre as classes.

2.3.2 *Multinomial Naive Bayes (MNB)*

O MNB é uma extensão do algoritmo Naive Bayes, sendo uma escolha popular para classificação de texto, onde os documentos são representados pela frequência das palavras ou termos que contêm (MCCALLUM; NIGAM *et al.*, 1998). Este modelo, que assume independência condicional entre as características dado o rótulo da classe, é robusto e eficaz em contextos onde os atributos são predominantemente categóricos, como palavras ou n-gramas em textos. No MNB, a probabilidade condicional de um termo t_k em um documento, dado um rótulo de classe c_j, é central para a classificação. Esta probabilidade é calculada pela fórmula:

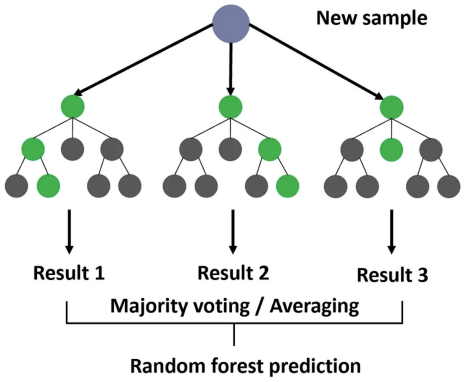
$$P(t_k|c_j) = \frac{1 + \sum_{d \in D} w_{d,t_k}}{|V| + \sum_{t \in V} \sum_{d \in D} w_{d,t}} \tag{2.2}$$

Onde, t_k representa um termo específico e c_j uma classe específica. D é o conjunto de todos os documentos na classe c_j, enquanto w_{d,t_k} denota o peso ou a frequência do termo t_k no documento d. V é o vocabulário, compreendendo todos os termos únicos em todos os documentos. O termo de suavização, 1, adicionado ao numerador, previne a atribuição de probabilidade zero a termos que não aparecem em documentos de uma classe específica, garantindo a robustez do modelo em diferentes conjuntos de dados.

2.3.3 *Random Forest*

Random Forest cria várias árvores de decisão (floresta) durante o treinamento e faz a classificação com base na votação de todas as árvores. Em vez de usar uma fórmula matemática para representar este processo, é mais intuitivo entender o Random Forest através de uma abordagem conceitual ou visual, conforme ilustrado na Figura 3. Cada árvore no floresta faz uma previsão independente e a classe que recebe a maioria dos votos é escolhida como a previsão final. Este método permite que o modelo capture interações complexas entre características e seja menos propenso a *overfitting*.

Figura 3 – Random Forest Voting. Fonte: (YEHOSHUA, 2023)



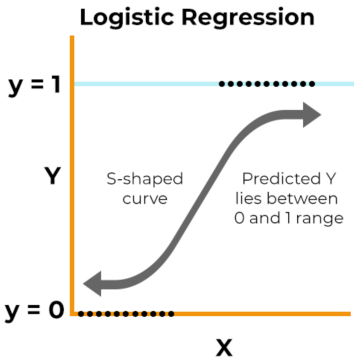
2.3.4 *Logistic Regression*

O *Logistic Regression* é um algoritmo de aprendizado supervisionado amplamente utilizado para tarefas de classificação binária. O algoritmo modela a probabilidade de uma determinada amostra pertencer à classe positiva, que é denotada como $y = 1$ ou negativa $y = 0$. O modelo utiliza a função logística (ou função sigmoide) para transformar sua saída e mapear qualquer valor real para o intervalo entre 0 e 1. Isso é útil para estimar probabilidades em problemas de classificação. A fórmula matemática que representa o modelo é dada por:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}\cdot\mathbf{x}+b)}} \tag{2.3}$$

Nesta equação, $\mathbf{x}$ representa o vetor de características da amostra, enquanto $\mathbf{w}$ e b são os parâmetros do modelo que são aprendidos durante o treinamento. A expressão $e^{-(\mathbf{w}\cdot\mathbf{x}+b)}$ é uma função exponencial de base e. O objetivo dessa função exponencial é transformar a combinação linear das características e dos parâmetros do modelo para um valor que pode ser mapeado para uma probabilidade entre 0 e 1 através da função sigmoide, como é ilustrado na Figura 4.

Figura 4 – Logistic Regression. Fonte: (KANADE, 2022)



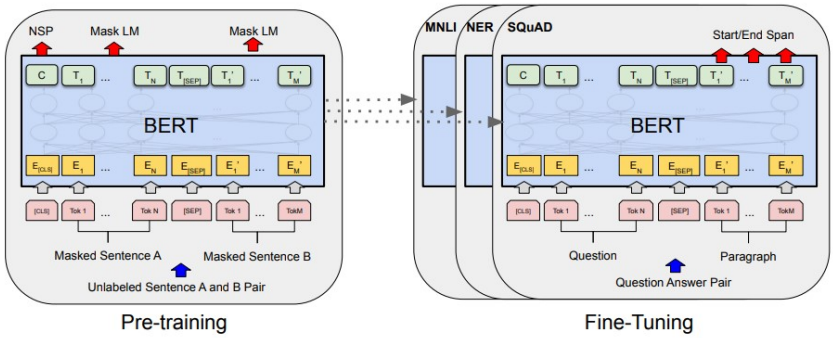
O objetivo durante o treinamento é ajustar os parâmetros $\mathbf{w}$ e b de forma a maximizar a verossimilhança dos dados de treinamento, o que, por sua vez, torna o modelo eficaz na classificação de novas amostras.

2.3.5 BERT, *fine-tuning*

Um procedimento comum ao se usar LLMs é tomar como base uma LLM pré-treinada como BERT, (DEVLIN *et al.*, 2018), onde a rede neural foi treinada com uma grande quantidade de textos, e ajustar os parâmetros da rede para uma tarefa específica. Esse procedimento, conhecido como *fine-tuning*, é uma técnica eficaz para a classificação de texto, capitalizando sobre a rica representação semântica aprendida durante o treinamento prévio do modelo em um grande corpus de texto (MOHAMMADI; CHAPON, 2020). Assim, em vez de iniciar o aprendizado do zero, o modelo pré-treinado BERT serve como ponto de partida e é ajustado com um conjunto de treinamento específico para a tarefa de classificação desejada.

O *fine-tuning* ajusta os pesos do modelo pré-treinado para se adaptar melhor à nova tarefa, permitindo que o BERT seja efetivamente usado em uma ampla variedade de tarefas de classificação de texto. O BERT é considerado um dos principais algoritmos de classificação de texto do estado da arte (GONZÁLEZ-CARVAJAL; GARRIDO-MERCHÁN, 2020). A Figura 5 representa a abordagem de pré-treinamento e *fine-tuning* do BERT.

Figura 5 – Processo de pré-treino e *fine-tuning* BERT. Fonte: (DEVLIN *et al.*, 2018)



Na Figura 5 os procedimentos gerais de pré-treinamento e *fine-tuning* do BERT. Com exceção das camadas de saída, as mesmas estruturas arquitetônicas são empregadas tanto na fase de pré-treinamento quanto no ajuste fino. Os parâmetros do modelo pré-treinado são utilizados como ponto de partida para inicializar modelos destinados a diferentes tarefas específicas.

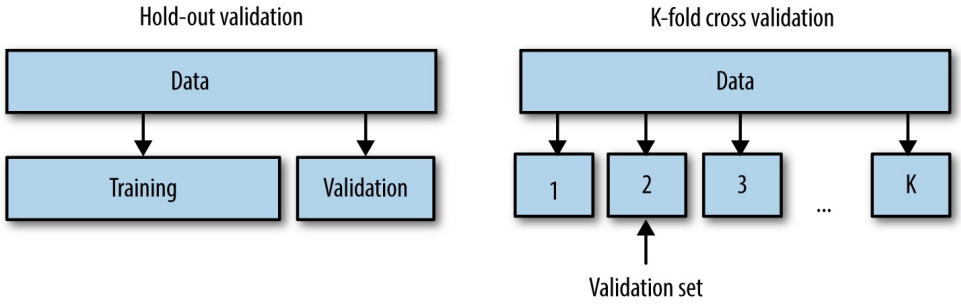
Cada algoritmo de aprendizado de máquina apresenta características distintas que os tornam mais ou menos adequados para diferentes cenários de classificação de texto. Enquanto árvores de decisão oferecem interpretabilidade e são eficazes para lidar com um grande número de características (HASTIE; TIBSHIRANI; FRIEDMAN, 2009), as redes neurais proporcionam flexibilidade para capturar complexidades nos dados (GOODFELLOW; BENGIO; COURVILLE, 2016). O algoritmo Naïve Bayes é conhecido por sua simplicidade e eficácia em espaços de alta dimensionalidade (MITCHELL, 1997), e o SVM é notável por sua eficiência em separar classes complexas em dados de alta dimensionalidade (BISHOP, 2016). Em contraste, o uso de modelos pré-treinados como BERT, ajustados através de *fine-tuning*, representa uma abordagem mais moderna que capitaliza em representações semânticas ricas para oferecer desempenho de estado da arte em uma variedade de tarefas de classificação de texto.

Dado que a precisão na classificação dos algoritmos pode variar conforme o conjunto de dados ou o domínio de aplicação, torna-se essencial avaliar o desempenho de classificação dos algoritmos. Os métodos para realizar essas avaliações serão apresentados na seção subsequente.

2.4 Avaliação da Performance de Classificação

Após a etapa de classificação, é hora de avaliar a performance do modelo de classificação treinado. Para isso, é necessário escolher o esquema de validação e as métricas de avaliação da performance de classificação. Um dos esquemas de validação comumente utilizados é o método *Hold-out*, onde o conjunto de dados é dividido em subconjuntos de treinamento e teste. Outro esquema é o método de validação cruzada k-fold (k-*fold cross-validation*), que divide o conjunto de dados em k subconjuntos. Em cada iteração, $k-1$ subconjuntos são usados para treinamento e o subconjunto restante é usado para teste. As ilustrações desses esquemas são apresentadas na Figura 6.

Figura 6 – Esquema de validação. Fonte: (ZHENG, 2015)



Além do esquema de validação, é necessário selecionar métricas adequadas para avaliar a performance do modelo. Métricas comuns de avaliação são geralmente baseadas em quatro valores (POWERS, 2020):

- *TP (True Positive ou Verdadeiros Positivos):* exemplos classificados como positivos e que são da classe positiva.
- *TN (True Negative ou Verdadeiros Negativos):* exemplos classificados como negativos e que são da classe negativa.
- *FP (False Positive ou Falsos Positivos):* exemplos classificados como positivos, mas que são da classe negativa.
- *FN (False Negative ou Falsos Negativos):* exemplos classificados como negativos, mas que são da classe positiva.

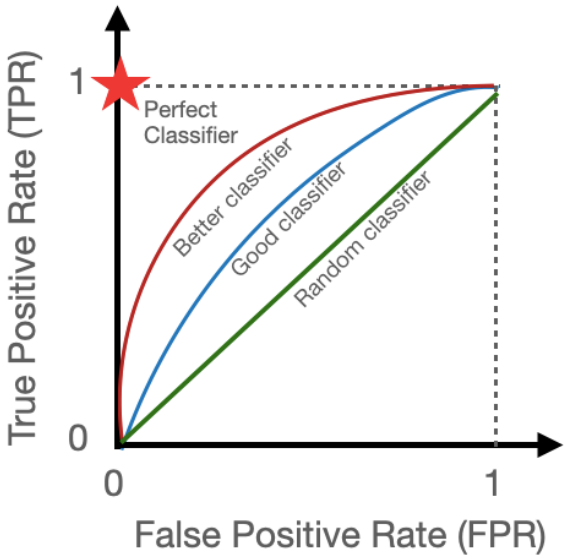
Com base nesses valores, algumas das métricas de avaliação mais utilizadas para classificação são:

- *Precisão (Precision)*: esta é a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos positivos, ou seja, tudo que foi classificado como positivo pelo modelo de classificação. É calculada pela fórmula: $\frac{TP}{TP+FP}$.
- *Revocação (Recall)*: esta é a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos, ou seja, a soma de todos os exemplos cujo rótulo real é positivo. É calculada pela fórmula: $\frac{TP}{TP+FN}$.
- *F1-Score*: esta é a média harmônica entre precisão e revocação. Fornece um equilíbrio entre essas duas métricas. É dada pela fórmula: $2 \times \frac{Precision \times Recall}{Precision+Recall}$.
- *Acurácia (Accuracy)*: esta é a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões. É dada pela fórmula: $\frac{TP+TN}{TP+TN+FP+FN}$.

Também são comumente utilizadas para avaliar a performance de modelos de classificação a curva ROC (Receiver Operating Characteristic) e a AUC-ROC (Area Under the ROC Curve) (POWERS, 2020). A curva ROC é uma representação gráfica que ilustra a relação entre a sensibilidade (taxa de verdadeiros positivos) e a especificidade (taxa de verdadeiros negativos) de um modelo de classificação para diferentes pontos de corte (NARKHEDE, 2018). Ela é construída ao variar o limiar de classificação do modelo e calcular a sensibilidade e a 1-especificidade correspondentes em cada ponto de corte.

A AUC-ROC é uma medida de desempenho que resume a curva ROC em um único valor numérico. Essa métrica representa a área sob a curva ROC e varia de 0 a 1. Quanto maior for a AUC-ROC, melhor será o desempenho do modelo de classificação. Uma AUC-ROC de 0,5 indica um modelo que tem um desempenho equivalente ao acaso, enquanto uma pontuação de 1,0 representa um modelo perfeito que é capaz de distinguir perfeitamente entre as classes positiva e negativa. Valores entre 0,5 e 1,0 indicam que o modelo possui algum poder de discriminação, sendo que valores mais próximos de 1,0 indicam um melhor desempenho. Geralmente os maiores valores de AUC-ROC são obtidos quando há uma classificação correta dos exemplos com maior confiança de classificação. Na Figura 7, é apresentada uma ilustração da curva ROC.

Figura 7 – Curva ROC. Fonte: (HOO; CANDLISH; TEARE, 2017)



3 PANORAMA DE MERCADO, INVESTIMENTOS E RESULTADOS ESPERADOS

Neste capítulo, será apresentado o panorama atual do mercado de Inteligência Artificial (IA), com foco em investimentos e na implementação desta tecnologia em diversas organizações. Além disso, será discutido o papel fundamental do processamento de linguagem natural (NPL) na IA e suas implicações para a classificação de texto. Por fim, os resultados esperados deste projeto, que envolve a utilização de IA para a detecção de cadastros desatualizados, serão detalhados. Este capítulo busca proporcionar uma visão completa do cenário atual da IA e do NPL, e como essas tecnologias podem ser aplicadas para melhorar a eficiência operacional, a conformidade com a legislação e a experiência do cliente.

3.1 Panorama de Mercado e Investimento em Inteligência Artificial

A IA está passando por um crescimento significativo no mercado global, impulsionado pela adoção crescente e pelos investimentos cada vez maiores das organizações. De acordo com a (MCKINSEY, 2022), a adoção dessa tecnologia mais que dobrou desde 2017, com aproximadamente 50% a 60% das organizações utilizando IA em pelo menos uma função, como é possível ver na Figura 8.

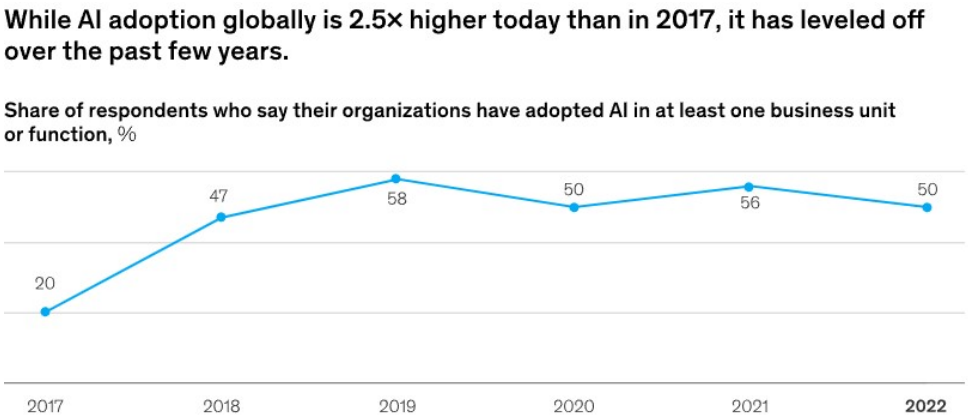


Figura 8 – Adoção de IA nas empresas. Fonte: (MCKINSEY, 2022)

Já os recursos financeiros dentro das empresas com a finalidade em IA saiu de 40% em 2018 para 52% em 2022. Além disso, o investimento em IA tem aumentado, com mais da metade das organizações relatando níveis significativos de investimento, e aproximadamente 63% esperam aumentar o investimento nos próximos três anos, como pode ser visto na Figura 9.

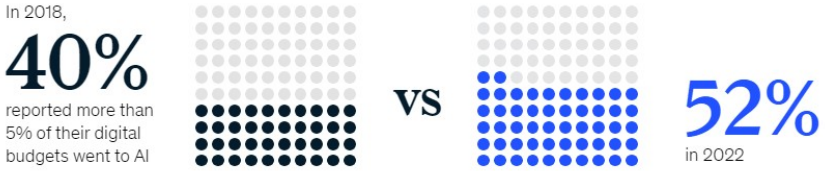


Figura 9 – Aumento de recursos financeiros para investimento em IA, Fonte: (MCKINSEY, 2022)

Dentro do mercado de IA, o processamento de linguagem natural (NPL, em inglês) desempenha um papel fundamental, especialmente no contexto da classificação de texto. A capacidade de compreender e classificar grandes volumes de texto de forma eficiente e precisa é essencial para diversas aplicações, como análise de sentimentos, processamento de linguagem natural e sistemas de recomendação. O NPL avançou rapidamente nos últimos anos, tornando-se uma das capacidades de IA mais comumente implantadas, logo atrás da visão computacional.

Além disso, as organizações líderes em IA estão se concentrando em práticas avançadas de IA, como o uso de plataformas de desenvolvimento de IA de ponta a ponta e o envolvimento de funcionários não técnicos na criação de aplicativos de IA por meio de programas de baixo código ou sem código (CUNNINGHAM, 2023). Essas práticas permitem que as organizações acelerem o desenvolvimento e a implantação de aplicativos de IA, incluindo aqueles relacionados à classificação de texto. Além disso, essas organizações estão adotando medidas para mitigar os riscos associados à IA, incluindo a governança de dados e a realização de testes e monitoramento contínuos de modelos de IA para identificar possíveis problemas.

No entanto, à medida que a adoção da IA e do NPL continua a crescer, surgem desafios e considerações importantes. Um desses desafios é o impacto ambiental do treinamento de modelos de IA, que pode ser intensivo em termos de energia. As organizações estão buscando reduzir o consumo de energia relacionado à IA, incluindo a aplicação de práticas de eficiência energética no treinamento e na execução de modelos de IA. Isso é particularmente relevante no contexto da classificação de texto, pois o treinamento de modelos de IA para essa tarefa pode envolver grandes volumes de dados de texto. Portanto, a eficiência energética e a sustentabilidade são considerações importantes no desenvolvimento e uso de tecnologias de NPL para a classificação de texto (MCKINSEY, 2022). Apesar dos avanços significativos da IA de de modelos de NPL nos últimos anos, ainda é necessário adaptar / treinar modelos para tarefas específicas de forma a se obter resultados satisfatórios em aplicações práticas.

3.2 Resultados Esperados

Os resultados esperados para este projeto estão alinhados com os objetivos de automação, eficiência, conformidade com a LGPD e a melhoria da experiência do cliente. Em termos de melhoria de qualidade e eficiência, espera-se que o modelo de classificação de texto seja capaz de identificar com assertividade os casos em que os dados cadastrais estão desatualizados. Isso resultará em um banco de dados mais preciso e atualizado, o que pode levar a um aumento significativo da eficiência operacional. A identificação automática de cadastros desatualizados deve agilizar o processo de atualização de dados, liberando tempo e recursos que podem ser redirecionados para outras tarefas importantes.

No que se refere à conformidade com a LGPD, especificamente no tocante à precisão e atualização dos dados pessoais, a identificação e correção rápida de cadastros desatualizados por meio deste projeto são essenciais. A LGPD estabelece que os dados pessoais devem ser exatos e atualizados, e o titular tem o direito de corrigir dados incompletos, inexatos ou desatualizados. Portanto, ao permitir a correção eficiente de cadastros desatualizados, este projeto auxilia as empresas a cumprir esses requisitos da Lei Geral de Proteção de Dados. Isso não só salvaguarda a privacidade e os dados pessoais dos cidadãos, mas também mitiga o risco de possíveis penalidades por não conformidade para as empresas.

A melhoria da experiência do cliente é outro resultado esperado importante. Ao manter os dados cadastrais atualizados, as empresas poderão se comunicar de forma mais eficaz com os clientes, o que pode levar a uma maior satisfação e a uma experiência de cliente melhorada, além de gerar menos insatisfação de clientes que não deveriam estar envolvidos no processo de comunicação.

Por fim, a automação da detecção de cadastros desatualizados pode ajudar a reduzir erros manuais e inconsistências nos dados. Com isso, é possível alcançar economia de custos, ao evitar penalidades por não conformidade e otimizar o tempo dos funcionários.

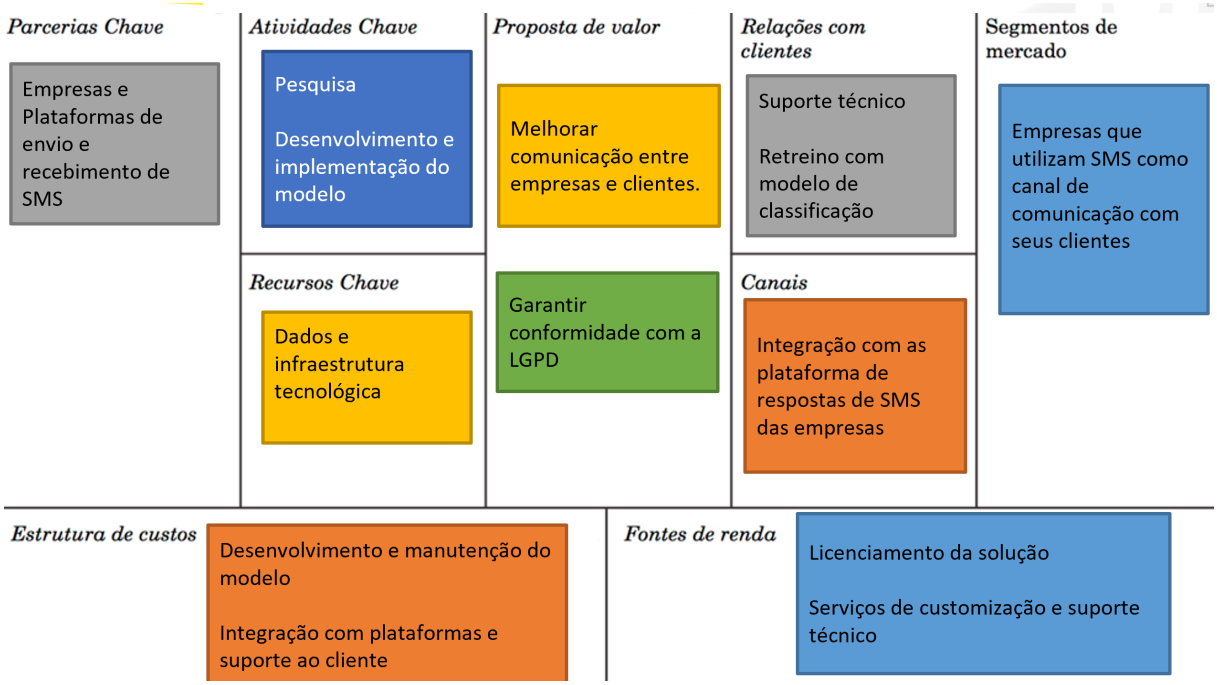
4 CANVAS, BENCHMARK TECH, SWOT

Neste capítulo, será realizada uma análise detalhada da proposta, iniciando-se com o Canvas de Negócios, uma ferramenta estratégica que permite visualizar e estruturar de maneira clara o modelo de negócio proposto. Em seguida, será feito um *Benchmarking* Tecnológico, examinando as melhores práticas e soluções inovadoras adotadas por líderes de mercado no campo da Inteligência Artificial e Big Data, dentro do contexto da solução. Posteriormente, uma análise SWOT será conduzida para identificar forças, fraquezas, oportunidades e ameaças inerentes à implementação dessas tecnologias.

4.1 Canvas

O Modelo de Negócios Canvas é uma ferramenta de gestão estratégica, que permite desenvolver e arquitetar modelos de negócio seja novo ou existens. O Canvas é visual e intuitivo, composto por nove blocos que representam as principais áreas de uma empresa ou projeto (GONÇALVES, 2019). A Figura 10 a seguir representa o Canvas da proposta da solução deste trabalho.

Figura 10 – Canvas da solução. Fonte: Autor



A proposta de valor centra-se em aprimorar a comunicação empresa-cliente e assegurar a conformidade com a LGPD. O projeto busca tornar as interações mais relevantes e eliminar a comunicação desnecessária, enquanto ajuda as empresas a proteger os dados dos clientes e a evitar penalidades legais.

As atividades-chave envolvem a pesquisa, desenvolvimento e implementação do modelo de classificação de texto. Para isso, é crucial o acesso a dados relevantes e uma robusta infraestrutura tecnológica, que são considerados os recursos-chave do projeto.

A relação com os clientes será mantida através de suporte técnico contínuo e re-treinamento do modelo de classificação conforme necessário, garantindo a eficácia e a atualização do modelo ao longo do tempo. O principal canal de entrega é a integração do modelo com as plataformas de respostas de SMS das empresas, permitindo a aplicação prática e direta do modelo.

O projeto está voltado para empresas que utilizam SMS como canal de comunicação com seus clientes, representando o principal segmento de mercado. A estrutura de custos está ligada principalmente ao desenvolvimento e manutenção do modelo, bem como à integração com as plataformas de respostas de SMS e ao suporte ao cliente.

As principais fontes de renda provêm do licenciamento da solução para empresas, bem como de serviços adicionais de customização e suporte técnico. Isso fornece um modelo de negócio sustentável que beneficia tanto a empresa quanto os clientes.

4.2 *Benchmark* Tecnológico

Existem várias soluções robustas disponíveis para o problema da classificação de texto geral. Estas plataformas utilizam métodos de classificação de última geração e, apesar de serem ferramentas poderosas, algumas podem apresentar desafios, como custos elevados, complexidade de uso e documentação insuficiente, elementos que precisam ser cuidadosamente avaliados ao selecionar a ferramenta mais apropriada para um projeto específico, a seguir na Tabela 3, será mostrado as principais soluções do mercado.

Tabela 3 – Comparação das Plataformas de Classificação de Textos

Plataforma	**Modelos Pré-treinados**	**Classificação de Textos**
Google Cloud NLP	Sim	Sim
Amazon Comprehend	Sim	Sim
IBM Watson NLP	Sim	Sim
Microsoft Azure Text Analytics	Sim	Sim
SpaCy	Sim	Sim
NLTK	Não	Sim
Hugging Face Transformers	Sim	Sim
ChatGPT (OpenAI)	Sim	Sim

Na Tabela 3 são apresentadas plataformas que podem ser utilizadas para o proces-

samento de linguagem natural e duas características: se possuem modelos de classificação pré-treinados e se podem ser utilizadas para a classificação de textos em geral. Essas plataformas podem ser bibliotecas de código aberto, como SpaCy ou NLTK, ou serviços oferecidos por empresas de tecnologia, como Google Cloud NLP, Amazon Comprehend, IBM Watson NLP, Microsoft Azure Text Analytics, Hugging Face Transformers ou ChatGPT (OpenAI).

A Coluna Modelos Pré-treinados, indica se a plataforma oferece modelos de classificação de texto que já foram treinados, são úteis porque podem economizar muito tempo e recursos computacionais. Eles foram treinados em grandes conjuntos de dados e, portanto, já aprendem uma quantidade considerável de informação sobre a linguagem. Esses modelos podem ser usados como estão, ou podem ser ajustados para tarefas mais específicas usando *fine-tuning*.

Por fim, a coluna classificação de textos, informa que a plataforma pode ser usada para tarefas gerais de classificação de texto. A seguir, são apresentados prós e contras para carda uma das plataformas apresentadas na Tabela 2.

Google Cloud NLP

Prós:
- Oferece uma API fácil de usar.
- Fornece uma variedade de recursos NLP úteis além da classificação de texto.
- Os modelos pré-treinados da Google são robustos e foram treinados em uma grande quantidade de dados.

Contras:
- Pode ser um pouco caro, especialmente para grandes volumes de dados.
- Apesar dos modelos pré-treinados serem poderosos, pode não ser a melhor opção se o seu problema é muito específico (como a classificação de respostas de SMS).

Amazon Comprehend

Prós:
- Fácil de usar e integrar com outros serviços da AWS.
- Oferece análises em tempo real, que podem ser úteis para alguns projetos.

Contras:
- Assim como o Google Cloud NLP, pode ser caro para grandes volumes de dados.
- A personalização do modelo pode ser limitada.

IBM Watson NLP

Prós:
- Oferece uma variedade de recursos NLP e a capacidade de personalizar modelos.

Contras:
- A documentação e o suporte podem ser um pouco menos acessíveis em comparação com outras opções.
- Pode não oferecer a mesma escala ou robustez que o Google ou a Amazon.

Microsoft Azure Text Analytics

Prós:
- Oferece uma API de NLP fácil de usar, e se integra bem com outros serviços Azure.

Contras:
- Assim como as outras opções de grandes empresas, pode ser caro para grandes volumes de dados.
- A personalização do modelo pode ser um pouco limitada.

SpaCy

Prós:
- Uma biblioteca Python leve e flexível, oferece a capacidade de treinar seus próprios modelos, o que é uma grande vantagem se você tem um problema muito específico.

Contras:
- Menos "pronto para uso"do que as opções de API acima.
- Requer mais trabalho de desenvolvimento e configuração.

NLTK

Prós:
- Uma das bibliotecas mais antigas e amplamente usadas para NLP em Python, o NLTK oferece muita flexibilidade.

Contras:
- Como o SpaCy, requer mais trabalho de desenvolvimento. Não fornece modelos pré-treinados, então você precisaria fornecer seus próprios dados de treinamento.

Hugging Face Transformers

Prós:
- Acesso a uma ampla gama de modelos de NLP pré-treinados de ponta, incluindo BERT, GPT-2 e outros. Permite a personalização de modelos.

Contras:
- Pode ser um pouco mais difícil de usar do que algumas das outras opções, e pode ser muito custoso para problemas mais simples de NLP.

ChatGPT (OpenAI)

Prós:
- Focado em geração e compreensão de texto conversacional, o que pode ser útil para a classificação de respostas de SMS. Oferece modelos pré-treinados robustos.

Contras: – Menos personalizável em termos de treinamento de novos modelos. A API pode ter um alto custo, dependendo do volume de uso.

A solução deste trabalho visa ser simples e de baixo custo para a classificação de respostas de SMS em relação a possíveis desatualizações cadastrais. Esse modelo analisa o conteúdo das mensagens de texto e identifica aquelas que indicam a necessidade de atualização dos dados cadastrais do remetente. A solução é fácil de implementar, requer recursos mínimos e pode ser integrada em sistemas existentes, permitindo uma resposta automatizada e direcionada para lidar com essas desatualizações.

4.3 SWOT

A análise SWOT é uma ferramenta estratégica que ajuda a identificar as forças, fraquezas, oportunidades e ameaças relacionadas a uma organização ou projeto. É utilizada para entender o ambiente interno (forças e fraquezas) e externo (oportunidades e ameaças) no qual a organização ou projeto está inserido (MADSEN, 2016). A análise SWOT deste projeto pode ser vista na Figura 11 a seguir.

Figura 11 – Swot da solução. Fonte: Autor

Com base na Figura 11, as forças estão na automação e a eficiência da aplicação, pois podem melhorar a eficácia e a velocidade da comunicação entre as empresas e os clientes. Além disso, a conformidade com a Lei Geral de Proteção de Dados (LGPD) é uma força importante, pois garante que as empresas esteja em conformidade com as regulamentações de privacidade de dados. Isso também contribui para a melhoria da experiência do cliente, proporcionando um ambiente seguro e confiável para a troca de informações.

Com relação a fraquezas pode-se notar a dependência de dados é uma fraqueza na sua aplicação. Se a qualidade, a quantidade ou a acessibilidade dos dados forem comprometidas, isso pode afetar a eficácia da aplicação.

A melhoria contínua do modelo é uma oportunidade para aprimorar ainda mais a eficácia da aplicação. Além disso, a demanda das empresas pela conformidade com a LGPD pode ser uma oportunidade para expandir o alcance da aplicação.

Possíveis ameaças podem ser a substituição dos canais tradicionais de SMS por aplicativos, como WhatsApp e Messenger, é uma ameaça potencial para a aplicação. Se mais empresas optarem por esses aplicativos em vez de SMS, isso pode reduzir a demanda pela aplicação.

5 MÉTODO DE PESQUISA - CONSTRUÇÃO DO MVP

Neste capítulo serão apresentados os detalhes no método de pesquisa adotado para o desenvolvimento do projeto. Mais especificamente, como foram feitas as instanciações do método apresentado no Capítulo 2.

5.1 Coleta e Rotulação de Textos

Os dados para este projeto foram obtidos de uma empresa varejista através do Azure Databricks, aos quais o executor do projeto tinha acesso. Esses dados compreendem respostas de SMS relacionadas a uma variedade de serviços prestados pela empresa, incluindo campanhas promocionais, avisos de compras, notificações de entrega e montagem de móveis, avisos gerais, serviços financeiros, entre outros.

Após a coleta desses dados foi utilizada a API da OpenAI, com os modelos GPT-3.5 e GPT-4, para rotular aproximadamente 21 mil exemplos de textos oriundos de respostas de SMS. Cada exemplo foi rotulado como "1" quando indicava um possível problema no cadastro do remetente, sugerindo um destinatário incorreto, e como "0" para qualquer outro contexto. Essa etapa de rotulação automática com os modelos de linguagem permitiu a classificação inicial dos textos de forma eficiente.

Após a rotulação automática, foi realizada uma checagem manual linha a linha de cada classificação. Esse processo de revisão manual foi crucial para garantir a qualidade e acurácia das classificações. No entanto, devido à grande quantidade de dados, essa etapa consumiu bastante tempo, levando mais de 14 dias para ser concluída.

5.2 Pré-processamento de Texto

O pré-processamento de texto foi realizado para remover palavras e caracteres irrelevantes, visando melhorar o desempenho dos algoritmos de classificação. Inicialmente, todas as palavras foram convertidas para minúsculas para garantir uniformidade e evitar duplicidades devido a diferenças de caixa. As *stopwords* foram removidas utilizando a lista de *stopwords* do NLTK, e a simplificação de palavras foi realizada através do processo de *stemming*, utilizando o algoritmo RSLPStemmer, também disponível no NLTK.

5.3 Balanceamento das Classes

No desenvolvimento deste projeto, foi identificado um desbalanceamento significativo entre as classes no conjunto de dados inicial, com 19602 instâncias da classe `0` e apenas 2008 instâncias da classe `1`. Para abordar este desbalanceamento e evitar um viés

indesejado nos modelos de classificação, foi aplicada a técnica de *oversampling* SMOTE (Synthetic Minority Over-sampling Technique) ao conjunto de treinamento.

O SMOTE foi aplicado apenas aos dados de treinamento, resultando em um conjunto balanceado, com 15686 instâncias em cada classe, conforme mostrado abaixo:

```
classe
0     15686
1     15686
```

Esta técnica de balanceamento foi implementada em todos os algoritmos de classificação utilizados neste estudo, com exceção do BERT, que emprega seus próprios métodos para lidar com o desbalanceamento de classes.

5.4 Representação de Texto

Foram usadas cinco técnicas de representação de texto para representar os dados de SMS:

- **Bag-of-words (BoW)**: cada documento foi representado como um vetor de frequências de palavras.

- **Word2vec**: um modelo Word2Vec foi treinado nos dados de SMS com um tamanho de vetor de 100, uma janela de 5 e um *min_count* de 1. Cada documento foi representado como o vetor médio dos vetores de palavras que o compõem, removendo palavras fora do vocabulário.

- **Doc2vec**: um modelo Doc2Vec foi treinado nos dados de SMS com um tamanho de vetor de 100, uma janela de 2 e um *min_count* de 1. Cada documento foi representado como o vetor médio dos vetores de documentos que o compõem.

- **GloVe**: um modelo GloVe pré-treinado foi usado para representar os dados de SMS. Cada documento foi representado como o vetor médio dos vetores de palavras que o compõem, removendo palavras fora do vocabulário. O número de dimensões do modelo GloVe utilizado é de 100.

- **FastText**: um modelo FastText foi treinado nos dados de SMS com um tamanho de vetor de 100, uma janela de 5 e um *min_count* de 1. Cada documento foi representado como o vetor médio dos vetores de palavras que o compõem, removendo palavras fora do vocabulário.

5.5 Classificação

Quatro algoritmos de classificação foram treinados nos dados de representação:

- **SVM**: um modelo SVM com kernel linear foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[1].
- **Logistic Regression**: um modelo Logistic Regression foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[2].
- **Multinomial Naive Bayes**: um modelo MNB, ele foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[3].
- **Random Forest**: um modelo RandomForestClassifier foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[4].
- **BERT**: foi realizado um *fine-tuning* usando o modelo `bert-base-portuguese-cased`. Os dados foram tokenizados com um comprimento máximo de 512, e os pesos das classes foram calculados para lidar com o desbalanceamento. O modelo foi treinado por 3 épocas com um tamanho de lote de treinamento e avaliação de 8 por dispositivo. Foram utilizados 500 passos de aquecimento, um decaimento de peso de 0.01, e a estratégia de avaliação foi aplicada a cada 500 passos[5].

5.6 Avaliação

Os modelos foram avaliados usando as métricas de *precision*, *recall* e *F1-score*. A precisão mede a proporção de exemplos classificados corretamente. O *recall* mede a proporção de exemplos positivos classificados corretamente. O *F1-score* é uma média ponderada da *precision* e do *recall*.

---

1 <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

2 <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

3 <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>

4 <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

5 Para mais detalhes sobre os parâmetros de treinamento do BERT, consulte a documentação do Hugging Face: <https://huggingface.co/transformers/main_classes/trainer.html#trainingarguments>

6 DISCUSSÃO E ANÁLISE DE RESULTADOS

Nesta seção, é realizada uma análise dos modelos de classificação de texto de SMS treinados. Para cada experimento, as métricas de precisão, revocação, F1-score e AUC foram calculadas, todas focadas na classe 1 ou classe de interesse. Estas métricas fornecem uma visão abrangente da eficácia de cada modelo em identificar corretamente a classe 1 no conjunto de dados em estudo.

Os modelos examinados incluem: Suporte a Vetores de Máquina (SVC) com técnicas de pré-processamento como StopWords e Stemmer, Regressão Logística com diversas representações textuais, Naive Bayes Multinomial, Floresta Aleatória, além do avançado modelo de linguagem Transformer BERT. Estes modelos foram treinados utilizando uma variedade de técnicas para a representação textual, como Bag of Words (BoW), Word2Vec, Doc2Vec, Global Vectors for Word Representation (GloVe) e FastText.

A comparação do desempenho na Tabela 4 desses modelos permite uma melhor compreensão de suas forças e fraquezas relativas, bem como a identificação do modelo mais adequado para as necessidades específicas de classificação de texto de SMS, que é o objetivo deste trabalho. Antes de prosseguir com a análise detalhada dos resultados, é importante esclarecer a formatação da Tabela 4. Os melhores resultados para cada medida estão assinalados em negrito e os melhores resultados de cada representação para cada medida estão assinalados em sublinhado. Isso foi feito para facilitar a comparação entre os diferentes modelos e representações.

Tabela 4 – Comparação modelos de classificação de texto para a classe 1 (cadastro desatualizado).

Representação	**Modelo**	**Precisão**	**Revocação**	F1-**Score**	**AUC-ROC**
BoW	SVC	0.37	0.06	0.11	0.80
	LogisticRegression	0.72	0.87	0.79	0.97
	MultinomialNB	0.51	0.92	0.65	0.97
	Random Forest	0.41	0.92	0.57	0.94
Word2Vec	SVC	0.79	0.89	0.84	0.98
	LogisticRegression	0.41	0.82	0.55	0.92
	Random Forest	0.82	0.71	0.76	0.96
Doc2Vec	SVC	0.56	0.82	0.67	0.96
	LogisticRegression	0.44	0.84	0.58	0.93
	Random Forest	0.72	0.75	0.74	0.96
Glove	SVC	0.43	0.88	0.58	0.94
	LogisticRegression	0.39	0.86	0.54	0.94
	Random Forest	0.73	0.73	0.73	0.96
FastText	SVC	0.57	**0.96**	0.72	0.97
	LogisticRegression	0.60	0.95	0.74	0.97
	Random Forest	0.75	0.90	0.82	0.98
BERT	BERT-BasePortugueseCased	**0.93**	0.94	**0.93**	**0.99**

A análise da Tabela 4 mostra que, de maneira geral, a representação utilizando o `BERT` obteve superioridade, alcançando os melhores resultados para a maioria das medidas, como *Precisão*, *F1-Score*, e *AUC-ROC*. No entanto, a combinação do algoritmo `SVC` com a representação `FastText` destacou-se em *Revocação*, evidenciando a importância de explorar diferentes combinações de representações e algoritmos, dependendo do foco da medida de desempenho.

Observando os resultados dos algoritmos de aprendizado de máquina em combinação com diferentes representações, é notável que o algoritmo `Random Forest` apresentou desempenho superior para a maioria das medidas. Este algoritmo, conhecido por sua robustez e capacidade de modelar relações complexas, provou ser uma escolha sólida para a tarefa de classificação de texto em questão. Por outro lado, o algoritmo `SVM` também se destacou, principalmente na medida de *Revocação*, indicando sua eficácia em minimizar falsos negativos.

Além da representação gerada pelo `BERT`, outras representações também se destacaram na avaliação experimental. O `Word2Vec` obteve resultados significativos em *Precisão* e *F1-Score*, enquanto o `FastText` destacou-se em *Revocação* e *AUC-ROC*. Surpreendentemente, a representação `Bag-of-Words (BoW)`, apesar de sua simplicidade, conseguiu superar algumas abordagens baseadas em *embeddings* em determinadas situações, com exceção da abordagem baseada em `BERT`.

Concluindo, ao comparar o desempenho do `BERT` com as demais abordagens, é crucial destacar as diferenças significativas observadas. A diferença entre o `BERT` e o segundo melhor modelo evidencia o ganho substancial proporcionado por abordagens mais complexas e sofisticadas. Essas diferenças quantificáveis reforçam a importância de considerar o *trade-off* entre complexidade e desempenho ao selecionar modelos e representações para tarefas específicas de classificação de texto.

7 CONCLUSÕES

O SMS continua sendo uma ferramenta vital para a comunicação entre empresas e clientes, especialmente em transações e atualizações. No entanto, lidar com respostas em linguagem natural apresenta desafios significativos, exigindo soluções robustas e precisas para identificar mensagens que requerem atualização cadastral. Este trabalho abordou esses desafios, avaliando diversas técnicas de classificação de texto, incluindo SVM, LLMs como BERT, entre outros, com diferentes abordagens de representação textual.

O BERT destacou-se entre os modelos avaliados, apresentando os melhores resultados em todas as métricas consideradas: precisão, revocação, F1-score e AUC. Este desempenho superior sugere que o BERT pode ser considerado um padrão de referência para tarefas que exigem alta precisão e revocação em linguagem natural.

No entanto, a seleção de um modelo vai além da sua eficácia. Fatores como custo operacional, facilidade de personalização e escalabilidade são cruciais. O *Random Forest*, por exemplo, demonstrou ser uma opção robusta e competitiva, sendo uma alternativa viável quando se considera a velocidade na predição e outros fatores práticos. Assim, a escolha do modelo mais adequado deve ser contextualizada, levando em conta as necessidades específicas da aplicação.

Este estudo fornece insights valiosos para empresas que buscam otimizar suas comunicações, sendo crucial para a conformidade com regulamentações de proteção de dados, como a LGPD no Brasil. A identificação precisa de cadastros desatualizados é vital para manter a integridade dos dados e a privacidade do cliente.

Para futuras investigações, a exploração de outras técnicas e modelos de classificação de texto, incluindo LLMs mais recentes, é recomendada. A eficácia dos modelos pode variar de acordo com o domínio e a complexidade das mensagens, tornando relevante sua avaliação em diferentes contextos. Estudos adicionais focados na otimização desses modelos poderiam fornecer diretrizes mais claras para aprimorar ainda mais sua eficácia.

Em resumo, este trabalho contribui significativamente para o campo da classificação de texto, servindo como um guia prático e teórico para empresas e pesquisadores interessados em implementar soluções automatizadas para a detecção de necessidades de atualização cadastral em mensagens de SMS.

REFERÊNCIAS

ABAYOMI-ALLI, O. *et al.* A review of soft techniques for sms spam classification: Methods, approaches and applications. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 86, p. 197–212, 2019.

AGGARWAL, C. **Machine Learning for Text**. Springer International Publishing, 2018. ISBN 9783319735306. Available at: <https://books.google.com.br/books?id=6UJBtAEACAAJ>.

BALAKRISHNAN, V.; LLOYD-YEMOH, E. Stemming and lemmatization: A comparison of retrieval performances. 2014.

BISHOP, C. **Pattern Recognition and Machine Learning**. Springer New York, 2016. (Information Science and Statistics). ISBN 9781493938438. Available at: <https://books.google.com.br/books?id=kOXDtAEACAAJ>.

Brasil. **LEI Nº 13.709, DE 14 DE AGOSTO DE 2018**. 2018. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 11 mar. 2023.

BROWN, J.; SHIPMAN, W.; VETTER, R. Sms: The short message service. **Computer**, v. 40, p. 106 – 110, 01 2008.

CONRADO, M. d. S. **O efeito do uso de diferentes formas de extração de termos na compreensibilidade e representatividade dos termos em coleções textuais na língua portuguesa**. 2009. Tese (Doutorado) — Universidade de São Paulo, 2009.

CUNNINGHAM, R. **Announcing a next-generation AI Copilot in Microsoft Power Apps that will transform low-code development**. 2023. Disponível em: <https://powerapps.microsoft.com/en-us/blog/announcing-a-next-generation-ai-copilot-in-microsoft-power-apps-that-will-transform-low-code-develo>. Acesso em: 04 jun. 2023.

DEVLIN, J. *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DING, B. *et al.* Is gpt-3 a good data annotator? **arXiv preprint arXiv:2212.10450**, 2022.

GONÇALVES, A. **Canvas: Como estruturar seu modelo de negócios**. 2019. Disponível em: <https://www.sebraepr.com.br/canvas-como-estruturar-seu-modelo-de-negocios/>. Acesso em: 04 jun. 2023.

GONZÁLEZ-CARVAJAL, S.; GARRIDO-MERCHÁN, E. C. Comparing bert against traditional machine learning text classification. **arXiv preprint arXiv:2005.13012**, 2020.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Available at: <https://books.google.com.br/books?id=Np9SDQAAQBAJ>.

HACOHEN-KERNER, Y.; MILLER, D.; YIGAL, Y. The influence of preprocessing on text classification using a bag-of-words representation. **PloS one**, Public Library of Science San Francisco, CA USA, v. 15, n. 5, p. e0232525, 2020.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition**. Springer New York, 2009. (Springer Series in Statistics). ISBN 9780387848587. Available at: <https://books.google.com.br/books?id=tVIjmNS3Ob8C>.

HOO, Z. H.; CANDLISH, J.; TEARE, D. **What is an ROC curve?** [*S.l.: s.n.*]: BMJ Publishing Group Ltd and the British Association for Accident ..., 2017. 357–359 p.

JANG, B.; KIM, I.; KIM, J. W. Word2vec convolutional neural networks for classification of news articles and tweets. **PloS one**, Public Library of Science San Francisco, CA USA, v. 14, n. 8, p. e0220976, 2019.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. Pearson Prentice Hall, 2009. (Prentice Hall series in artificial intelligence). ISBN 9780131873216. Available at: <https://books.google.com.br/books?id=fZmj5UNK8AQC>.

KANADE, V. **What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices**. 2022. Disponível em: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>. Acesso em: 14 set. 2023.

KHANNA, C. **Text pre-processing: Stop words removal using different libraries**. 2021. Disponível em: <https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a>. Acesso em: 04 jun. 2023.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. *In*: PMLR. **International conference on machine learning**. [*S.l.: s.n.*], 2014. p. 1188–1196.

LIAO, W. *et al.* Mask-guided bert for few shot text classification. **arXiv preprint arXiv:2302.10447**, 2023.

MADSEN, D. Ø. Swot analysis: a management fashion perspective. **International Journal of Business Research**, v. 16, n. 1, p. 39–56, 2016.

MCCALLUM, A.; NIGAM, K. *et al.* A comparison of event models for naive bayes text classification. *In*: MADISON, WI. **AAAI-98 workshop on learning for text categorization**. [*S.l.: s.n.*], 1998. v. 752, n. 1, p. 41–48.

MCKINSEY, C. **The state of AI in 2022—and a half decade in review**. 2022. Disponível em: <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review>. Acesso em: 10 jun. 2023.

MIKOLOV, T. *et al.* Distributed representations of words and phrases and their compositionality. **Advances in neural information processing systems**, v. 26, 2013.

MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Available at: <https://books.google.com.br/books?id=EoYBngEACAAJ>.

MKOM. **API de SMS: o que preciso saber sobre?** 2022. Disponível em: <https://mkom.com.br/api-de-sms-o-que-preciso/>. Acesso em: 10 fev. 2023.

MOHAMMADI, S.; CHAPON, M. Investigating the performance of fine-tuned text classification models based-on bert. *In*: IEEE. **2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)**. [*S.l.: s.n.*], 2020. p. 1252–1257.

MULLEN, L. A. *et al.* Fast, consistent tokenization of natural language text. **Journal of Open Source Software**, The Open Journal, v. 3, n. 23, p. 655, 2018.

NARKHEDE, S. Understanding auc-roc curve. **Towards Data Science**, v. 26, n. 1, p. 220–227, 2018.

NASEEM, U. *et al.* A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. **Transactions on Asian and Low-Resource Language Information Processing**, ACM New York, NY, v. 20, n. 5, p. 1–35, 2021.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. *In*: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [*S.l.: s.n.*], 2014. p. 1532–1543.

PITA, M.; PAPPA, G. L. Strategies for short text representation in the word vector space. *In*: IEEE. **2018 7th Brazilian Conference on Intelligent Systems (BRACIS)**. [*S.l.: s.n.*], 2018. p. 266–271.

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. **arXiv preprint arXiv:2010.16061**, 2020.

SHAKEEL, M. H.; KARIM, A.; KHAN, I. A multi-cascaded deep model for bilingual sms classification. **arXiv preprint arXiv:1911.13066**, 2019.

SHIHA, M.; AYVAZ, S. The effects of emoji in sentiment analysis. **Int. J. Comput. Electr. Eng.(IJCEE.)**, v. 9, n. 1, p. 360–369, 2017.

TAN, P. *et al.* **Introduction to Data Mining**. Pearson, 2019. (What's New in Computer Science Series). ISBN 9780133128901. Available at: <https://books.google.com.br/books?id=_ZQ4MQEACAAJ>.

THINKDATA. **ENRIQUECIMENTO CADASTRAL**. 2022. Disponível em: <https://www.thinkdata.com.br/guia-de-solucoes/enriquecimento-cadastral/>. Acesso em: 10 fev. 2023.

WANG, C.; LI, M.; SMOLA, A. J. Language models with transformers. **arXiv preprint arXiv:1904.09408**, 2019.

WANG, S. *et al.* Want to reduce labeling cost? gpt-3 can help. **arXiv preprint arXiv:2108.13487**, 2021.

YEHOSHUA, R. **Random Forests**. 2023. Disponível em: <https://medium.com/@roiyeho/random-forests-98892261dc49>. Acesso em: 14 set. 2023.

ZHAO, W. X. *et al.* A survey of large language models. **arXiv preprint arXiv:2303.18223**, 2023.

ZHENG, A. **Evaluating Machine Learning Models: A Beginner's Guide to Key Concepts and Pitfalls**. O'Reilly Media, 2015. ISBN 9781491932469. Available at: <https://books.google.com.br/books?id=OFhauwEACAAJ>.