

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma comparação entre aprendizado de máquina tradicional e aprendizado profundo para classificação de títulos de notícias

Caio C. Fasolak Alves

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

Orientadora
Prof.a Dr.a Roberta Barbosa Oliveira

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma comparação entre aprendizado de máquina tradicional e aprendizado profundo para classificação de títulos de notícias

Caio C. Fasolak Alves

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

Prof.a Dr.a Roberta Barbosa Oliveira (Orientadora)
CIC/UnB

Prof. Dr. Thiago Paulo Faleiros
Universidade de Brasília

Prof. Dr. Camilo Chang Dórea
Universidade de Brasília

Prof. Dr. Guilherme Novaes Ramos
Coordenador do Curso de Engenharia da Computação

Brasília, 21 de julho de 2025

Dedicatória

Dedico este trabalho especialmente à minha esposa. Juliana, que esteve comigo em todos os momentos desde o início dessa jornada, sempre me ajudando a seguir em frente e me superar, chegando onde pensei que jamais chegaria. E a minha recém falecida mãe, Estelina, que em seus últimos pedidos me fortaleceu a finalizar minha graduação.

Agradecimentos

Agradeço aos meus pais, Ivan e Estelina, por terem feito tudo o que fizeram, proporcionando-me chegar onde estou, à minha esposa, Juliana, pelo apoio incondicional em todos os momentos. Também agradeço aos meus amigos Jean, Matheus, Marcelo e Rafael por terem enfrentado parte dessa jornada comigo e termos superados as dificuldades juntos. Agradeço especialmente à Profa. Dra. Roberta Barbosa Oliveira pela paciência com as adversidades que passei, e por ter aceitado me orientar mesmo sem ter mais vagas.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O mercado financeiro é um setor dinâmico que envolve a negociação de diversos ativos, como ações, títulos, moedas e *commodities*, englobando uma ampla gama de participantes, desde investidores individuais até instituições financeiras. Sua natureza complexa e volátil apresenta desafios e oportunidades associados ao uso de informações relacionadas a cada tipo de ativo. A classificação automática de títulos de notícias é uma tarefa relevante no contexto do aprendizado de máquina, especialmente considerando o crescente volume de informações publicadas diariamente em portais sobre o mercado financeiro. As técnicas tradicionais de aprendizado de máquina vêm sendo complementadas e, em alguns casos, substituídas por métodos mais avançados baseados em aprendizado de máquina profundo. Portanto, o objetivo deste trabalho é realizar a classificação automática de títulos de notícias relacionadas ao mercado financeiro, analisando modelos tradicionais e de aprendizado profundo. Para isso, foram utilizadas três bases de dados compostas por títulos coletados de portais financeiros (Suno, InfoMoney e Money Times), sendo aplicadas técnicas de pré-processamento textual e extração de características. Os modelos de classificação foram avaliados com base em métricas como acurácia e F1-Score. Os melhores resultados foram obtidos com o modelo BERT (*Bidirectional Encoder Representations from Transformers*), que superou os demais algoritmos em todas as bases, alcançando 83.7% de acurácia para o Suno, 84.8% de acurácia para o InfoMoney e 89.3% de acurácia para o Money Times. Os experimentos demonstram que modelos baseados em aprendizado profundo apresentam desempenho superior em tarefas de classificação de textos curtos, como títulos de notícias, destacando-se como alternativa promissora para categorização automática em domínios especializados.

Palavras-chave: Aprendizado de Máquina, Processamento de Linguagem Natural, *Transformers*, Aprendizado Profundo, Classificação de Texto

Abstract

The financial market is a dynamic sector that involves the trading of various assets, such as stocks, bonds, currencies, and commodities, encompassing a wide range of participants, from individual investors to financial institutions. Its complex and volatile nature presents challenges and opportunities associated with the use of information related to each type of asset. Automatic classification of news headlines is an important task in the context of machine learning, especially considering the growing volume of information published daily on financial news portals. Traditional machine learning techniques have been increasingly complemented, and in some cases, replaced by more advanced methods based on deep learning. This work aims to perform automatic classification of news headlines related to the financial market by analyzing both traditional and deep learning models. Three datasets composed of headlines collected from financial portals (Suno, InfoMoney, and Money Times) were used, with the application of textual preprocessing and feature extraction techniques. The classification models were evaluated using metrics such as accuracy and F1-score. The best results were obtained with the BERT model (Bidirectional Encoder Representations from Transformers), which outperformed other algorithms on all datasets, achieving 83.7% accuracy for Suno, 84.8% for InfoMoney, and 89.3% for MoneyTimes. Deep learning models demonstrate superior performance in short text classification tasks, thereby demonstrating their potential for automatic categorization in specialized domains.

Keywords: Machine Learning, Natural Language Processing, Transformers, Deep Learning, Text Classification

Sumário

1 Introdução **1**

1.1 Contextualização 1

1.2 Objetivos 2

1.3 Estrutura do Trabalho 3

2 Fundamentação Teórica **4**

2.1 O mercado financeiro 4

2.2 Processamento de Linguagem Natural 5

2.2.1 Classificação de Dados Textuais 5

2.2.2 Técnicas de Normalização Textual 6

2.2.3 Extração das Características 7

2.3 Algoritmos de Aprendizado de Máquina 9

2.3.1 Naive Bayes 10

2.3.2 K-Nearest Neighbors 10

2.3.3 *Support Vector Machine* 11

2.3.4 Redes Neurais Artificiais 11

2.3.5 *Transformers* 12

2.4 Métricas de Desempenho 15

3 Trabalhos Relacionados **19**

4 Metodologia **23**

4.1 Classificação de Títulos de Notícias entre os Tópicos mais Recorrentes . . . 23

4.2 Bases de Dados 24

4.3 Pré-processamento 26

4.4 Extração de Características do Texto 27

4.5 Treinamento dos Algoritmos 28

4.6 Avaliação dos Resultados 30

5 Resultados **31**
5.1 Ambiente de Desenvolvimento 31
5.2 Principais Palavras do Corpus 31
5.3 Modelos Tradicionais 32
5.4 *Deep Learning* 34

6 Conclusão **41**

Referências **43**

Lista de Figuras

2.1 Representação de neurônio de McCulloch e Pitts [1]. 11
2.2 Diferença de implementação entre ML e DL[1] 12
2.3 Representação de uma Rede Neural Recorrente 13
2.4 Attention 14
2.5 Encoder 14
2.6 Transformer 16
2.7 CM 17

4.1 Pipeline da Classificação de Textos. 24
4.2 Amostra de texto extraído do portal. 25
4.3 Exemplo de título de notícia extraído do portal de notícias Suno Research. 26
4.4 Pipeline da função de tokenização do BERT [2]. 28

5.1 Nuvens de palavras geradas a partir dos *datasets* utilizados.(Autoria Própria) 32
5.2 Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* Suno 35
5.3 Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* InfoMoney 36
5.4 Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* Moneytimes 37
5.5 Matrizes de confusão BERT aplicado à todos os *datasets* 39
5.6 *Loss* e Acurácia do algoritmo BERT para o *dataset* Suno 39
5.7 *Loss* e Acurácia do algoritmo BERT para o *dataset* InfoMoney 40
5.8 *Loss* e Acurácia do algoritmo BERT para o *dataset* Moneytimes 40

Lista de Tabelas

2.1 Exemplo de valores TF-IDF referente às palavras 9

3.1 Revisão de Literatura 22

4.1 Amostra de um *dataset* 27
4.2 Quantidade de amostras dos *datasets* 27
4.3 Hiperpârametros e valores para algoritmos implementados 29

5.1 Hiperparâmetros utilizados nos modelos tradicionais. 33
5.2 Resultados dos modelos tradicionais. 34
5.3 Hiperparâmetros BERT. 38
5.4 Resultados do BERT. 38

Lista de Abreviaturas e Siglas

ALBERT *A Lite Bidirectional Encoder Representations from Transformers.*

ANN *Artificial Neural Network.*

API *Application Programming Interface.*

BERT *Bidirectional Encoder Representations from Transformers.*

Bi-LSTM *Bidirectional Long Short-Term Memory.*

BoW *Bag of Words.*

CM (MC, *Confusion Matrix*).

CNN *Convolutional Neural Networks.*

DL *Deep Learning.*

FinALBERT *Financial A Lite Bidirectional Encoder Representations from Transformers.*

FinBERT *Financial Bidirectional Encoder Representations from Transformers.*

GPU *Graphical Processing Unit.*

IA Inteligência Artificial.

IDE *Integrated Development Environment.*

JSON *JavaScript Object Notation.*

KNN *K-Nearest-Neighbors.*

LLM *Large Language Model.*

LR *Logistic Regression.*

LSTM *Long Short-Term Memory.*

ML (AM, *Machine Learning*).

NB *Naive Bayes.*

NLP *Natural Language Processing.*

NN (RN, *Neural Networks*).

REGEX *Regular Expressions.*

RNN *Recurrent Neural Networks.*

RoBERTa *Robustly Optimized Bidirectional Encoder Representations from Transformers Aproach.*

SVM *Support Vector Machine.*

TF-IDF *Term Frequency - Inverse Document Frequency.*

TPU *Tensor Processing Unit.*

Capítulo 1

Introdução

Neste capítulo será apresentada uma contextualização sobre o assunto abordado neste trabalho, bem como as motivações, o problema a ser tratado e os objetivos, servindo como um guia da estrutura geral deste estudo.

1.1 Contextualização

O mercado financeiro é um setor em constante movimento que envolve a negociação de diversos ativos financeiros, como ações, títulos, moedas e *commodities* [3]. Esse mercado abrange uma vasta gama de participantes, incluindo investidores individuais, instituições financeiras, corretoras e gestores de fundos, entre outros. Graças à constante mudança atrelada a este tipo de mercado, é possível utilizar a tecnologia, incluindo a Inteligência Artificial (IA) [4], campo que visa criar sistemas capazes de realizar tarefas que normalmente requerem inteligência humana, para analisar dados e tomar decisões.

Dentre as principais aplicações do aprendizado de máquina no mercado financeiro, a análise de enormes volumes de dados em tempo real é crucial para identificar tendências, padrões e oportunidades de investimento [5]. Com isso em vista, técnicas tradicionais [6] de análise de dados, como *Naive Bayes* (NB) um método probabilístico que se baseia no Teorema de Bayes [7], o *K-Nearest-Neighbors* (KNN) algoritmo que se baseia na proximidade entre os dados dentro do espaço vetorial [8] e *Support Vector Machine* (SVM) que busca encontrar o hiperplano ótimo que separe os dados de diferentes classes, estão sendo substituídas por métodos mais avançados, tais como o *Bidirectional Encoder Representations from Transformers* (BERT), um dos algoritmos mais recentes e robustos para classificação de texto [9], baseados em aprendizado profundo (*Deep Learning*, DL) [10], com grande capacidade de aprendizado e generalização de aplicações. Graças a esse tipo de tecnologia, investidores e gestores de fundos usam modelos de aprendizado de máquina para tomar decisões mais precisas sobre a compra, venda ou manutenção de ativos [5] e

algoritmos de negociação automatizada utilizam ML para identificar padrões em dados de mercado e executar negociações em alta velocidade [11].

Levando-se em conta os pontos acima citados, vale salientar que existem diversos portais de notícias, como InfoMoney[1], SUNO[2] e MoneyTimes[3], específicos do mercado financeiro, levando ao público um apanhado de informações que podem ser decisivas no momento de compra de ativos para o pequeno investidor. Como esses portais contêm grandes quantidades de dados acessíveis e facilmente armazenáveis, é possível utilizá-los como bases de dados em algoritmos de *Natural Language Processing* (NLP), para realizar a classificação dos títulos das notícias dentre as classes mais recorrentes de cada portal de notícias e comparar os resultados obtidos para verificar o desempenho dos algoritmos utilizados.

Utilizando-se do NLP [12], um campo da IA focado em permitir que computadores interpretem ou gerem textos, é possível obter informações determinantes para se tomar uma decisão de onde e quando aplicar o dinheiro [2][12][13]. Sabendo que o tempo é um fator determinante no mercado de ações [14], é interessante para o pequeno investidor que seja utilizado um método eficaz e que não seja muito oneroso para auxiliá-lo na tomada de decisão. O processo de classificação segue um padrão de ações que se repetem; é necessário, primeiramente, retirar parte dos dados que interferem no aprendizado, também conhecidos como ruído. São retirados caracteres como acentos, pontuações e *stop-words*, termos que servem apenas para manter a leitura e entonação. A partir dos dados previamente pré-processados, os algoritmos são treinados com um conjunto de treinamento definido, permitindo a posterior avaliação de desempenho.

1.2 Objetivos

O objetivo deste trabalho é realizar uma análise comparativa entre modelos baseados em aprendizado tradicional ML e aprendizado profundo DL na tarefa de classificação automática de títulos de notícias provenientes de portais sobre o mercado financeiro, considerando as categorias temáticas definidas previamente. Tendo isso em vista, os principais objetivos específicos são:

- Investigar a eficácia de classificadores tradicionais de aprendizado de máquina, KNN, NB e SVM, na tarefa de categorização temática de títulos de notícias do domínio financeiro;

[1] `https://www.infomoney.com.br/`
[2] `https://www.suno.com.br/`
[3] `https://www.moneytimes.com.br/`

- Analisar o impacto da utilização de técnicas de ensemble com classificadores tradicionais na melhoria do desempenho da classificação de títulos financeiros, verificando se a combinação de modelos contribui para ganhos em acurácia e robustez;
- Avaliar o desempenho do modelo BERT, representando a abordagem baseada em aprendizado profundo, na classificação automática de títulos de notícias em categorias temáticas específicas do contexto financeiro;
- Realizar uma comparação sistemática entre os modelos tradicionais (individuais e em *ensemble*) e o modelo baseado em *transformers*, a fim de identificar qual abordagem apresenta maior precisão, estabilidade e capacidade de generalização para a categorização de notícias financeiras;
- Investigar a influência das características dos dados sobre o desempenho dos modelos avaliados, examinando possíveis variações nos resultados de classificação em função de diferentes fontes de notícias financeiras.

1.3 Estrutura do Trabalho

A estrutura deste documento segue a seguinte organização: no Capítulo 2 são abordados os fundamentos teóricos, nos quais são expostas as teorias estudadas para a realização dos experimentos presentes neste trabalho; no Capítulo 3 é realizada uma revisão de literatura, incluindo os principais trabalhos utilizados como fonte de inspiração e embasamento teórico para as experimentações; no Capítulo 4 são descritas as tarefas, juntamente com a metodologia adotada para a sua execução; no Capítulo 5 são apresentados os resultados experimentais dessas tarefas; finalmente, no Capítulo 6 é apresentada uma conclusão com base no que foi descrito no trabalho, bem como o que pode ser melhorado em trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo são descritos os principais conceitos e teorias que serviram de embasamento para o desenvolvimento deste trabalho. Bem como a explicação das técnicas empregadas durante o desenvolvimento dos experimentos.

2.1 O mercado financeiro

O mercado financeiro é um sistema complexo que facilita a alocação de recursos financeiros e a negociação de diversos ativos [14]. A classificação dos ativos de uma empresa em categorias distintas é essencial para a compreensão da sua estrutura patrimonial. Nesse sentido, pode-se afirmar que:

> "O ativo circulante abrange os recursos financeiros disponíveis, como o dinheiro em caixa, os saldos em contas bancárias e quaisquer valores que possam ser transformados em dinheiro dentro de um curto período. Por outro lado, o ativo fixo inclui bens como imóveis, equipamentos, utensílios, ferramentas e patentes; itens fundamentais para a continuidade das operações da empresa, que não possuem liquidez imediata."[1]

Outro ponto que marca fortemente este mercado é a volatilidade, uma característica que faz os preços dos ativos financeiros flutuarem significativamente em curtos períodos de tempo, devido a diversos fatores como notícias econômicas, políticas ou eventos inesperados. O mercado financeiro no Brasil é um sistema complexo que desempenha um papel fundamental na economia do país. A Bolsa de Valores de São Paulo, também conhecida como [B][3] (Brasil, Bolsa, Balcão)[2], é a bolsa oficial brasileira, sendo a principal instituição

[1] `https://www12.senado.leg.br/manualdecomunicacao/guia-de-economia/ativo`, acessado em jul, 2024

[2] `http://www.b3.com.br/`, acessado em 2024

de intermediação do mercado de capitais [5], onde empresas de capital aberto listam suas ações para arrecadar capital, permitindo que investidores comprem os ativos.

Notícias do mercado financeiro

São informações e atualizações que reportam acontecimentos, desenvolvimentos econômicos e financeiros, desempenhando um papel crucial na formação de expectativas, na tomada de decisões de investimento e na avaliação do risco, afetando, assim, o movimento das ações e outros ativos. Os investidores e *traders* frequentemente acompanham de perto as notícias financeiras para se manterem atualizados e reagirem às mudanças nas condições do mercado. Geralmente, essas notícias são distribuídas em portais específicos, como SUNO[3], Money Times[4] e InfoMoney[5], redigidas de forma a serem objetivas, com dados precisos e atualizados sobre o mercado financeiro. Nesses portais especializados, o objetivo do título da notícia tende a ser o mesmo: apresentar uma prévia rápida e eficiente sobre o assunto abordado e qual o impacto causado no mercado. Para simplificar a busca, os portais ainda classificam as notícias quanto ao tópico, facilitando a busca de publicações por assunto como, por exemplo, os tópicos "mercado" e "economia" estão entre os mais comuns dentro dos portais sobre o mercado financeiro.

2.2 Processamento de Linguagem Natural

Para a realização da tarefa de classificação de textos é necessário compreender uma série de conceitos e etapas fundamentais no campo do NLP. A seguir, são apresentados os principais fundamentos, técnicas e procedimentos utilizados desde a coleta dos dados até sua preparação e representação para aplicação em modelos de ML [12].

2.2.1 Classificação de Dados Textuais

A classificação de dados textuais é o processo de atribuir rótulos ou classes a textos automaticamente, com base em padrões aprendidos a partir de um conjunto de dados rotulado previamente. Para se realizar a tarefa de classificação dos dados, é necessário primeiramente coletar grandes quantidades de dados textuais. Esses dados são comumente adquiridos por meio de *web scraping* [4], ou seja, através de *scripts* que vasculham sites em busca de determinados tipos de informações. Também é possível montar bases de dados utilizando uma interface de programação de aplicação *Application Programming Interface*

[3] `https://www.suno.com.br/`
[4] `https://www.moneytimes.com.br/`
[5] `https://www.infomoney.com.br/`

(API), uma coleção de código que outros programadores podem utilizar para ajudar no desenvolvimento de tarefas de programação [15], interfaces onde pode-se conseguir os dados por meio de requisições disponibilizadas por mídias sociais [9], sendo uma opção mais simples em relação ao *web scraping* e oferecendo dados com menos ruído e mais acessíveis; ou até mesmo através de bases de dados prontas disponibilizadas *online* por grandes empresas [2], como por exemplo a *Amazon*.

Os dados coletados para a formação das bases de dados deste trabalho contêm ruído, como caracteres especiais, palavras irrelevantes e acentuação, já que foram extraídos diretamente dos portais de notícias do mercado financeiro por meio de *scripts* que buscam automaticamente por dados armazenados nas páginas. Portanto, o texto é pré-processado para gerar um conjunto de dados com menos ruído e que seja melhor analisado pelo modelo de ML aplicado na tarefa de classificação de textos. Este processo consiste em dois passos: normalização das bases de dados e extração das características atreladas a cada amostra de texto que compõe a base de dados.

2.2.2 Técnicas de Normalização Textual

A normalização de textos é um processo fundamental no campo de NLP que envolve a mudança e padronização do texto de maneira a torná-lo mais consistente, limpo e adequado para análise computacional. O objetivo da normalização de textos é lidar com as variações linguísticas, erros tipográficos e outras inconsistências que podem prejudicar a eficácia das tarefas de NLP, como a análise de sentimentos.

Transformações aplicadas ao texto

No procedimento de normalização, são aplicadas transformações no texto. Um exemplo de transformação realizada é a unificação da tipografia para somente caixa baixa ou caixa alta. Desta forma, evitamos que palavras iguais com caixas distintas, como “Carro” e “carro”, sejam tratadas como palavras diferentes pelo algoritmo.

Expressões Regulares

Uma técnica útil para realizar transformações no texto são as Expressões Regulares, do inglês *Regular Expressions* (REGEX), método que surgiu do trabalho do matemático norte-americano Stephen Cole Kleene [16], tendo como objetivo retirar, substituir ou identificar determinadas sequências ou tipos de caracteres, possibilitando que o trabalho de retirar dígitos, pontuações ou caracteres especiais, como acentos gramaticais, seja realizado de forma automática ao longo de toda a base de dados.

Remoção de *stopwords*

As *stopwords* são palavras que não contêm valor relevante para os algoritmos de NLP. Portanto, essas palavras podem ser consideradas como ruído e são preferencialmente removidas dos dados durante a fase de pré-processamento. Alguns exemplos desses tipos de palavras são preposições, conjunções e artigos.

Tokenização

A tokenização é um processo fundamental no campo do NLP que envolve a divisão de um texto em unidades linguísticas menores chamadas *tokens*. Um *token* geralmente corresponde a uma palavra ou a uma parte de uma palavra, podendo ser uma sílaba ou até mesmo um caractere. A tokenização ajuda a transformar um texto contínuo em algo que o computador possa entender e processar de maneira mais eficaz.

Stemização e Lematização

A stemização (*stemming*) e a lematização (*lemmatization*) são técnicas usadas em NLP para reduzir palavras a suas formas base, a fim de simplificar o texto e melhorar a análise. Ambas as técnicas visam transformar diferentes formas de uma palavra em sua forma raiz ou base, mas elas têm abordagens ligeiramente diferentes. A stemização é um processo de corte dos sufixos ou prefixos de uma palavra para extrair sua forma básica ou raiz sem levar em conta o significado final da palavra. Por exemplo, ao se aplicar a transformação em palavras como correr, correndo, correu, obtendo a forma raiz "corr". Por outro lado, a lematização reduz as palavras à sua forma de dicionário ou lema, sendo assim, um processo mais sofisticado e trabalhoso que leva em consideração a semântica e a morfologia das palavras para reduzi-las. Neste caso, a palavra correu é lematizada como correr. A escolha entre stemização e lematização depende das necessidades específicas da tarefa de NLP e da precisão desejada. A realização dos procedimentos descritos anteriormente pode ser conhecida como normalização de textos.

2.2.3 Extração das Características

É comum utilizar representações numéricas para extrair características dos dados textuais. As características extraídas podem incluir informações como a presença e a frequência de palavras-chave, a estrutura gramatical, o contexto das palavras ou a semântica. Este processo desempenha um papel fundamental em várias tarefas de NLP, demonstrado na Figura 2.2, como classificação de texto e análise de sentimentos. A extração de características ajuda a transformar dados de texto não estruturados em representações numéricas

estruturadas que podem ser processadas por técnicas de ML ou outras técnicas de análise de dados. Existem várias abordagens e técnicas para a extração de características em NLP, incluindo representações vetoriais, modelos de linguagem e análise de tópicos. A escolha da técnica de extração de características depende da tarefa específica e dos requisitos do projeto. Um bom exemplo de classificação de textos é a classificação de um conjunto de *emails* em *spam* ou não *spam*, e geralmente é utilizado o método *Bag of Words* (BoW) para se realizar a extração das características dos *emails*.

Bag of Words

Em NLP, *Bag of Words* (BoW) é uma técnica básica para análise de texto e extração de características [12]. Ela representa um documento ou um trecho de texto como uma coleção desordenada de palavras, tratando cada palavra como uma entidade independente sem considerar a ordem, estrutura ou gramática do texto. Os dois tipos mais comuns de BoW são o tipo Contagem e o tipo *Term Frequency - Inverse Document Frequency* (TF-IDF) [12]. No primeiro tipo, apenas a frequência de ocorrência de cada palavra é levada em conta, utilizando-se a Equação 2.1, na qual, t e d são, respectivamente, a palavra e o documento em que a palavra está, e a função *count* retorna o número de ocorrências da palavra no documento [12].

$$TF_{t,d} = count(t, d) \tag{2.1}$$

Já o método TF-IDF, além de realizar o que o tipo Contagem faz, também leva em consideração o inverso da ocorrência dessas palavras em todo o *corpus* (todos os documentos). Ao se realizar o cálculo de uma representação TF-IDF, para cada termo em um documento, multiplicam-se o valor de $\text{TF}_{t,d}$ (frequência de cada termo no documento) pelo valor de IDF_t (inverso da frequência do termo em todos os documentos); como demonstrado na Equação 2.2. Esse cálculo resulta em uma pontuação que indica a importância relativa de cada termo no documento em relação ao *corpus*. Termos que têm um alto valor de TF-IDF em um documento são considerados importantes e específicos para aquele documento [12].

$$TF\text{-}IDF_{t,d} = TF_{t,d} x IDF_t \tag{2.2}$$

Por exemplo, ao se extrair os valores de uma amostra em que foi realizado o processo de normalização; d do *corpus* deste trabalho:

- ***d*** *- projeto preve renegociacao dividas produtores rurais pantanal*

Considerando que foi realizado o método TF-IDF de contagem, são obtidos os valores referentes a cada palavra na Tabela 2.1. É possível observar que os valores estão dentro do intervalo de 0 a 1, e seus valores são referentes às ocorrências das palavras não só nesta amostra, como no *corpus* completo. As palavras com valores mais próximos de 0 são penalizadas por ocorrerem mais vezes ao longo de outras amostras do *corpus*, ao contrário das palavras com valores mais próximos de 1, que ocorrem menos vezes ao longo das demais amostras. Um dos problemas da representação BoW são os vetores esparsos e longos, devido a estas características estarem atreladas ao vocabulário do *corpus*, tornando seu processamento pelos modelos preditivos mais lento; porém, é um modelo direto e simples, tornando-o acessível.

Tabela 2.1: Exemplo de valores TF-IDF referente às palavras

projeto	**preve**	**renegociacao**	**dividas**	**produtores**	**rurais**	**pantanal**
0.231	0.284	0.424	0.354	0.409	0.431	0.453

2.3 Algoritmos de Aprendizado de Máquina

O aprendizado de máquina é um campo interdisciplinar que se desenvolveu ao longo do tempo, combinando conceitos de matemática, estatística, ciência da computação e engenharia. Dentre as possíveis classificações de tipos de algoritmos, existem os métodos supervisionados e os não supervisionados. No método supervisionado, os rótulos associam cada amostra a uma etiqueta que indica a saída desejada. Alguns exemplos de algoritmos supervisionados estão o *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN), *Logistic Regression* (LR) e *K-Nearest-Neighbors* (KNN). Já no método de aprendizado não supervisionado, os algoritmos são treinados com dados que não têm rótulos ou respostas corretas. Esses algoritmos devem aprender padrões e estruturas nos dados por conta própria, os quais não serão abordados neste trabalho.

Os primeiros algoritmos de aprendizado de máquina datam das décadas de 1950 e 1960 [8], em especial a LR, um método de aprendizado supervisionado utilizado até os dias atuais em tarefas de classificação. Na década de 1980, surgiu outro algoritmo de bastante relevância e que é utilizado até o momento, o SVM [17]. Na década seguinte, surgiram os primeiros algoritmos de ANN, modelos inspirados na biologia, consistindo em camadas de neurônios interconectados. Nas duas últimas décadas surgiram metodologias de DL, como as redes neurais recorrentes *Recurrent Neural Networks* (RNN) e as redes neurais convolucionais *Convolutional Neural Networks* (CNN), que consistem em uma especialização de ML destinada a trabalhar com grande quantidade de dados e construir modelos complexos a partir da composição de estruturas mais simples [18]. As CNN per-

mitiram um grande avanço no processamento de imagens, enquanto as RNN são bastante utilizadas para lidar com dados sequenciais como séries temporais e texto [19].

2.3.1 Naive Bayes

NB é um algoritmo de classificação probabilística baseado no Teorema de Bayes, um conceito da teoria da probabilidade [7], sendo utilizado em aprendizado de máquina para tarefas de classificação, como categorização de texto, filtragem de spam e diagnóstico médico [7]. Esse método descreve a probabilidade de um evento ocorrer baseando-se em suas ocorrências anteriores. Em tarefas de classificação, esse algoritmo é usado para calcular a probabilidade de uma amostra pertencer a uma determinada classe com base nas probabilidades das características observadas. A suposição fundamental do NB é a independência condicional ingênua. Isso significa que o algoritmo assume que as características usadas para a classificação são independentes entre si, embora essa suposição nem sempre seja verdadeira na prática [7]. Existem diferentes variantes do NB, sendo os modelos Multinomial e Gaussiano os mais frequentemente usados [7]. O NB Multinomial é frequentemente usado [7] para dados de contagem, como palavras em documentos de texto; portanto, foi o método adotado durante os experimentos deste trabalho.

2.3.2 K-Nearest Neighbors

O KNN é um dos algoritmos mais simples e intuitivos em aprendizado de máquina [8]. No entanto, seu desempenho pode ser afetado pela dimensionalidade dos dados e pelo tamanho do conjunto de treinamento. Além disso, a seleção de uma métrica de distância apropriada é importante para obter resultados significativos. Ou seja, é importante realizar testes e ajustes finos nos parâmetros do algoritmo para se obter os melhores resultados para cada base de dados ou tipos de dados a serem categorizados ou previstos. O parâmetro k no KNN representa o número de vizinhos mais próximos que serão considerados ao tomar uma decisão de classificação ou regressão. Por exemplo, se $k = 3$, o algoritmo considerará os três vizinhos mais próximos para fazer uma decisão, sendo considerada a distância euclidiana, uma vez que os vizinhos se encontram no espaço euclidiano [8]. A distância pode ser calculada pela Equação 2.3, para realizar os cálculos de proximidade, onde P é a distância no espaço vetorial, e p_i, q_i são as coordenadas dos pontos em cada dimensão, e n é o número de dimensões do espaço vetorial considerado.

$$P = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.3}$$

[5] `https://builtin.com/machine-learning/deep-learning`, acessado em jul, 2024

2.3.3 *Support Vector Machine*

SVM é um algoritmo de ML amplamente utilizado para tarefas de classificação e regressão. Proposto por Cortes e Vapnik [17], o SVM busca encontrar um hiperplano, região dentro do espaço vetorial com uma dimensão a menos que o número de dimensões do espaço, que melhor separa as classes de um conjunto de dados. Através dessa separação, é realizada a classificação; inicialmente foi desenvolvido para classificação binária, mas posteriormente foi expandido para lidar com múltiplas classes e problemas de regressão. Os *Kernels* são funções que transformam os dados pré-processados em um espaço de características dentro do espaço vetorial, permitindo a separação de dados linearmente e não linearmente dependentes. É um algoritmo de ML eficiente em espaços de altas dimensões, e se adapta bem a conjuntos de dados pequenos [17].

2.3.4 Redes Neurais Artificiais

O desenvolvimento das ANN começou com a proposta de um modelo matemático simples baseado no neurônio biológico por McCulloch e Pitts em 1943 [1], estando representado na Figura 2.1.

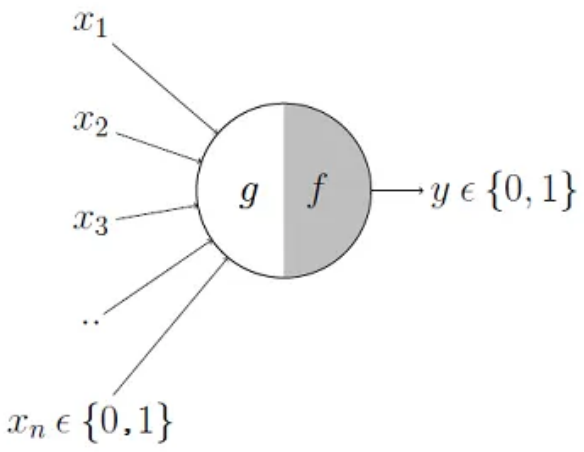


Figura 2.1: Representação de neurônio de McCulloch e Pitts [1].

Desde então, o campo recebeu diversas mudanças e melhorias para suprir limitações dos algoritmos, já que estavam limitados a tarefas mais simples e tarefas lineares [1]. As ANNs são compostas por unidades interconectadas chamadas neurônios artificiais ou perceptrons, organizadas em camadas. O funcionamento dessas redes é baseado na transmissão de sinais entre essas camadas de neurônios, onde cada conexão tem um peso associado que determina a importância do sinal transmitido. Esse tipo de abordagem

é usado para diversas tarefas de aprendizado de máquina, como classificação, regressão, reconhecimento de padrões e processamento de linguagem natural. No treinamento, os pesos das conexões são ajustados para que a ANN mapeie entradas para uma das saídas de maneira eficaz com base em dados de treinamento. Existem diferentes arquiteturas de ANN, incluindo redes de alimentação direta e redes recorrentes. As CNN e as RNN [18] são exemplos de redes baseadas em DL. Como representado na Figura 2.2, os algoritmos de DL fazem a extração das características do texto de forma automática, otimizando a entrada dos dados para o algoritmo a ser utilizado.

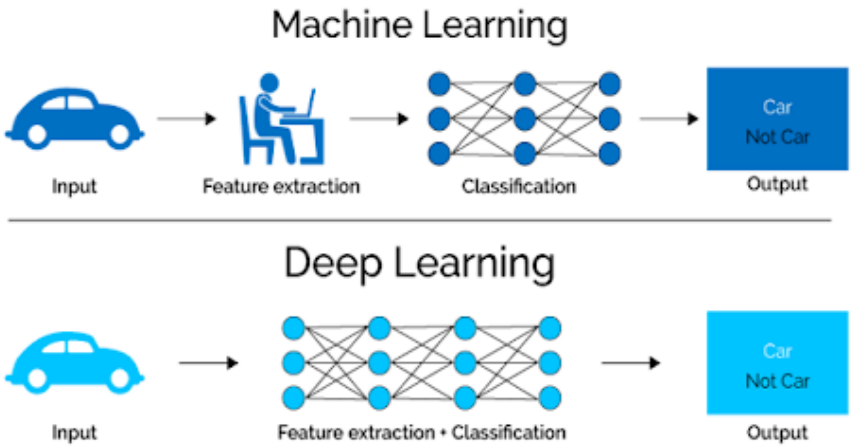


Figura 2.2: Diferença de implementação entre ML e DL[6]

As CNNs são projetadas para processar dados de grade, como imagens e vídeos. Elas usam camadas de convolução para detectar padrões locais em dados de entrada, tornando-as eficazes na extração de recursos visuais. Por outro lado, as RNNs, representadas na Figura 2.3 onde cada parte representa um neurônio com seu determinado estado, possuem uma arquitetura que permite que as informações permaneçam em estados cíclicos, possibilitando que as redes mantenham uma memória de estados anteriores. RNNs são adequadas para tarefas que dependem de contextos sequenciais, como o NLP.

2.3.5 *Transformers*

Um avanço em relação às redes citadas anteriormente são os *Transformers*, modelos de ANN que revolucionaram o campo de NLP. Esses modelos são baseados em uma arquitetura de rede neural chamada *attention* [20], que permite que eles capturem relações de dependência entre palavras em uma sentença, levando em consideração o contexto global, em vez de depender de uma janela fixa de palavras, como as RNN.

[6] `https://medium.com/levity/deep-learning-vs-machine-learning-whats-the-difference-e367803bb96d`, acessado em julho, 2024

Attention-Layer

A camada *attention* veio para resolver alguns problemas relacionados às RNNs, dentre eles o problema das dependências de longo prazo. Existem casos em que é necessário mais contexto, por exemplo, uma RNN está tentando descobrir a próxima palavra na seguinte situação: "Eu cresci na França...(continuação da frase até o seguinte excerto). Eu falo fluentemente...". Através das informações dadas, supõe-se que a próxima palavra seria um idioma, mas para se determinar qual idioma é necessário o contexto informando a França, que está mais adiante no texto. As RNN acabam sendo ineficazes quando a distância entre a informação e o ponto em que ela é necessária se torna muito grande. Isso ocorre porque a informação é passada por cada etapa da NN, e quanto mais longa for a cadeia de recorrência, maior a probabilidade de perder informações ao longo da cadeia. Então, para tentar suprir essa dificuldade encontrada nas RNN surgiram as *Long Short-Term Memory* (LSTM) [20], esse método faz algumas operações com as informações ao longo dos ciclos, desta forma as LSTM podem se "lembrar" ou "esquecer" de informações de acordo com a necessidade. Porém, mesmo sendo melhores que as RNN em casos de frases muito longas, ainda ocorrem problemas com as LSTM, ou seja, o modelo muitas vezes esquece o conteúdo de posições distantes na sequência. Além da situação citada, as RNN e as LSTM não permitem a paralelização do processamento, uma vez que é necessário processar palavra por palavra.

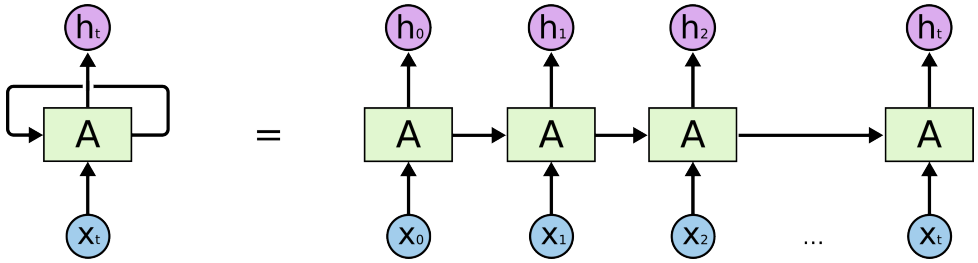


Figura 2.3: Representação de uma Rede Neural Recorrente[7]

A camada *attention*, representada na Figura 2.4 [21] a qual demonstra o funcionamento interno desta camada, faz uma transformação de palavra em palavra da frase de entrada, o que é passado para as demais camadas da RNN, para então se decodificarem os dados. Desta forma, informações relevantes que podem estar presentes em cada palavra de uma sentença são passadas adiante. Consequentemente, para que a decodificação seja eficiente, é preciso levar em conta cada palavra de entrada. Com isso, a camada de *attention* previne que informações distantes sejam perdidas no processamento dos dados. Então, para se resolver o problema de realizar o processamento em paralelo, os *trans-*

[7] `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, acessado em julho, 2024

formers surgiram. Com a possibilidade de processar paralelamente os dados de grandes *corpus*, os *transformers* também implementam uma camada de *self-attention* [20] (Figura 2.5), ou seja, nesta camada o algoritmo compara todas as sequências de entradas, e individualmente cada item aos outros, então modifica as saídas. Permitindo que o máximo de informações seja repassado para camadas posteriores do método.

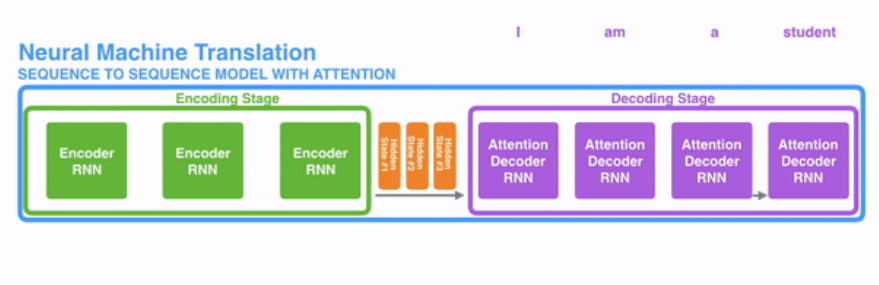


Figura 2.4: Como uma camada de *attention* funciona[8]

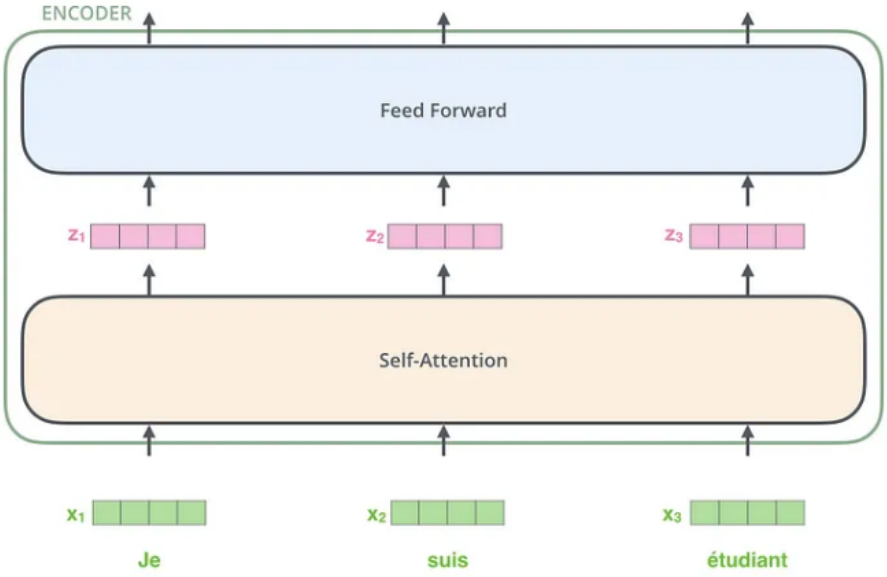


Figura 2.5: Como um encoder da camada de *self-attention* é representado[9]

Os *Transformers* são compostos por várias camadas, geralmente chamadas *transformer layers*, representadas na Figura 2.6, que processam a entrada em paralelo, tornando-os altamente paralelizáveis e eficientes para treinamento [20] em *hardware* moderno, como

[8] `https://towardsdatascience.com/transformers-141e32e69591`, acessado em julho, 2024

[9] `https://medium.com/@mail2rajivgopinath/part-4-the-comprehender-transformer-encoders-of-the-series-from-sequences-to-sentience-35d24e386158`, acessado em julho, 2024

unidade de processamento gráfico *Graphical Processing Unit* (GPU) ou unidade de processamento tensor *Tensor Processing Unit* (TPU). Além disso, eles podem ser pré-treinados em grandes conjuntos de dados com tarefas de linguagem, como o modelo *Bidirectional Encoder Representations from Transformers* (BERT).

BERT

O modelo BERT, desenvolvido pelo Google em 2019 [9], se destacou por sua capacidade de compreender o contexto das palavras em uma sentença ou texto, capturando relações complexas em linguagem natural. Esse modelo se sobressai pela capacidade de processar texto bidirecionalmente, ou seja, considera o contexto das palavras à esquerda e à direita de uma palavra específica ao analisar uma amostra. Isso permite que o modelo entenda melhor o significado de uma palavra com base em seu contexto circundante, tornando-o altamente eficaz para realizar o NLP. O BERT é classificado como um *Large Language Model* (LLM), ou seja, modelos desenvolvidos para compreender e gerar linguagem humana com alto grau de sofisticação. Esses modelos são treinados com grandes volumes de texto provenientes de diversas fontes, como livros, artigos, sites e outros documentos disponíveis na internet, e utilizam arquiteturas avançadas de redes neurais, especialmente os *transformers*, como base de seu funcionamento [22]. Outro benefício do BERT é a possibilidade de utilizar seu treinamento em grandes volumes de texto como base para refinar modelos em tarefas específicas, como classificação, tornando-se uma base sólida para uma ampla variedade de aplicações de NLP. Desde sua introdução, esse modelo se tornou o estado da arte em muitas tarefas [6], impulsionando significativamente o desempenho em várias áreas da IA.

2.4 Métricas de Desempenho

Para avaliar o quão bem os modelos de previsão funcionam em tarefas de classificação, como será visto na Seção 4.6, são utilizadas diversas métricas. Através dos resultados obtidos durante as classificações, é possível extrair uma matriz de confusão (MC, *Confusion Matrix*) (CM), que compara as previsões feitas por um modelo de ML com as classificações corretas de um conjunto de dados de teste. Na representação de uma CM, o somatório dos valores atribuídos pela saída do algoritmo de classificação é classificado em quatro tipos diferentes.

- Verdadeiro Positivo (VP): quando a amostra é classificada corretamente quanto ao rótulo ou valor;

[10] `https://towardsdatascience.com/transformers-141e32e69591`, acessado em julho, 2024

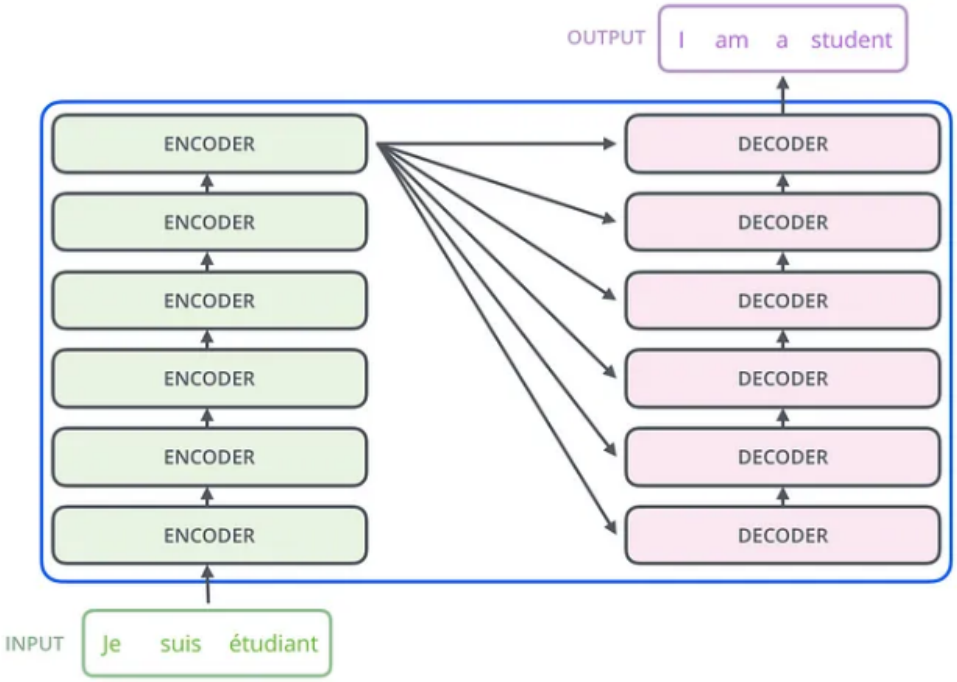


Figura 2.6: Representação de como um *transformer* processa os dados[10]

- Verdadeiro Negativo (VN): quando a amostra não é classificada como determinado rótulo de forma correta, já que é associada a outro rótulo;
- Falso Positivo (FP): quando a amostra é classificada de forma errada com um rótulo, porém pertence a outro rótulo;
- Falso Negativo (FN): quando a amostra deixa de ser classificada com um rótulo, mas este rótulo seria o correto.

A matriz de confusão da Figura 2.7 fornece uma visão detalhada do desempenho do modelo de classificação, permitindo a análise de quais tipos de erros estão sendo cometidos e como o modelo está se saindo em relação aos verdadeiros positivos e negativos. Com base nestes quatro valores, várias métricas de desempenho podem ser calculadas, incluindo Acurácia, Precisão, Revocação e F1-Score.

Acurácia

A acurácia é uma métrica de avaliação que reflete a proporção de previsões corretas em relação ao número total de previsões. No entanto, normalmente, ela não é usada isoladamente , pois pode dar uma impressão enganosa do desempenho real de um modelo preditivo. Por exemplo, ao se implementar um exemplo clássico de classificação de texto,

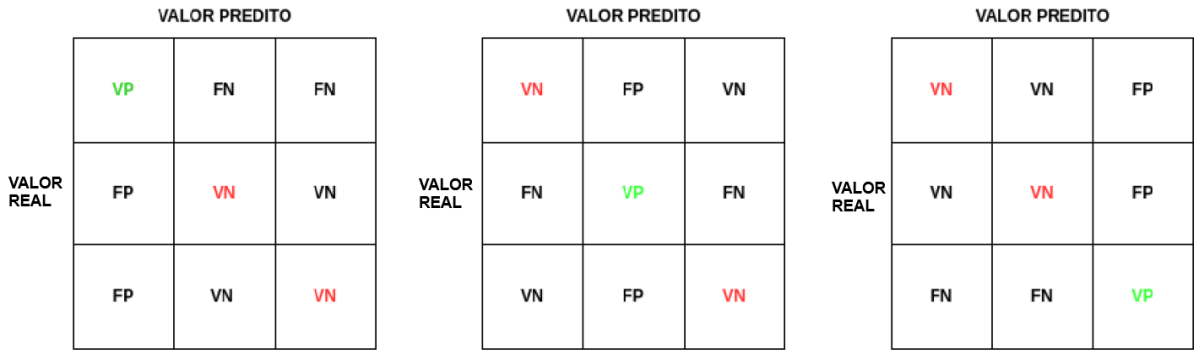


Figura 2.7: Representação da uma Matriz de Confusão para três classes (fonte: Souza [23])

para definir se e-mails são spam ou não, o modelo pode rotular todos os e-mails como spam. Tendo em vista que 90% dos e-mails de teste são spam, o modelo terá uma acurácia de 90%, embora não tenha feito nenhuma classificação correta. Por esse motivo, outras métricas são frequentemente apresentadas e usadas em conjunto para mitigar essa limitação. A acurácia pode ser calculada pela seguinte equação:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.4}$$

Precisão

Nesta métrica é medida a relação entre as predições positivas corretas e o total de predições positivas, corretas ou não, mostrando o quanto o algoritmo é preciso em classificar positivamente, levando a um resultado com menos falsos positivos. Pode-se calcular a precisão pela equação:

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.5}$$

Revocação

A revocação relaciona as predições positivas corretas e o total de positivos corretos. Desta forma, quanto maior o valor da revocação, menor a quantidade de falsos negativos. A revocação pode ser calculada pela equação:

$$\text{Revocação} = \frac{VP}{VP + FN} \tag{2.6}$$

F1-Score

Quando apenas a métrica de precisão é avaliada, destaca-se o desempenho do modelo em relação aos falsos positivos. Enquanto avaliar apenas a revocação destaca o desempenho do modelo em relação aos falsos negativos. Ao se combinar ambas as métricas, obtém-se a métrica F1-Score, que tem o objetivo de unir os pontos positivos da precisão e da revocação. A F1-Score está representada na seguinte equação:

$$\text{F1-Score} = \frac{2 \times \text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{2.7}$$

Capítulo 3

Trabalhos Relacionados

Neste capítulo, são abordados estudos propostos na literatura que motivaram este trabalho, destacando os pontos que foram relevantes e serviram como base para os experimentos. A partir da revisão de estudos, também foi possível identificar melhorias e metodologias diferentes que puderam ser consideradas neste trabalho. Os estudos que são apresentados se referem à área de NLP e têm aplicações desde os algoritmos de ML tradicionais até os mais recentes, tais como os *Transformers*.

Na literatura existem diversas maneiras de classificar textos desde a aplicação de algoritmos tradicionais, como NB e KNN aos mais recentes estado da arte, tais como os *Transformers*. Geralmente, os dados utilizados na literatura são críticas de filmes, detecção de *fake-news*, classificação de e-mails como spam ou não spam. Esses exemplos citados foram revisitados com o intuito de embasar a tarefa de classificação de texto realizada neste trabalho. Através da pesquisa de Schmitz *et al.* [4], pode-se estabelecer uma base de dados com notícias do mercado financeiro estruturada em português e rotulada em tópicos, palavras utilizadas para separar notícias nos portais de notícias do mercado financeiro.

O estudo de Wang *et al.* [24] explora a classificação de patentes *FinTech*s chinesas com uso de modelos como CNN e BERT, com o objetivo de identificar tendências tecnológicas. Um dos grandes méritos do trabalho é a construção de uma base de dados anotada manualmente, permitindo treinar modelos robustos. Foi utilizada a análise sistemática da inovação em *FinTech* com aplicação direta à economia chinesa. No entanto, essa anotação manual mostrou ter alto custo, além do foco geográfico ser restrito à China.

O artigo de Kamal *et al.* [11] aplica técnicas de NLP para prever movimentos no mercado de ações com base em manchetes de notícias financeiras. Modelos como CNN, LSTM, BERT e RoBERTa foram testados, com destaque para o BERT que obteve 90% de acurácia e o RoBERTa com 88%. O estudo mostrou a abrangência dos modelos testados e a eficácia dos *transformers*. A base de dados utilizou apenas manchetes, sem aproveitar

textos completos ou outras fontes.

A pesquisa de Karanikola *et al.* [25] realizam uma comparação abrangente entre métodos clássicos de ML e modelos baseados em DL para análise de sentimento financeiro. *Robustly Optimized Bidirectional Encoder Representations from Transformers Aproach* (RoBERTa) e *Financial Bidirectional Encoder Representations from Transformers* (FinBERT) foram testados, com o primeiro superando os demais, obtendo a medida de *F1-score* de 0.88. Houve uma ampla variedade de modelos comparados. Contudo, a performance abaixo do esperado do FinBERT levanta questionamentos sobre sua efetividade no domínio que supostamente domina.

Cunha *et al.* [13] realizam um apanhado de comparações em relação ao custo-benefício entre algoritmos de ML tradicionais e algoritmos baseados em NN. Nesse estudo de Cunha *et al.* foram analisados mais de 1500 experimentos, contando com 14 métodos diferentes e em 9 *datasets* distintos. Assim como alguns estudos apresentados anteriormente, foi utilizado o método TF-IDF para representar os dados textuais de forma numérica; os *datasets*, em sua maioria, são formados por notícias de diversos assuntos. Nesse trabalho, foram levadas em conta as métricas *Macro F1-Socre* e *Micro F1-Score*.

O artigo de Jaggi *et al.* [26] propõe o modelo *Financial A Lite Bidirectional Encoder Representations from Transformers* (FinALBERT), uma adaptação leve do ALBERT pré-treinado em textos financeiros, com foco na previsão de preços de ações a partir de dados do *StockTwits*. A inovação metodológica inclui a rotulagem automática com base na variação real do preço de ações. FinALBERT combinado com *Bidirectional Long Short-Term Memory* (Bi-LSTM) apresentou excelente desempenho. No entanto, vale destacar a necessidade de múltiplas fontes de dados e alto custo computacional para o pré-treinamento do modelo.

A pesquisa de Blümel e Zaki [27] foca na priorização automática de reclamações de clientes usando NLP. A pesquisa testa métodos como TF-IDF, *Word2Vec*, *Random Forest*, CNN e BERT, sendo que a melhor performance foi alcançada pela combinação de TF-IDF com BERT e regressão logística. O trabalho mostrou a praticidade da solução e sua aplicabilidade direta em ambientes. Contudo, a área de estudo não está relacionada ao mercado financeiro.

O trabalho de Sy *et al.* [28] utiliza BERT, RoBERTa e *A Lite Bidirectional Encoder Representations from Transformers* (ALBERT) para classificar respostas qualitativas de estudantes sobre o programa de acesso gratuito ao ensino superior nas Filipinas. O algoritmo RoBERTa teve o melhor desempenho. Tendo destaque na aplicação real em políticas públicas e na participação de especialistas na definição das categorias; porém, a aplicabilidade é restrita ao contexto educacional filipino, o que pode dificultar a generalização dos resultados.

Kalra e Prasad [29] conduzem experimentos com objetivos semelhantes aos de Sousa *et al.* [10], pois também buscam evidenciar a hipótese de que o sentimento expresso em notícias possui uma correlação significativa com a movimentação do mercado financeiro. Foram extraídas notícias relacionadas a essas ações para análise de sentimento. O pré-processamento dos dados textuais envolveu diversas etapas, como tokenização, conversão para letras minúsculas, remoção de stopwords e transformação em uma representação BoW baseada no modelo TF-IDF. Os experimentos envolveram a aplicação de diferentes modelos de aprendizado de máquina, incluindo KNN, NB, SVM e ANN. Para alimentar esses modelos, foi utilizado exclusivamente o atributo correspondente ao título de cada notícia. A avaliação do desempenho dos modelos foi realizada utilizando a abordagem *Holdout*, com 70% dos dados destinados ao treinamento e os 30% restantes para teste. A análise dos resultados foi conduzida com base em quatro métricas de desempenho, conforme abordado também nos estudos de Sousa *et al.* [10]. Os resultados obtidos demonstram que o modelo KNN apresentou o melhor desempenho em dois dos quatro bancos analisados, alcançando taxas de acurácia de 85% e 91%, respectivamente. Para os outros dois bancos, os modelos NB e SVM se destacaram, obtendo acurácias de 80% e 81%, respectivamente. Esses achados evidenciam a relevância da escolha do modelo adequado para cada contexto e reforçam a relação entre o sentimento expresso nas notícias.

Após a análise da literatura, foi possível perceber que os resultados obtidos pela comparação direta entre algoritmos de ML tradicionais e os recentes *Transformers* , os algoritmos baseados em DL apresentam resultados ligeiramente melhores, como pode ser verificado na Tabela 3.1. Portanto, este trabalho aplica a implementação de dois modelos de ML tradicionais, NB e KNN, um modelo de DL SVM além de um *ensemble* dos três modelos mencionados anteriormente, e compara os resultados e métricas da implementação e treinamento de um dos modelos do estado da arte, o BERT, buscando mensurar o quanto melhor os algoritmos baseados em DL são em relação aos algoritmos de ML tradicionais.

Tabela 3.1: Revisão de Literatura

Autores	**Ano**	**Tarefa**	**Métodos**	**Resultados**
Sy *et al.* [28]	2024	Classificar respostas qualitativas sobre o programa UAQTE	BERT, RoBERTa, ALBERT	*Acurácia: 87,2%, 85,5% e 84,7%*
Karanikola *et al.* [25]	2023	Comparar modelos clássicos e DL na análise de sentimentos financeiros	SVM, FinBERT, RoBERTa	*F1 Score: 0.88 , 0.84 e 0.78*
Blümel and Zaki *et al.* [27]	2022	Automatizar a priorização de reclamações de clientes com NLP	RF, BERT + TF-IDF + Logistic Regression	*F1 Score: 0.82 e 0.76*
Wang *et al.* [24]	2022	Classificar patentes FinTech chinesas e identificar tendências tecnológicas	KNN, NB, SVM, CNN, BERT	*Acurácia: 61%, 53%, 72%, 84%, e 83%*
Kamal *et al.* [11]	2022	Analisar impacto de notícias financeiras nas ações com NLP	LSTM, BERT, RoBERTa	*Acurácia: 84%, 88% e 90%*
Schmitz *et al.* [4]	2021	Classificar textos curtos derivados de múltiplas fontes	Bi-LSTM	*F1 Score: 0.91 Acurácia:84,8%*
Cunha *et al.* [13]	2021	Comparar os métodos	CNN, LSTM, BERT	Macro F1: 0.55%, 0.77%, e 0.85%
Jaggi *et al.* [26]	2021	Prever ações com modelo FinALBERT e rotulagem baseada em variação	FinALBERT + BiLSTM, FinBERT, BERT	*F1 Score: 0.90, 0.88 e 0.86%*
Kalra e Prasad [29]	2019	Classificar texto multiclasses	KNN, SVM	*Acurácia: 63%, 81%*

Capítulo 4

Metodologia

Neste capítulo é explicado de maneira mais profunda como foi realizado o estudo abordado neste trabalho: o que foi proposto para se realizar uma comparação do desempenho entre algoritmos tradicionais de ML e os mais recentes modelos *Transformers*. Também será explicado qual base de dados foi utilizada nos experimentos, os métodos de pré-processamento de texto, quais algoritmos e métricas analisadas, e as especificidades dos métodos aplicados.

4.1 Classificação de Títulos de Notícias entre os Tópicos mais Recorrentes

A metodologia de comparação realizada neste trabalho é composta de quatro etapas que estão descritas a seguir (Figura 4.1): (i) **Coleta e organização das bases de dados**: etapa responsável por obter grandes massas de dados, organizar de forma a se montar os *datasets* para que possam ser utilizados como bases para o treinamento dos algoritmos utilizados nos experimentos; (ii) **Pré-processamento do texto**: etapa em que os dados são modificados para reduzir o ruído, processo denominado normalização dos dados, seguindo para que os dados sejam formatados e tenham uma transformação de tipo para serem utilizados no treinamento dos algoritmos de ML, etapa conhecida como extração das características; (iii) **Treinamento dos algoritmos**: etapa onde os dados são processados pelos modelos de ML, utilizados no treinamento e na otimização dos hiperparâmetros para cada algoritmo utilizado e então são aplicados ao conjunto de dados de teste buscando verificar os resultados obtidos. Foram utilizados três modelos de ML tradicionais, um *ensemble* de votação utilizando os modelos de ML tradicionais e um modelo baseado em *transformers*; (iv) **Avaliação dos classificadores**: etapa na qual, através das classificações realizadas nos *datasets*, são calculados os valores das diferentes

métricas de avaliação dos modelos, que servem para realizar a comparação de desempenho entre os algoritmos implementados.

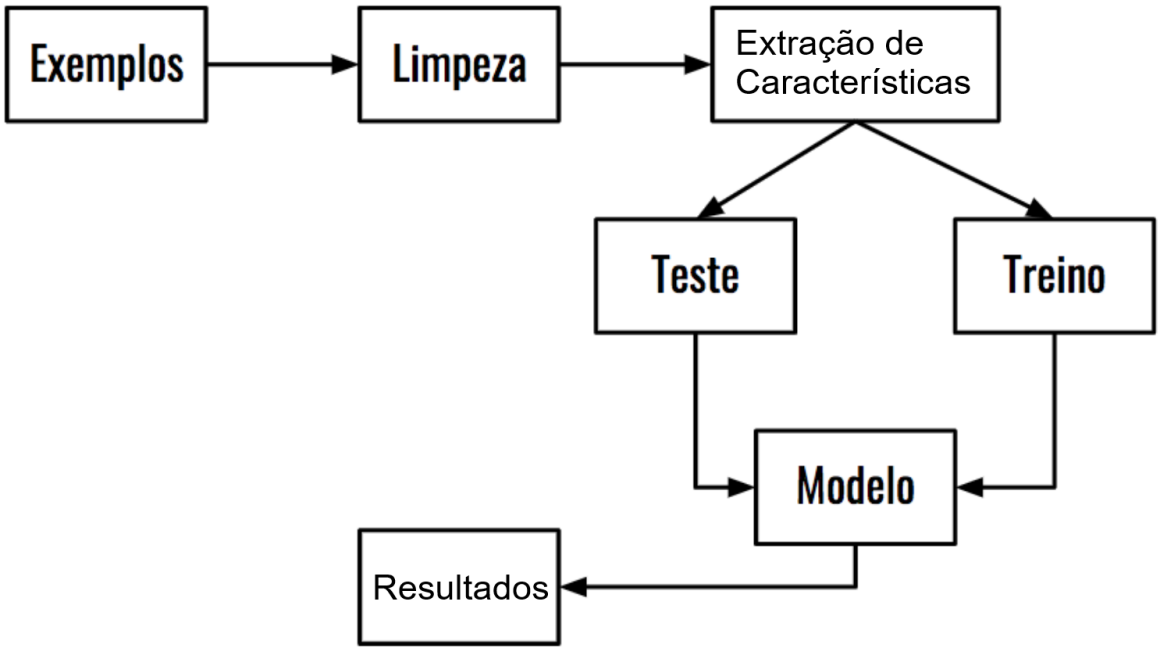


Figura 4.1: Pipeline da Classificação de Textos.

É importante salientar que para a etapa de extração de características, exposta no Capítulo 2, foram utilizados dois modelos distintos de métodos a depender do algoritmo de ML, para algoritmos tradicionais e o modelo de DL foi utilizado o TF-IDF já que foi o mais utilizado na literatura [19]. Já para o BERT foi utilizado o *Bert-tokenizer* devido à sua melhor integração com o modelo baseado em *Transformers*. Já na etapa de otimização, as abordagens são diferentes, na qual os algoritmos tradicionais não necessitam de tantos parâmetros a serem testados [7], já nos algoritmos baseados em *Transformers*, existem muitos parâmetros e hiperparâmetros que mudam completamente o tempo de teste e o resultado de cada experimento [2].

4.2 Bases de Dados

Para se realizar a classificação dos títulos, foram definidos os quatro tópicos mais recorrentes, devido à maior quantidade de amostras, com intuito de manter o número de amostras equilibrado e utilizando apenas dados provindos de portais de notícias. Foi gerado um *dataset* contendo apenas títulos classificados nos quatro tópicos. Cada notícia é atrelada a apenas um tópico, desta forma, os algoritmos também atribuem apenas um rótulo a cada amostra. A classificação foi realizada dentro dos tópicos.

As amostras de dados utilizadas neste trabalho são qualitativas, ou seja, são dados textuais extraídos de sites de notícias sobre o mercado financeiro. Para o processo de extração dos dados foram utilizados *scripts* do tipo *crawler*, os quais vasculharam páginas *web* salvando os dados textuais de determinados campos em formato *JavaScript Object Notation* (JSON), formando uma base de dados semi-estruturada. Em cada amostra estão presentes o título, tópico, texto completo, *tags*, url de acesso e data de publicação da notícia (Figura 4.2). A base em questão foi montada utilizando *scripts* que extraíram as informações presentes nos sites Suno Research[1], Money Times[2] e Infomoney[3]. Ao realizar a contagem de amostras de um dos *datasets* a diferença entre a quarta classe e a quinta classe com mais ocorrências é grande ao ponto de desbalancear a base de dados. Visando continuar uma base de dados bem balanceada e buscando verificar o comportamento dos algoritmos com bases não balanceadas, os *datasets* foram montados utilizando as quatro classes mais recorrentes de cada um dos portais de notícias mencionados. Em seguida, foi realizada a distribuição das amostras restantes em *datasets* de treinamento e de teste, sendo a proporção de 80% e 20%, respectivamente, para os algoritmos tradicionais; o conjunto de treinamento foi também utilizado para ajuste de hiperparametros por meio da estratégia de validação cruzada. Para o BERT, foi considerada a distribuição 80% amostras para o treinamento e 20% para o teste final do modelo; do conjunto de treinamento, 20% foram reservados para a validação durante o ajuste de hiperparâmetros.


```
"topic": "Negócios",
"title": "Ibovespa: Petrorio (PRIO3) lidera ganhos da semana; fusão de BRF(BRFS3)
  e Marfrig (MRFG3) anima mercado",
"date": "02/10/2021 17:05",
"full_text": "O Ibovespa inaugurou bem o mês de outubro, fechou em forte alta seu
  primeiro pregão. (...)",
"url": "https://www.suno.com.br/noticias/ibovespa-petrorio-prio3-brf3-marfrig/",
"tags": [
  "Bradesco (BBDC3/BBDC4)",
  "BRF (BRFS3)",
  "Ibovespa",
  "Itaú Unibanco (ITUB3/ITUB4)",
  "Marfrig (MRFG3)",
  "PetroRio (PRIO3)"
]
```


Figura 4.2: Amostra de texto extraído do portal.

[1] https://www.suno.com.br/
[2] https://www.moneytimes.com.br/
[3] https://www.infomoney.com.br/

As bases de dados que foram utilizadas neste trabalho podem ser reproduzidas utilizando os dados disponibilizados em repositório público[4].

4.3 Pré-processamento

Após realizar a divisão das bases de dados, os procedimentos de transformação do texto são cruciais para que os algoritmos aplicados tenham o melhor desempenho, o *dataset* não pode conter caracteres que prejudicam a interpretação como apresentado na Seção 2.2.2, e apresentando pouca variação quanto ao padrão textual; os algoritmos ainda apresentam dificuldade para processar os *datasets* sem um pré-processamento.

Primeiramente foram retirados os dados que não seriam utilizados, deixando somente os dados das colunas tópicos e títulos. As amostras apresentam caracteres como parênteses, acentos gramaticais, pontuações e variações de caixa, então foi necessário retirar tais características do texto e deixar todo o texto em caixa baixa. Outro processo no pré-processamento do texto é a remoção de *stop words*, que são palavras utilizadas gramaticalmente para dar uma quebra no ritmo de leitura e dar um melhor entendimento ao leitor; porém essas palavras não são relevantes para os modelos implementados, então também são retiradas durante a etapa de pré-processamento. Por exemplo, o título presente na Figura 4.3, após pré-processado, ficaria da seguinte forma:

- **Título não processado:** Embraer (EMBR3): Aérea gringa faz nova rodada de encomendas; veja detalhes
- **Título processado:** embraer embr3 aerea gringa faz nova rodada de encomendas

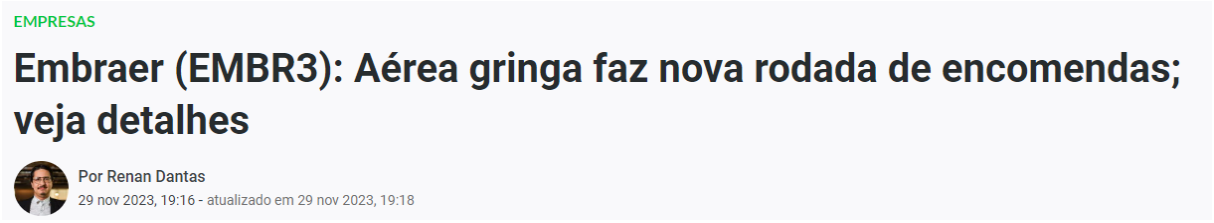


Figura 4.3: Exemplo de título de notícia extraído do portal de notícias Suno Research.

Um exemplo de amostra de um dos *datasets* pode ser visualizado na Tabela 4.1. Mantendo-se as mesmas amostras como entrada para todos os algoritmos aplicados. A contagem de amostras pode ser visualizada nas Tabelas 4.2a, 4.2b e 4.2c. O *dataset* do portal Moneytimes é bem balanceado; já os outros dois *datasets* não estão balanceados, permitindo que os algoritmos sejam testados em diferentes condições.

[4] `https://github.com/mso13/SnorkelTextClassification/`

Tabela 4.1: Amostra de um *dataset*

Tópico	**Título**
mercados	small caps botaram ibovespa no bolso durante o(...)
empresas	a disputa entre equatorial eqtl3 e cpfl cpfe3 (...)
empresas	spotify atualiza precos veja reajustes e qual (...)
mercados	de volta aos r 70 vale vale3 sobe forte e empu(...)
mercados	a fala de haddad que derruba as acoes de itau (...)

Tabela 4.2: Quantidade de amostras dos *datasets*

Tópico	Total
empresas	19511
politica	17612
mercados	17243
internacional	16418

(a) Moneytimes

Tópico	Total
mercados	36197
politica	14631
economia	9732
ondeinvestir	7604

(b) InfoMoney

Tópico	Total
negocios	25274
mercado	8222
economia	4328
internacional	3144

(c) Suno

4.4 Extração de Características do Texto

Nesta etapa foram utilizados dois métodos distintos a se depender do tipo de algoritmo de classificado a ser utilizado. A extração das características do texto é realizada em todos os *datasets*, englobando todas as amostras utilizadas no treinamento e teste. Durante a extração de características para os algoritmos de ML tradicionais, foi utilizada a técnica TF-IDF. Foram passadas as amostras de texto separadas para o algoritmo *count_vectorizer* da biblioteca **Scikit-learn**, e como resultado deste processo obteve-se a representação numérica de cada amostra. A partir desses valores numéricos foram criados os *n_grams* de tamanho um, ou seja, cada palavra fica associada a apenas um valor.

Já para o algoritmo BERT, foi utilizada uma função de tokenização para realizar a extração das características dos *datasets* para o algoritmo de DL BERT. Este método utiliza algo conhecido como tokenização baseada em subpalavras (Seção 2.2.3). O BERT utiliza o algoritmo *WordPiece* para gerar o vocabulário, no qual subpalavras são geradas com base na probabilidade de caracteres ocorrerem juntos. Mas, ao contrário do TF-IDF, a função de tokenização Figura 4.4 é aplicada em todas as amostras de todos os *datasets*. Esse método divide o texto em tokens e adiciona tokens especiais, como *CLS* (representando o início da sequência) e *SEP* (indicando separação entre sequências em tarefas específicas). Além disso, o BERT utiliza uma técnica chamada tokenização de subpalavras, que divide palavras em partes menores, especialmente útil para lidar com palavras desconhecidas ou palavras que não estão presentes no vocabulário do modelo.

4.5 Treinamento dos Algoritmos

Após toda a preparação dos dados, foram realizados os treinamentos dos algoritmos de ML apresentados na Seção 2.3, sendo quatro algoritmos tradicionais e um algoritmo de DL. Como mencionado anteriormente, para manter os experimentos sem viés, foram mantidos os mesmos *datasets* e também os mesmos conjuntos de treinamento e teste. Então, com os resultados obtidos, foi possível analisar a qualidade da classificação a partir dos algoritmos selecionados.

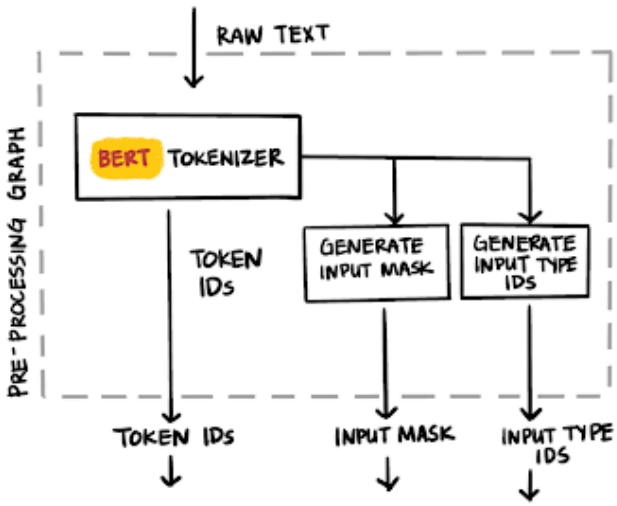


Figura 4.4: Pipeline da função de tokenização do BERT [2].

Algoritmos Tradicionais

Foram aplicados os algoritmos NB [7], KNN [8] e SVM [17] além de um *ensemble* [19] do tipo votação por maioria, combinando os três algoritmos individuais mencionados. E então foi necessário realizar a otimização dos algoritmos, tarefa realizada automaticamente utilizando a biblioteca **GridSerachCV**. Seu funcionamento baseia-se na criação de uma grade de valores possíveis para cada hiperparâmetro. A partir dessa grade, o algoritmo testa todas as combinações possíveis, treinando o modelo e avaliando seu desempenho para cada uma delas. A avaliação é feita por meio de *Cross-validation* a patir do conjunto de treino, o que permite uma estimativa mais robusta da performance do modelo ao reduzir o risco de *overfitting*. Ao final do processo, o **GridSearchCV** seleciona automaticamente a combinação que apresentou o melhor desempenho médio (acurácia) nos *folds* da validação, retornando o modelo ajustado com esses parâmetros.

Entre as principais vantagens do uso do **GridSearchCV** destaca-se a sua capacidade de padronizar o processo de otimização de hiperparâmetros, garantindo uma análise sistemática e estatisticamente fundamentada. Por outro lado, uma limitação importante é o alto custo computacional, principalmente quando se trabalha com um grande número de parâmetros e valores [29]. Na Tabela 4.3 podem ser visualizados todos os valores testados pela função **GridSearchCV** na busca dos parâmetros otimizados para os algoritmos tradicionais e também os valores testados empiricamente no BERT.

Tabela 4.3: Hiperpârametros e valores para algoritmos implementados

Modelo	Hiperparâmetro	Valores
KNN	n neighbors	1 - 31
	weights	distance, uniform
	metric	cosine, euclidean, manhattan
NB	alpha	1, 0.1, 0.01, 0.001, 0.0001
	fit prior	True, False
SVM	C	0.1, 1, 10, 100
	kernel	Linear, RBF, Poly
	gamma	scale, auto
BERT	batch size	16, 32
	learning rate	$1e^{-5}$, $5e^{-5}$, $1e^{-6}$
	epochs	5, 8, 10
	loss function	categorical cross entropy, sparse categorical cross entropy
	dropout	0.6, 0.7, 0.8, 0.9

Algoritmo de Aprendizado Profundo

O algoritmo de DL utilizado foi o BERT, um dos principais algoritmos utilizados em NLP [9]. O ajuste de hiperparâmetros foi realizado a partir de 20% do conjunto de treino. Devido ao BERT ser um algoritmo com uma complexidade muito maior, os ajustes dos hiperparâmetros foram muito mais relevantes, levando a mudanças consideráveis nos resultados a cada ajuste. Além disso, o tempo consumido na otimização e validação dos novos hiperparâmetros é impactado pelo tempo demandado para realizar a execução de cada teste. O algoritmo foi treinado individualmente utilizando cada um dos *datasets* buscando alcançar 20 épocas, e durante o treinamento do modelo foi utilizada a técnica de parada antecipada (do inglês, *Early Stopping*), que monitora a acurácia e interrompe o ciclo de treinamento caso não ocorra melhora, neste caso, em cinco épocas consecutivas.

4.6 Avaliação dos Resultados

Nesta etapa são avaliados os resultados obtidos após a execução dos experimentos a partir do conjunto de teste, buscando evidenciar os pontos fortes e fracos de cada abordagem, bem como comparar o desempenho dos algoritmos tradicionais frente ao algoritmo de DL para os datasets obtidos de cada portal de notícias. Dentre as métricas apresentadas na Seção 2.4, foram utilizadas a *acurácia*, precisão, revocação e F1-Score, sendo essa última considerada, especialmente para avaliar o desempenho na classificação de falsos positivos e negativos. No caso do estudo aqui descrito, a métrica F1-Score consegue lidar bem com dados que não estão completamente balanceados. Portanto, essa métrica foi utilizada como principal para a avaliação [13].

Capítulo 5

Resultados

Este capítulo tem como objetivo apresentar e analisar os resultados das aplicações dos algoritmos e desta forma avaliar os avanços do ML desde o seu princípio até os tempos atuais. Inicialmente, serão expostas as descrições dos cenários de testes e os hiperparâmetros selecionados. Em seguida, serão apresentados os resultados das aplicações de cada um dos algoritmos considerados, a partir dos quais é possível verificar os avanços alcançados pelos algoritmos mais recentes.

5.1 Ambiente de Desenvolvimento

Os experimentos foram realizados no sistema operacional Windows 11, utilizando a linguagem Python, sendo o ambiente de programação a aplicação *PyCharm* que é um *Integrated Development Environment* (IDE). Essa ferramenta possibilita a execução da aplicação Jupyter Notebook contendo as bibliotecas **SciKit-learn**, **Pytorch**, **Pandas**, **Numpy** e **MatplotLib**. Esse ambiente foi utilizado para realizar a implementação dos algoritmos e a realização dos experimentos. Também foi utilizado um computador para realizar os experimentos com as seguintes configurações de *hardware*:

- Processador: AMD Ryzen 3800x @4,2Ghz;
- RAM: 16Gb;
- GPU: GeForce RTX 4060TI 8Gb;

5.2 Principais Palavras do Corpus

A nuvem de palavras exposta na Figura 5.1 foi construída com base nos *datasets* dos portais de notícias após a aplicação do processo de normalização dos dados, descrito na



(a) MoneyTimes



(b) MoneyTimes



(c) MoneyTimes

Figura 5.1: Nuvens de palavras geradas a partir dos *datasets* utilizados.(Autoria Própria)

Seção 2.2.2. Quanto maior o tamanho da palavra na nuvem, mais frequente ela é no *corpus* dos dados utilizados para treinamento, descrito na Seção 2.2.1.

Verifica-se que palavras como “china, ibovespa, alta e mercado” são muito frequentes, ou seja, estão em boa parte dos títulos e são determinantes ao tópico que cada notícia está atrelada, como os exemplos anteriores estão atrelados ao tópico “mercado”.

5.3 Modelos Tradicionais

Nesta seção são apresentados os resultados obtidos com a aplicação dos algoritmos tradicionais de ML. Uma vez que a extração das características foi realizada, foi necessário realizar um ajuste de hiperparâmetros para cada um dos modelos de algoritmos tradicionais selecionados, tendo em vista melhorar e otimizar os resultados dos experimentos. Para essa tarefa, foi aplicada a função **GridSearchCV** passando como argumento um vetor contendo diversos valores de hiperparâmetros para cada um dos algoritmos utilizados. Então, a função testa individualmente cada um dos valores, analisando a acurácia como métrica para verificar a melhora do algoritmo. Esses valores foram utilizados para se verificar a acurácia obtida ao fim de cada iteração, obtendo-se os valores otimizados para os hiperparâmetros.

Através dos resultados obtidos na saída dos modelos, foi possível realizar a montagem das matrizes de confusão; essa forma de analisar os resultados foi apresentada na Seção 2.4, na qual é possível visualizar de forma mais abrangente o resultado de cada um dos

algoritmos. O que ficou claro comparando a literatura com os resultados obtidos é que, graças a uma maior quantidade de texto para realizar o treinamento dos modelos, o resultado dos modelos foi satisfatório em vista da complexidade e dos recursos empenhados para realizar tal tarefa de classificação. Com a aplicação dos algoritmos utilizando os valores otimizados apresentados na Tabela 5.1, foi possível obter os resultados expostos na Tabela 5.2.

Tabela 5.1: Hiperparâmetros utilizados nos modelos tradicionais.

Dataset	Modelo	Hiperparâmetro	Valor
Suno	KNN	n_neighbors	14
		weights	distance
		metric	cosine
	NB	alpha	0.1
		fit_prior	True
	SVM	C	10
		kernel	RBF
		gamma	scale
InfoMoney	KNN	n_neighbors	14
		weights	distance
		metric	cosine
	NB	alpha	0.1
		fit_prior	True
	SVM	C	10
		kernel	RBF
		gamma	scale
Moneytimes	KNN	n_neighbors	29
		weights	distance
		metric	cosine
	NB	alpha	0.1
		fit_prior	True
	SVM	C	1
		kernel	RBF
		gamma	scale

Também fica claro que, entre os modelos de DL tradicionais, o SVM se destaca como o mais preciso, mostrando que os algoritmos foram evoluindo e melhorando a capacidade de classificação. Além disso, o *ensemble* obteve resultados ainda melhores, mostrando que, ao se combinar vários classificadores, é possível superar os resultados dos modelos individuais, sendo um classificador mais robusto e preciso mesmo para *datasets* menos balanceados, como é o caso dos dados coletados dos portais SunoResearch e InfoMoney.

Tabela 5.2: Resultados dos modelos tradicionais.

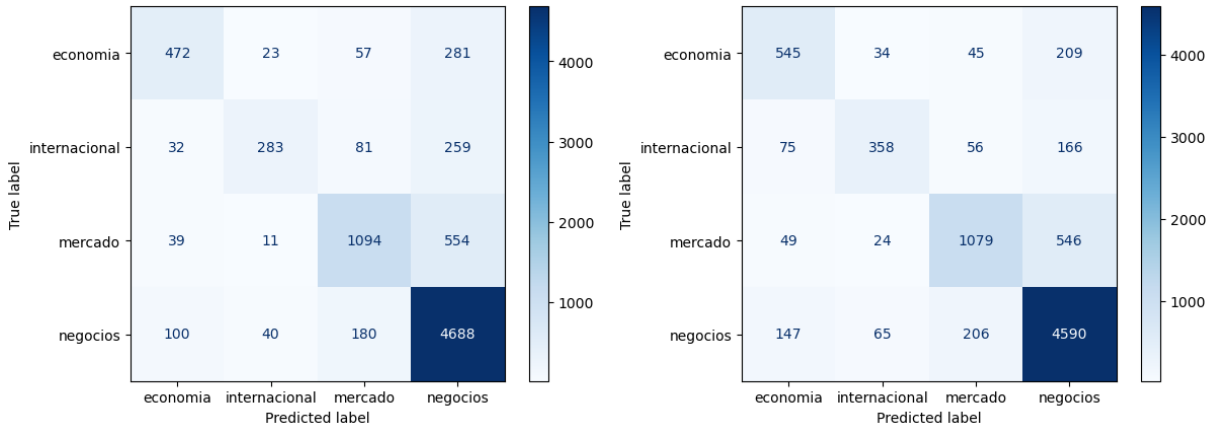
	Datasets					
	Suno		Infomoney		Moneytimes	
Modelo	Acurácia	F1-Score	Acurácia	F1-Score	Acurácia	F1-Score
KNN	79.77%	69.0%	80.90%	77.0%	82.79%	83.0%
NB	80.20%	72.0%	81.97%	78.0%	84.15%	84.0%
SVM	79.77%	77.0%	80.92%	81.0%	82.80%	88.0%
Ensemble	82.13%	74.0%	83.71%	80.0%	87.80%	86.0%

Através desses resultados, foi possível gerar as matrizes de confusão para cada um dos experimentos realizados. As matrizes de confusão possibilitam ter uma visualização melhor quanto à quantidade de cada classificação realizada pelo algoritmo. Também fica bem nítido o quão balanceados estavam os *datasets* verificando a quantidade de ocorrências na diagonal principal. Existe uma melhora na classificação utilizando dados mais balanceados, como no *dataset* do Moneytimes, porém a queda de acurácia não foi muito significativa, com a maior variação de aproximadamente 5 pontos percentuais em relação ao *dataset* menos balanceado, a base de dados do portal Suno. Os resultados estão expressos de forma visual nas Figuras 5.2, 5.3 e 5.4.

5.4 *Deep Learning*

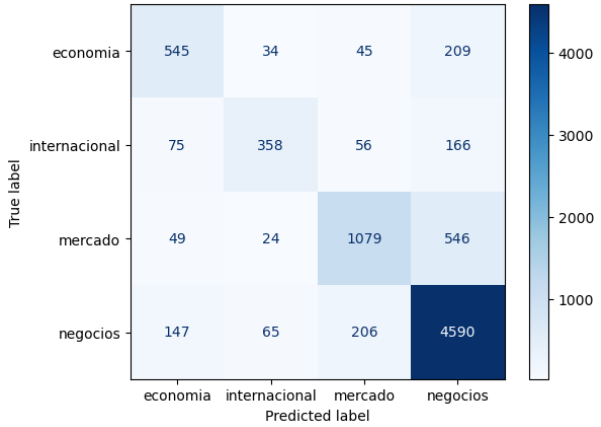
Nesta seção serão apresentados os principais resultados obtidos pela aplicação dos algoritmos de DL apresentados na Seção 2.3. Devido a limitações durante os experimentos, a busca pelos melhores hiperparâmetros foi realizada de forma empírica Tabela 5.3 buscando melhorar os resultados obtidos pelo classificador, parte que se mostrou bastante demorada e custosa; uma vez que o algoritmo demanda muito mais recursos para se obter resultados e, consequentemente, muito tempo para realizar um ajuste fino dos hiperparâmetros. Também, devido a esse maior custo, não foi possível utilizar-se da técnica de *Cross Validation*. Entretanto, mesmo com as adversidades, é possível verificar a partir dos gráficos de acurácia e *loss*, que o modelo de DL apresentou os melhores resultados entre todos os experimentos, mesmo ocorrendo *overfitting*. Característica que se deve à quantidade de dados utilizada para realizar o treinamento do modelo, levando o BERT a ficar menos generalista em outras aplicações.

O classificador BERT obteve melhores resultados em todos os *datasets* e com melhores valores de acurácia e *F1-Score*, mostrando grande capacidade de se adequar a *datasets* menos balanceados, como pode ser verificado na Tabela 5.4. Também é possível ver

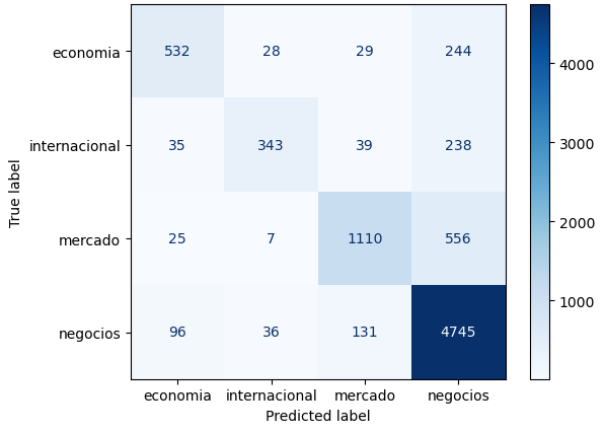


(a) KNN

(b) NB



(c) SVM

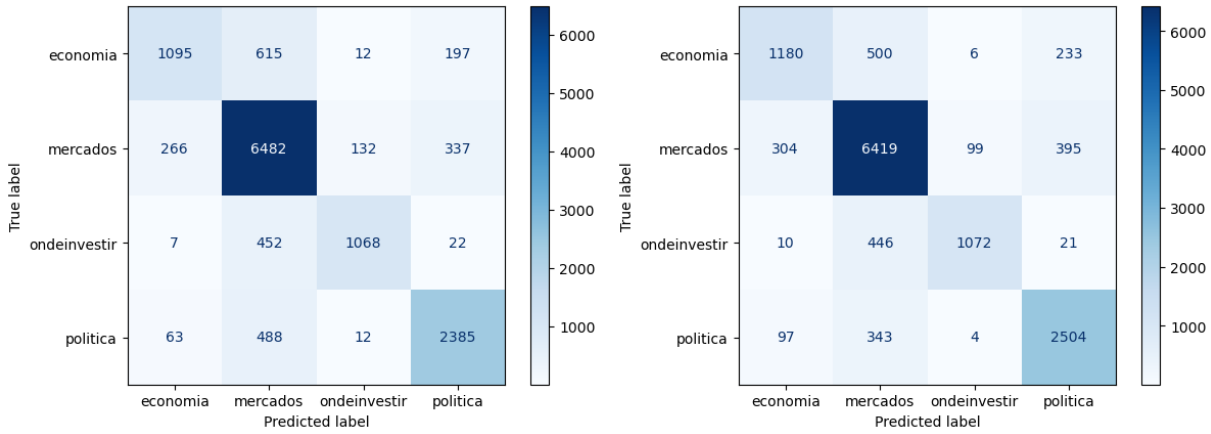


(d) *Ensemble*

Figura 5.2: Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* Suno

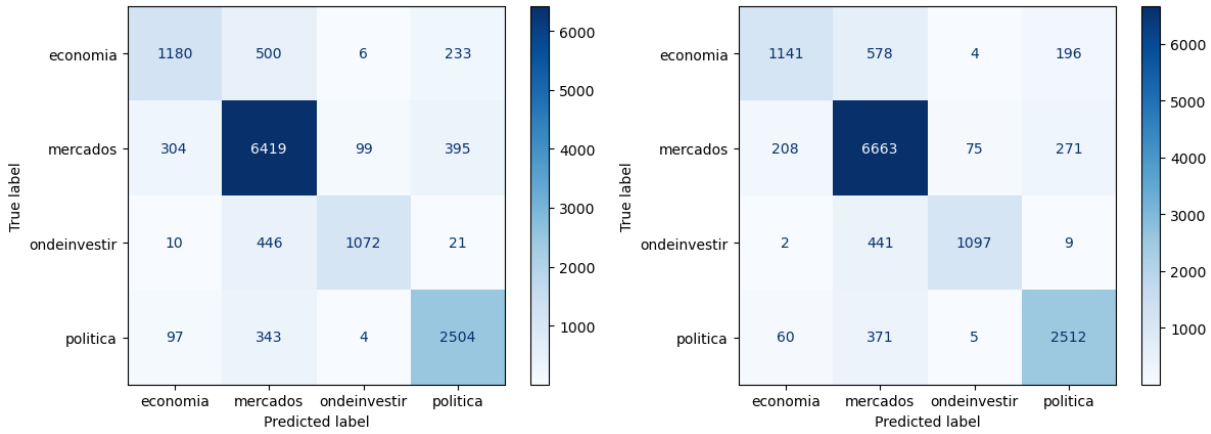
como os falsos positivos e falsos negativos foram menos presentes, como representado nas matrizes de confusão da Figura 5.5. As matrizes também indicam como mesmo as classificações erradas ainda seguem um padrão, já que, no *dataset* do Suno ficou destacado como alguns títulos dentro da classe "mercado" foram classificados como "negocios" e no *dataset* do InfoMoney algumas amostras da classe "economia" foram classificadas como "mercados". Como são classes semelhantes, o algoritmo obteve bons resultados e os erros foram cometidos na classificação de classes com assuntos bem semelhantes.

O overfitting pode ser detectado ao visualizar os gráficos nas Figuras 5.6, 5.7, 5.8 de acurácia e *loss*, o que significa que o algoritmo pode perder a capacidade de generalização. Devido a isso, o modelo treinado com os *datasets* utilizados nesse trabalho poderia não se mostrar tão efetivo em outros dados. Possivelmente, esse comportamento decorre da quantidade de dados, ou seja, demanda grandes quantidades de dados para ser treinado e os *datasets* utilizados não contêm uma quantidade de dados tão grande para o BERT.



(a) KNN

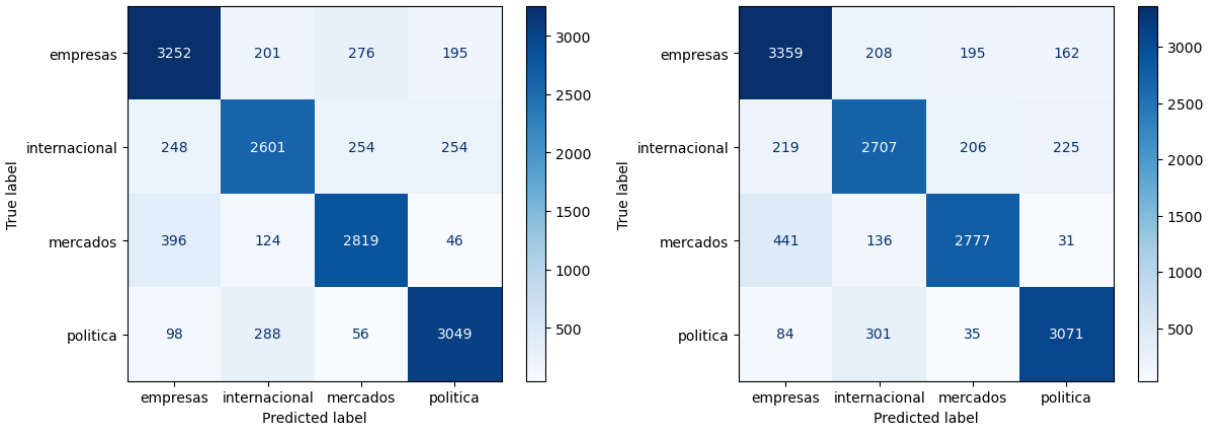
(b) NB



(c) SVM

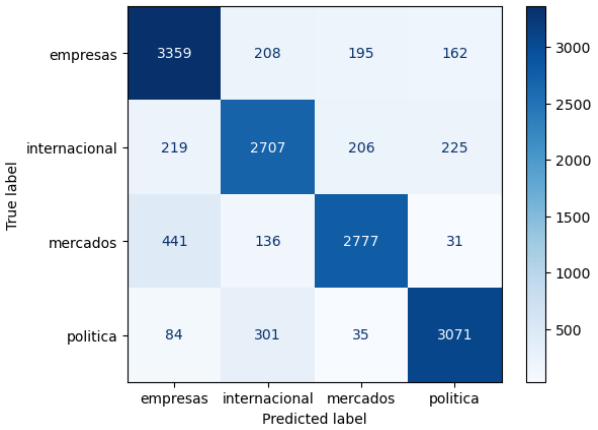
(d) *Ensemble*

Figura 5.3: Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* InfoMoney

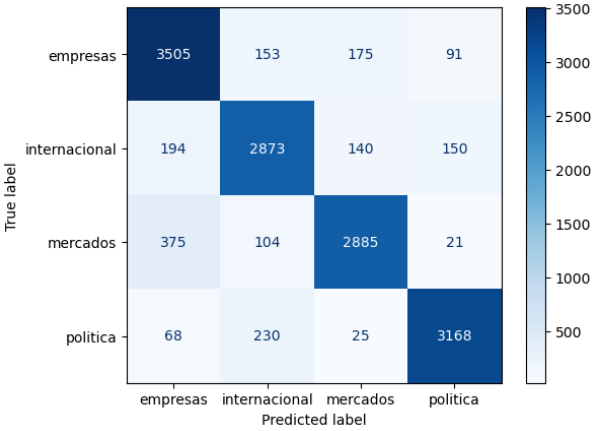


(a) KNN

(b) NB



(c) SVM



(d) *Ensemble*

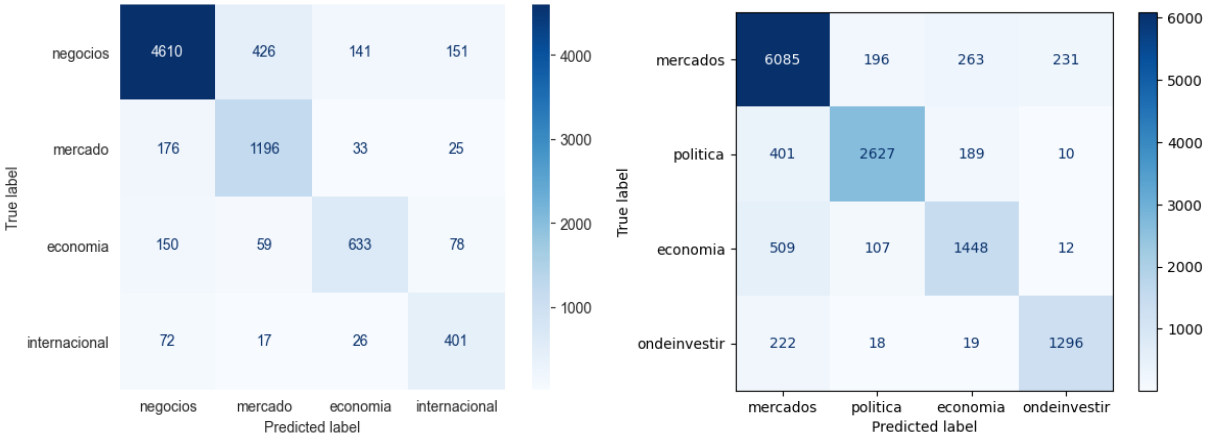
Figura 5.4: Matrizes de confusão dos algoritmos tradicionais e *ensemble* aplicados ao *dataset* Moneytimes

Tabela 5.3: Hiperparâmetros BERT.

Dataset	Hiperparâmetro	Valor
Suno	batch_size	16
	learning_rate	$1e^{-5}$
	epochs	20
	loss_function	categorical_crossentropy
	dropout	0.7
InfoMoney	batch_size	16
	learning_rate	$1e^{-5}$
	epochs	20
	loss_function	categorical_crossentropy
	dropout	0.8
Moneytimes	batch_size	16
	learning_rate	$1e^{-5}$
	epochs	20
	loss_function	categorical_crossentropy
	dropout	0.9

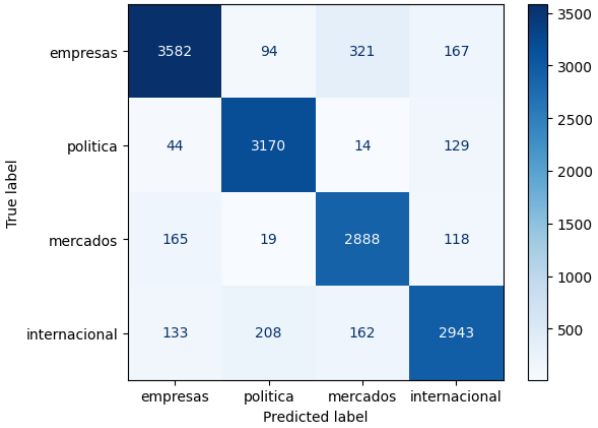
Tabela 5.4: Resultados do BERT.

	Datasets					
	Suno		Infomoney		Moneytimes	
Modelo	Acurácia	F1-Score	Acurácia	F1-Score	Acurácia	F1-Score
Bert	83.70%	77.0%	84.80%	82.0%	89.30%	89.0%



(a) Suno

(b) InfoMoney



(c) Moneytimes

Figura 5.5: Matrizes de confusão BERT aplicado à todos os *datasets*

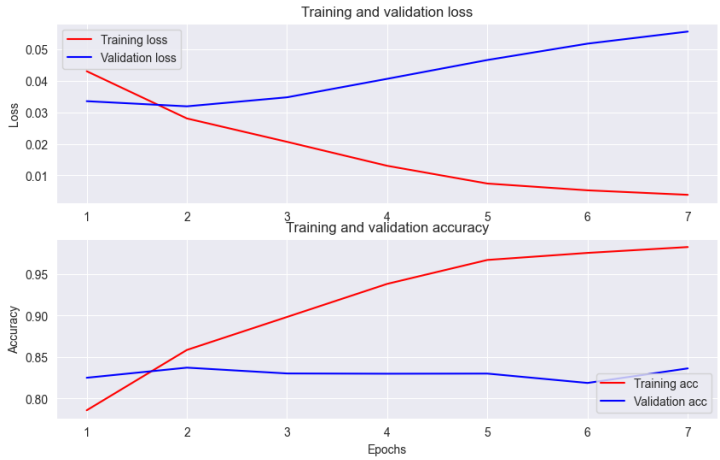


Figura 5.6: *Loss* e Acurácia do algoritmo BERT para o *dataset* Suno

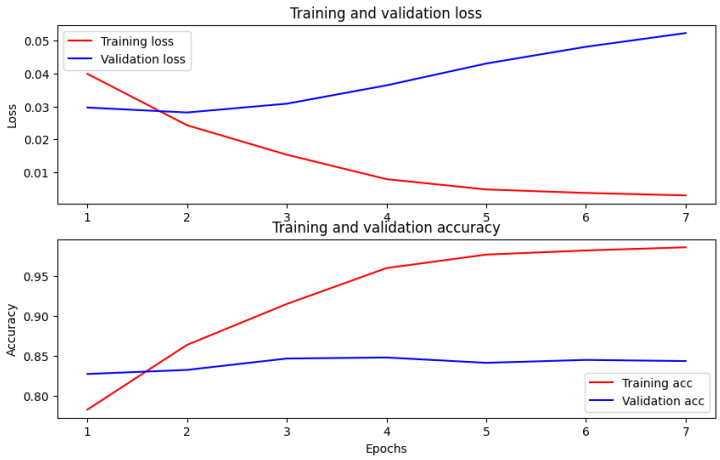


Figura 5.7: *Loss* e Acurácia do algoritmo BERT para o *dataset* InfoMoney

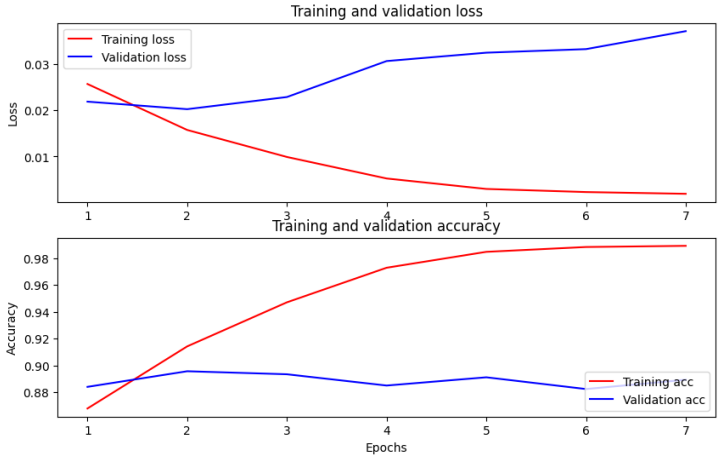


Figura 5.8: *Loss* e Acurácia do algoritmo BERT para o *dataset* Moneytimes

Capítulo 6

Conclusão

Através da extração dos dados de portais *online*, foi possível classificar os dados de acordo com sua escrita de forma assertiva, alcançando o objetivo geral, classificar os títulos de notícias dentro das quatro categorias expostas anteriormente. Ao longo dos testes, foi possível notar que os algoritmos tradicionais são bastante efetivos e conseguem obter um bom desempenho com um baixo custo computacional, porém essa simplicidade reflete em uma baixa generalização e pouca capacidade de se adequar a dados desbalanceados. Com a utilização do *ensemble*, a robustez do classificador melhorou e os resultados também, verificando como diferentes técnicas melhoram os resultados dos classificadores. Já o algoritmo BERT se demonstrou mais versátil em suas aplicações, principalmente pela capacidade de generalização e de se adequar a dados menos balanceados, mas demanda mais tempo e custo computacional para ser treinado; esse *trade-off* acaba sendo reduzido, já que com o treinamento da NN realizado, é possível realizar apenas a classificação de novos textos dentro dos tópicos treinados sem a necessidade de retreinamento do algoritmo.

Outro ponto descoberto durante a implementação dos algoritmos é o quanto sua complexidade cresce, e com isso ficaram mais sensíveis à mudança de seus parâmetros. Os algoritmos KNN e NB se mostram muito sensíveis aos dados, ou seja, são muito dependentes de um bom pré-processamento para realizar a limpeza dos dados e deixá-los adequados à interpretação do algoritmo. Enquanto o BERT é mais permissivo quanto à entrada de dados, seja devido à sua maior capacidade de extrair e agrupar os padrões do texto, como também obter parte da sua análise a partir de dados pré-treinados [9]. O algoritmo de DL se mostrou mais custoso e complexo durante o estudo para realizar a sua implementação, levando consideravelmente mais tempo para se realizar uma boa interpretação e implementação satisfatória do algoritmo; ao realizar o estudo sobre *transformers* foi constatado que é necessário ter um bom entendimento das tecnologias anteriores para se obter uma boa compreensão sobre os algoritmos mais recentes, portanto um bom estudo sobre algoritmos de *Deep Learning* (DL) se tornou necessário para então partir para o estudo

da tecnologia *transformers*.

Quanto aos resultados obtidos durante o treinamento e testes, foi possível constatar os seguintes itens:

- O resultado do algoritmo BERT foi superior aos algoritmos tradicionais KNN e NB, também se demonstrou superior ao algoritmo de DL SVM bem como ao *ensemble* de votação dos três algoritmos citados.
- Os algoritmos tradicionais demonstraram grande simplicidade e efetividade na classificação de texto, sendo eficazes para realizar a classificação de textos de forma simplificada, e muito mais rápida podendo ser implementado em questão de minutos.
- O BERT se mostrou muito eficaz quanto a classificação de textos, obtendo grande precisão na classificação, mas as custas de tomar muito mais tempo, podendo tomar até dias dependendo da quantidade de dados a se realizar o treinamento ou a precisão requerida.

Durante os experimentos, houve limitações quanto ao custo computacional para execução do algoritmo BERT. Devido ao modelo ser bastante custoso para ser treinado, não foi possível realizar o *cross-validation*. A forma de buscar os melhores hiperparâmetros também não foi a ideal. Porém, mesmo com ressalvas, o modelo de DL obteve melhores resultados, principalmente com os *datasets* menos balanceados, alcançando o objetivo esperado.

A área de NLP está em constante evolução, revelando continuamente novos campos de investigação e reafirmando sua relevância como um dos ramos mais importantes da pesquisa em inteligência artificial. Observa-se, no entanto, que a maioria dos estudos nessa área é conduzida com base em corpora da língua inglesa, o que evidencia uma lacuna significativa quando se trata do idioma português. Diante disso, torna-se essencial o desenvolvimento de abordagens e adaptações específicas para algoritmos de ML que atendam às particularidades linguísticas do português. Além disso, o uso de modelos pré-treinados voltados para essa língua, a integração de ferramentas como APIs, por exemplo, a da **OpenAI** e **DeepSeek** bem como a comparação entre essas tecnologias e métodos tradicionais de ML e assim como a aplicação desses recursos em diferentes conjuntos de dados e domínios temáticos, configuram caminhos promissores para investigações e aprimoramentos em pesquisas futuras.

Referências

[1] Mcculloch, W. e Pitts, W: *A logical calculus of ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics, 1943. ix, 11

[2] Hapke, Hannes: *Part 1: Fast, scalable and accurate nlp: Why tfx is a perfect match for deploying bert.* `https://blog.tensorflow.org/2020/03/part-1-fast-scalable-and-accurate-nlp-tensorflow-deploying-bert.html`, 2020. Accessed July 2025. ix, 2, 6, 24, 28

[3] Reis, Tiago: *Tudo sobre o mercado financeiro: o que é e como ele funciona?* `https://www.suno.com.br/guias/mercado-financeiro`. Accessed July 2025. 1

[4] Matheus Schmitz, Roger Immich, Gustavo Pessin e Geraldo P. Rocha Filho: *Towards the categorization of brazilian financial market headlines.* IEEE LATIN AMERICA TRANSACTIONS, 20(02), 2022. 1, 5, 19, 22

[5] B3: *Uma das principais empresas de infraestrutura de mercado financeiro do mundo.* `https://www.b3.com.br/pt_br/b3/institucional/quem-somos/`. Accessed July 2025. 1, 5

[6] Liu, Jian, Junjie Yan, Jun Jiang, Yitong He, Xuren Wang, Zhengwei Jiang, Peian Yang e Ning Li: *Tricti: an actionable cyber threat intelligence discovery system via trigger-enhanced neural network.* Cybersecurity volume 5, Article number: 8 (2022), 2022. 1, 15

[7] Aisah Rini Susanti, Taufik Djatna, Wisnu Ananta Kusuma: *Twitter's sentiment analysis on gsm services using multinomial naïve bayes.* TELKOMNIKA, Vol.15, No.3, 2017. 1, 10, 24, 29

[8] Bramer, Max: *Principles of data mining.* Publishing Company, Incorpo- rated, 2nd edição, 2013. 1, 9, 10, 29

[9] Devlin, Jacob, Ming Wei Chang e Kristina Toutanova Kenton Lee: *Bert: pre-training of deep bidirectional transformers for language understanding.* Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019. 1, 6, 15, 30, 41

[10] Sousa, Matheus Gomes, Kenzo Sakiyama, Lucas de Souza Rodrigues, Pedro Henrique Moraes, Eraldo Rezende Fernandes e Edson Takashi Matsubar: *Bert for stock market sentiment analysis.* IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019. 1, 21

[11] Kamal, Saurabh, Sahil Sharma, Hammam Alshazly Vijay Kumar, Hany S. Hussein e Thomas Martinetz: *Trading stocks based on financial news using attention mechanism*. Mathematics, 2001(10), 2022. 2, 19, 22

[12] Daniel Jurafsky, James H. Martin: *Speech and language processing: An introduction to natural language processing*. Computational Linguistics, and Speech Recognition, Third Edition, 2023. 2, 5, 8

[13] Cunha, Washington, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha e Marcos André Gonçalves: *A survey on text classification: From traditional to deep learning*. Information Processing and Management, 58, 2021. 2, 20, 22, 30

[14] Federal, Senado: *Ativos*. `https://www12.senado.leg.br/manualdecomunicacao/guia-de-economia/ativo`. Accessed July 2025. 2, 4

[15] Jeffrey Stylos, Brad Myers: *Mapping the space of api design decisions*. Dover Publications, 2002. 6

[16] Klene, Stephen Cole: *Mathematical logic*. IEEE Symposium on Visual Languages and Human-Centric Computing, 2007. 6

[17] Corinna Cortes, Vladmir Vapnik: *Support-vector networks*. Machine Learning, volume 20, 1995. 9, 11, 29

[18] Goodfellow I., Bengio Y. e Courville, A: *Deep learning*. `http://www.deeplearningbook.org`. Accessed July 2025. 9, 12

[19] Sepp Hoehreiter, Jürgen Schmidhuber: *Long short-term memory*. Neural computation, volume 9, 1997. 10, 24, 29

[20] C. Baziotis, N. Pelekis, C. Doulkeridis: *Deep lstm with attention for message-level and topic-based sentiment analysis*. Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), 2017. 12, 13, 14

[21] Giacaglia, Giuliano: *How transformers work*. `https://towardsdatascience.com`. Accessed July 2025. 13

[22] Boyu Zhang, Hongyang (Bruce) Yang, Xiao Yang Liu1: *Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models*. IJCAi, 2023. 15

[23] Sousa, Alexandre Santana: *Análise comparativa de redes neurais convolucionais para a detecção de câncer de pulmão em tomografias computadorizadas*. ix(15), 2023. 17

[24] Hao Wang, Xizhuo Chen, Jiangze Du e Kin Keung Lai: *Classification of fintech patents by machine learning and deep learning reveals trends of fintech development in china*. Mathematical Problems in Engineering, 2022(1852447), 2022. 19, 22

[25] Aikaterini Karanikola, Gregory Davrazos, Charalampos M. Liapis e Sotiris Kotsiantis: *Financial sentiment analysis: Classic methods vs. deep learning models*. Intelligent Decision Technologies, 17, 2023. 20, 22

[26] Jaggi, Mukul, Priyanka Mandal, Shreya Narang, Usman Naseem e Matloob Khushi: *Text mining of stocktwits data for predicting stock prices*. Feature Paper Collection in Applied System Innovation, 4(13), 2021. 20, 22

[27] Jan Blumel, Mohamed Zaki: *Comparative analysis of classical and deep learning-based natural language processing for prioritizing customer complaints*. Proceedings of the 55th Hawaii International Conference on System Sciences, 2022. 20, 22

[28] Sy, Christian Y., Lany L. Maceda, Mary Joy P. Canon e Nancy M. Flores: *Beyond bert: Exploring the efficacy of roberta and albert in supervised multiclass text classification*. International Journal of Advanced Computer Science and Applications, 15(3), 2024. 20, 22

[29] Sneh Kalra, Jay Shankar Prasad: *Efficacy of news sentiment for stock market prediction*. Em 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019. 21, 22, 29