UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
DEPARTAMENTO DE COMPUTAÇÃO CIENTÍFICA
CURSO DE CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL
PROCESSAMENTO DE LINGUAGEM NATURAL

EDGARD HENRIQUE COELHO TAVARES DA SILVA

GABRIEL VIEIRA COUTINHO

GUSTAVO HENRIQUE DE CARVALHO COSTA FILHO

CLASSIFICAÇÃO DE REVIEWS OLIST: COMPARAÇÃO DE MODELOS REDES NEURAIS

23 DE OUTUBRO DE 2024
JOÃO PESSOA – PB

1. Apresentação do Problema

A análise de sentimentos tem ganhado relevância no contexto de negócios e plataformas digitais, especialmente em ambientes de e-commerce, onde a opinião dos clientes exerce influência significativa nas decisões estratégicas das empresas. Com o aumento das compras online e a facilidade de acesso a plataformas de avaliação, torna-se fundamental para as empresas identificar e interpretar essas opiniões. Avaliações negativas podem indicar problemas na qualidade dos produtos ou serviços, enquanto avaliações positivas evidenciam aspectos que agradam aos consumidores. Dessa forma, compreender o conteúdo e a polaridade das avaliações permite que as empresas aprimorem processos, aumentem a satisfação dos clientes e tomem decisões mais assertivas.

A presente pesquisa propõe a aplicação de técnicas de Processamento de Linguagem Natural (NLP) e aprendizado de máquina para automatizar a classificação de avaliações em uma base de dados extraída da plataforma Olist, um marketplace que integra diversas lojas e permite aos clientes avaliarem produtos e serviços. Essas avaliações, por serem geradas em grande volume e de forma não estruturada, tornam-se desafiadoras para a análise manual. Portanto, o desenvolvimento de modelos automáticos para identificar a polaridade dessas avaliações classificando-as como positivas ou negativas é necessário e vantajoso.

Este estudo investiga a eficácia de três abordagens complementares: (1) um modelo de aprendizado de máquina tradicional utilizando o algoritmo Naive Bayes Multinomial, (2) uma abordagem baseada em redes neurais recorrentes (RNNs), como LSTM (Long Short-Term Memory) e GRU (Gated Recurrent Unit) e (3) o modelo pré treinado Bert Base em Português. Os métodos serão aplicados para classificar as avaliações como positivas ou negativas, com o objetivo de comparar suas performances e fornecer recomendações sobre a melhor estratégia para tarefas de classificação de texto.

A automação da análise de sentimentos por meio dessas técnicas permitirá identificar padrões comportamentais dos consumidores e aprimorar a experiência de compra na plataforma. Assim, este estudo se insere no escopo das práticas atuais de análise de dados e ciência de dados, oferecendo soluções para desafios comuns no contexto de marketplaces e comércio eletrônico.

2. Objetivos

2.1. Objetivo Geral

Este estudo tem como objetivo geral desenvolver e avaliar a eficácia de modelos automatizados para a classificação de avaliações de clientes da plataforma Olist, determinando se essas avaliações refletem uma experiência positiva ou negativa. Com a implementação desses modelos, espera-se contribuir para a melhoria da análise de sentimentos no contexto do comércio eletrônico, permitindo que a plataforma tome decisões estratégicas com base em padrões identificados nas opiniões dos consumidores. A automação desse processo é essencial para lidar com o grande volume de dados textuais de forma eficiente e proporcionar uma experiência de compra mais satisfatória.

2.2. Objetivos Específicos

Os objetivos específicos são:

1. **Visualização de termos frequentes:**
   Serão geradas **nuvens de palavras** para explorar os termos mais comuns em avaliações positivas e negativas, proporcionando insights sobre os fatores que mais influenciam a satisfação e insatisfação dos clientes
2. **Implementação de um modelo Naive Bayes Multinomial:**
   Este modelo utilizará a técnica de vetorização **TF-IDF** para transformar textos das avaliações em uma representação numérica. A partir desse modelo tradicional, será possível verificar sua eficácia na classificação de sentimentos.
3. **Desenvolvimento de redes neurais recorrentes (LSTM e GRU):**
   Será realizada a implementação de dois tipos de **redes neurais recorrentes (RNNs)**:
   - **LSTM (Long Short-Term Memory):** Camada conhecida por sua capacidade de reter informações importantes por longos períodos em sequências de dados textuais.
   - **GRU (Gated Recurrent Unit):** Variante da LSTM, com arquitetura mais simples e potencialmente mais eficiente.
4. **Uso de embeddings pré-treinados:**
   As RNNs utilizarão **embeddings pré-treinados**, o que permitirá um entendimento mais profundo das relações semânticas entre palavras nas avaliações. Isso visa melhorar a performance do modelo em comparação com vetores TF-IDF.
5. **Implementação do BERT para classificação de texto:**
   Com base no modelo BERT, especializado no processamento de linguagem natural contextualizada, o estudo também avaliará a eficácia do modelo **neuralmind/bert-base-portuguese-cased** para classificação binária das avaliações. A inclusão do BERT permitirá explorar como uma abordagem de estado da arte se compara às demais abordagens empregadas.
6. **Comparação entre os modelos:**
   Serão utilizadas métricas como **acurácia, matriz de confusão e tempo de execução** para comparar a eficácia das abordagens Naive Bayes, LSTM E GRU com embeddings pré treinados e BERT. Essa análise fornecerá uma visão abrangente sobre o desempenho de cada técnica aplicada ao contexto do e-commerce.

Com esses objetivos, o estudo visa não apenas comparar diferentes abordagens de classificação de texto, mas também fornecer uma análise prática que apoie melhorias operacionais e estratégicas para a plataforma Olist.

3. Dados utilizados e pré processamento de dados

Dataset: Brazilian E-Commerce Public Dataset by Olist. O conjunto de dados têm informações de 100 mil pedidos de 2016 a 2018 feitos em vários marketplaces no Brasil com múltiplas dimensões: desde o status do pedido,

preço, pagamento e entrega até a localização do cliente, atributos do produto e, finalmente, comentários escritos pelos clientes.

As duas principais informações retiradas do dataset foram a avaliação do pedido e o comentário deixado para esta avaliação. Por isto, foram removidas as outras colunas e removidas as reviews com nota 3(inconclusivas para a rotulação). As reviews receberam label 0 e 1 de acordo com a nota: 1 para as notas 4 e 5, 0 para as notas 1 e 2.

Além disso, foi utilizado os seguintes regex para remoção de ruídos no texto:

- r'([nN][ãÃaA][oO]|[ñÑ]| [nN])'
  - Esse padrão busca diferentes variações da palavra "não", incluindo acentos e o "ñ", ou apenas a letra "n" isolada com um espaço ao redor e troca pela palavra “Negação”.
- r'http\S+|www\S+'
  - Este padrão busca URLs que começam com "http" ou "www" e que são seguidos por qualquer sequência de caracteres não-espaço.
- r'\s+'
  - Esse padrão busca uma ou mais ocorrências de espaços em branco (incluindo espaços, tabs, quebras de linha, etc.).
- r'\d+'
  - remoção de números
- r'R\$'
  - Troca de real para a palavra “Dinheiro”
- r'(.)\1{2,}'
  - Este padrão busca qualquer caractere (.) que se repete consecutivamente 3 ou mais vezes. Ele usa um grupo de captura (.) e refere-se a esse caractere repetido com \1, seguido por {2,} que indica pelo menos duas repetições adicionais. O objetivo é reduzir repetições de caracteres a apenas uma ocorrência.

Para os modelos que utilizam embeddings foi realizada uma tokenização utilizando o Tokenizer do Keras, com algumas configurações específicas. Primeiro, foi definido um token especial <OOV> para lidar com palavras fora do vocabulário (OOV - Out Of Vocabulary). Em seguida, o tokenizer foi ajustado ao conjunto de textos de treino (X_treino) utilizando o método fit_on_texts, que mapeia cada palavra presente nos textos para um índice único. Após o ajuste, o word_index foi gerado, contendo todas as palavras e seus respectivos índices. O tamanho do vocabulário (vocab_size) foi calculado como o número de palavras mapeadas mais um, para incluir o token <OOV>. Além disso, o comprimento máximo dos textos foi limitado a 120 tokens, com truncamento e padding aplicados no final das sequências

3.1 Visualização dos dados

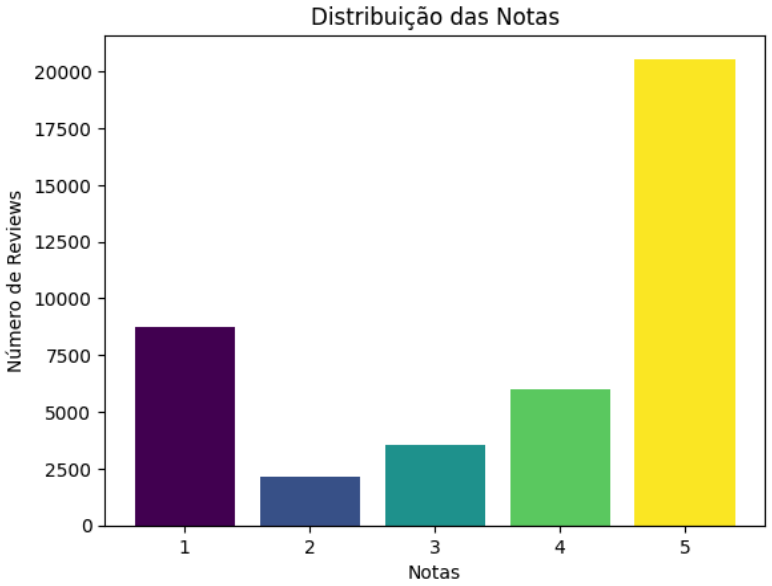


Imagem 1: Distribuição das notas.

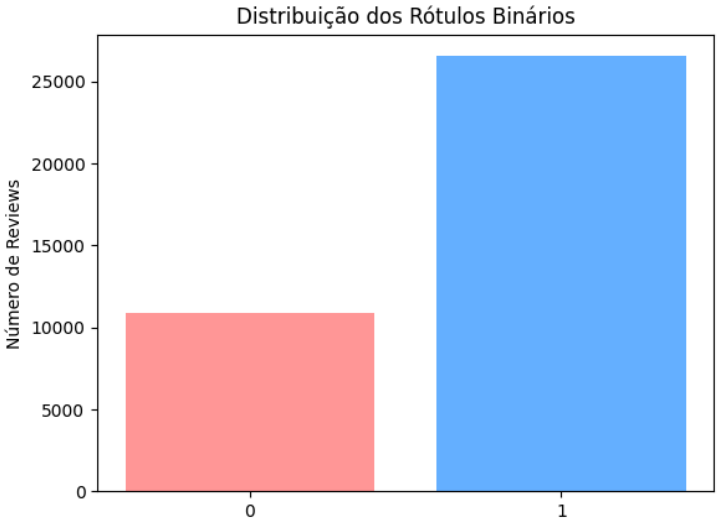


Imagem 2: Distribuição das notas após transformação em dados binários.

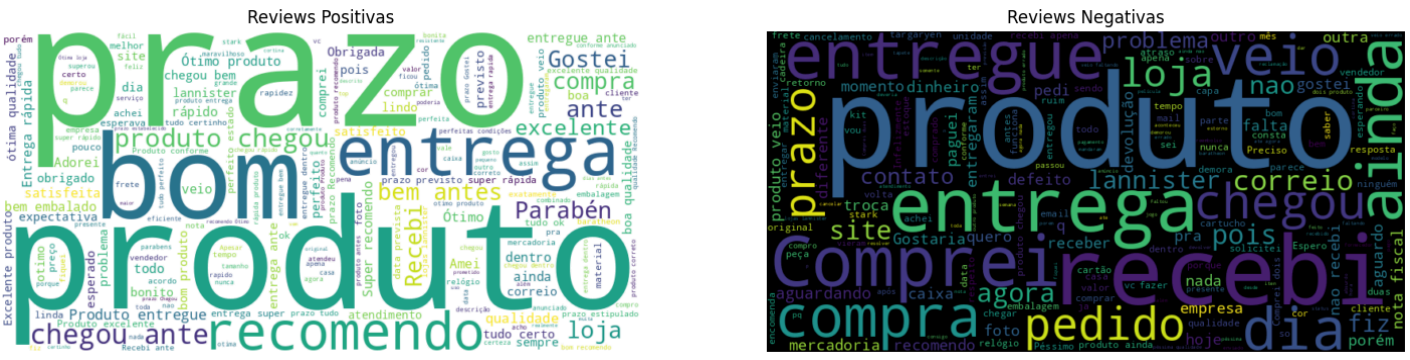


Imagem 3: Nuvem de palavras mais frequentes, nas reviews positivas e negativas respectivamente.

4. Metodologia

4.1 Técnica Utilizada

- **Abordagem 1:** TF-IDF combinado com o algoritmo Multinomial Naive Bayes

Nesta abordagem, foi implementada uma pipeline completa de classificação de textos utilizando a vetorização por TF-IDF (Term Frequency-Inverse Document Frequency) e o algoritmo Multinomial Naive Bayes para a tarefa de classificação.

Para transformar os textos em representações numéricas, foi aplicada a técnica TF-IDF, que atribui um peso a cada termo com base na frequência em um documento (TF) e na raridade do termo em todo o corpus (IDF). O objetivo dessa técnica foi destacar os termos mais relevantes para a análise, atenuando a influência de palavras comuns e sem relevância, como preposições e artigos.

A matriz TF-IDF foi configurada para limitar-se aos 300 termos mais relevantes, excluindo palavras que apareciam em mais de 80% dos documentos e também aquelas presentes em menos de 7 documentos. Assim, garantiu-se que apenas termos significativos fossem considerados, reduzindo a complexidade e o ruído dos dados. Como resultado, foi gerada uma matriz onde cada linha corresponde a um documento e cada coluna representa um termo, com os valores refletindo a relevância de cada termo para o documento.

Após a vetorização dos textos, foi utilizado o Multinomial Naive Bayes para a tarefa de classificação. Esse algoritmo é adequado para problemas de PLN, pois assume que as características (palavras) são independentes, o que simplifica os cálculos probabilísticos, e trabalha bem com dados discretos, como frequências ou contagens de palavras.

- **Abordagem 2:** Embedding Pré treinado em Português combinado com modelos híbridos LSTM e GRU com camada convolucional

Foi utilizado um embeddings Word2Vec com a arquitetura Continuous Bag of Words (CBOW) de 100 dimensões, treinado com dados do repositório do Núcleo Interinstitucional de Linguística Computacional (NILC) da USP São Carlos. Este modelo foi escolhido por sua capacidade de mapear palavras para vetores em um espaço semântico, onde palavras com significados semelhantes estão próximas. Cada palavra é representada como um vetor de 100 números, com essas dimensões correspondendo a características ocultas como emoção, função e intenção. A escolha por 100 dimensões visou manter um equilíbrio entre detalhamento e simplicidade, de forma a capturar nuances sem aumentar a complexidade do modelo.

O repositório do NILC forneceu uma base rica para as 929.607 palavras únicas presentes no dataset da olist, representadas nesse modelo de 100 dimensões.

Duas arquiteturas de redes recorrentes foram implementadas e comparadas: LSTM e GRU, ambas em versões bidirecionais, para capturar relações contextuais das palavras em ambas as direções (frente e trás).

LSTM - Long Short Term Memory

Camada Conv1D + MaxPooling: Para reduzir a dimensionalidade e destacar padrões locais no texto, uma camada Conv1D foi aplicada com 32 filtros, kernel size de 5, e ativação ReLU, seguida de uma camada de MaxPooling1D com tamanho 2.

LSTM Bidirecional: Foram utilizadas duas camadas LSTM, ambas com 64 unidades. A primeira camada retornou sequências completas para a segunda, enquanto a segunda retornou apenas os estados finais.

Dropout e Recurrent Dropout: Para prevenir overfitting, dropout de 30% foi aplicado nas unidades LSTM.

Regularização: Para melhorar a capacidade de generalização do modelo, aplicou-se um dropout de 50% antes da camada densa, além de normalização em batch (Batch Normalization).

Camada Densa: A última camada contém 64 neurônios com ativação ReLU e regularização L2 (λ=0.01), garantindo um controle adicional contra overfitting.

GRU - Gated Recurrent Unit

Camada Conv1D + MaxPooling: Similar ao LSTM, uma camada Conv1D com 32 filtros e kernel size 5 foi aplicada com ativação ReLU, seguida de uma camada de MaxPooling1D com tamanho 2 para redução de dimensionalidade.

GRU Bidirecional: Foram utilizadas duas camadas GRU bidirecionais, com 64 unidades cada. A primeira retornou sequências, enquanto a segunda forneceu os estados finais.

Dropout: O modelo GRU utilizou um dropout de 20%.

Regularização: Além do dropout, aplicou-se um dropout de 30% antes da camada densa, em conjunto com normalização em batch (Batch Normalization).

Camada Densa: A camada densa final incluiu 64 neurônios com ativação ReLU e regularização L2 (λ=0.005).

- **Abordagem 3:** Transfer Learning utilizando o modelo BERT Base para o português

A arquitetura do modelo é baseada no BERT Base, composto por 12 camadas Transformer, totalizando 110 milhões de parâmetros. Este modelo foi integrado com uma camada densa para classificação de duas classes, utilizando a estrutura TFBertForSequenceClassification.

Camada Densa: A última camada densa tem 2 neurônios, representando as duas classes. Essa camada recebe como entrada um vetor de 768 dimensões e, a partir dos pesos ajustáveis, gera duas saídas (logits), uma para cada classe de sentimento.

4.2 Experimento para avaliar a técnica utilizada

Para avaliar o desempenho do modelo, foi gerado um relatório de classificação, apresentando métricas como precisão, recall e F1-score para cada classe.

Além disso, foi construída uma matriz de confusão para mostrar o número de acertos e erros em cada classe, permitindo uma visualização clara dos resultados.

Para a **Abordagem 2** foi utilizado os seguintes valores para o treinamento:

- 20 épocas
- Batch Size: 128
- Otimizador: Adam (learning rate = 0.001)
- Função de Perda: Binary Crossentropy
- Early Stopping:
    - Monitoramento de val_loss (perda na validação).
    - Paciencia: 5 épocas sem melhora.
    - Após 3 épocas, o menor val_loss foi alcançado.
- Recuperação de Pesos:
    - Melhor modelo recuperado a partir da época 3° para a GRU e 4° época para a LSTM, onde o menor val_loss foi observado.

para a **Abordagem 3** foi utilizado os seguintes valores para o treinamento:

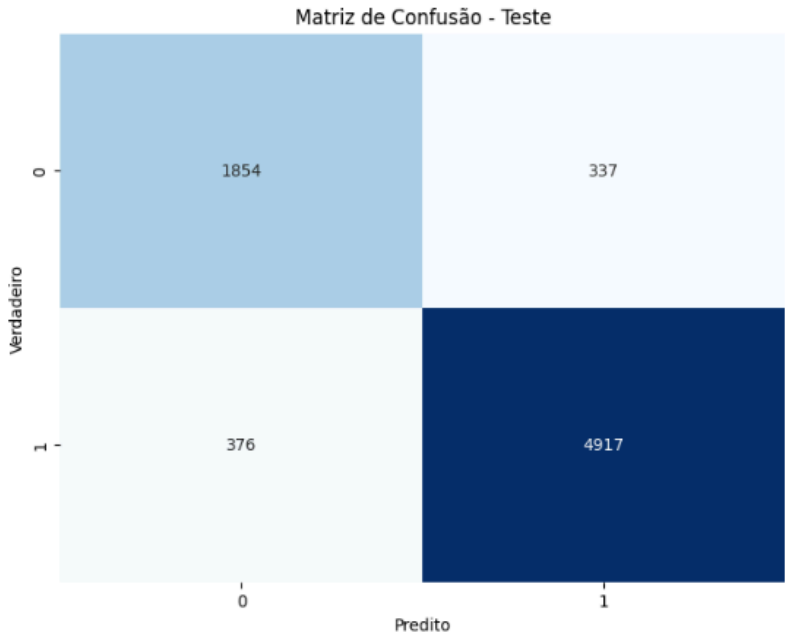
- 3 épocas
- Batch size: 16
- Adam (learning_rate = 5e-5)
- Função de Perda: Sparse Categorical Crossentropy

5. Resultados

- **Abordagem 1:**

Multinomial Naive Bayes

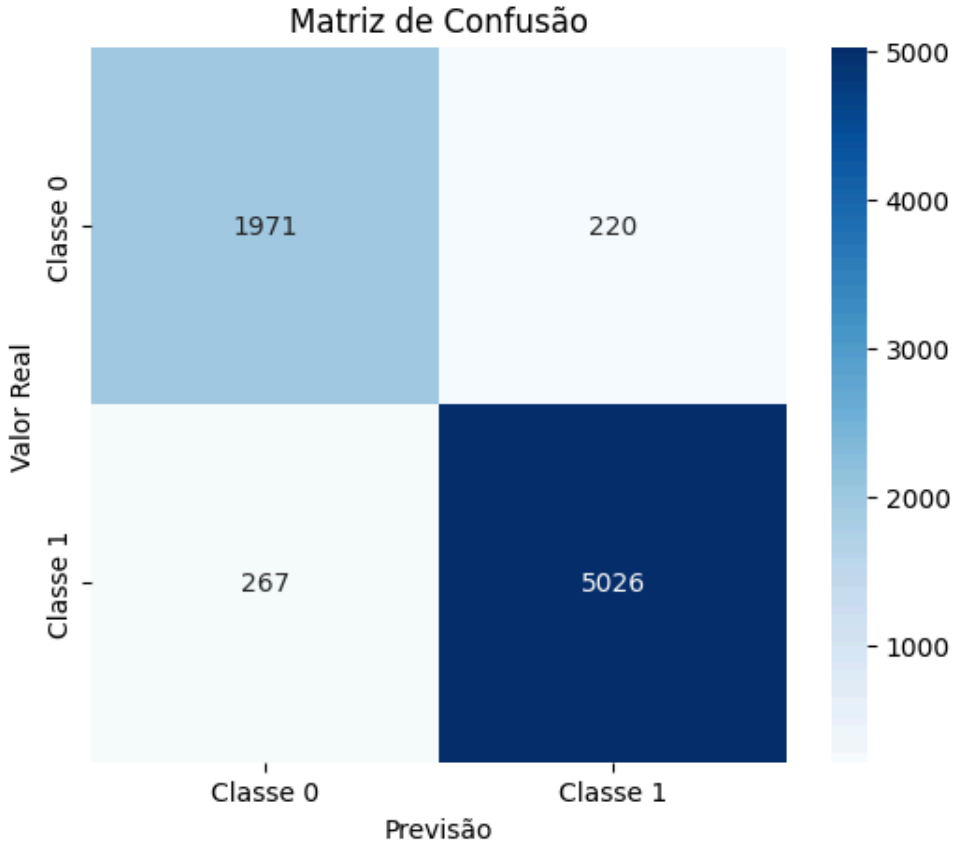
Acurácia geral: 90% / Acurácia Positivas: 94% / Acurácia Negativas: 83%



- **Abordagem 2:**

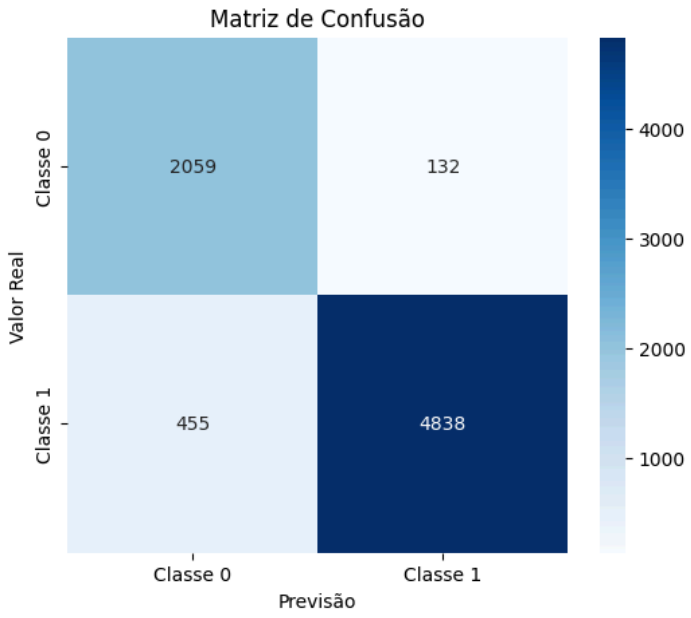
LSTM:

Acurácia geral: 93% / Acurácia Positivas: 96% / Acurácia Negativa: 88%



GRU:

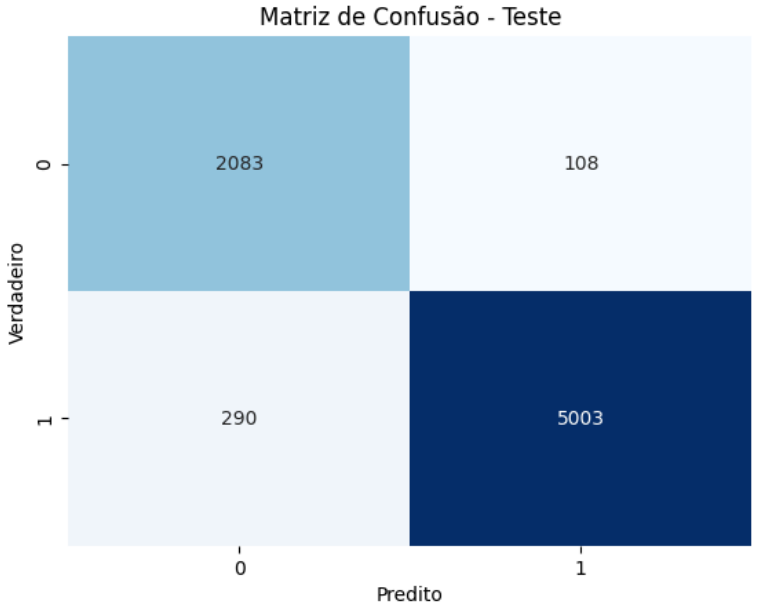
Acurácia geral: 92% / Acurácia Positivas: 97% / Acurácia Negativa: 82%



- Abordagem 3

Bert:

Acurácia geral: 95% / Acurácia Positivas: 98% / Acurácia Negativa: 88%



Comparativo entre as abordagens:

Abordagem	**Acurácia**	**Tempo de Treinamento**
Multinomial Naive Bayes (TF-IDF)	90%	>1 min
GRU (Embedding pré treinado)	92%	2 min
LSTM (Embedding pré treinado)	93%	20 min
Bert (Transfer Learning)	95%	45 min

Com base na comparação dos modelos para o problema de classificação da Olist, é possível observar que, embora o Multinomial Naive Bayes com TF-IDF tenha o menor tempo de treinamento, sua acurácia (90%) é inferior às abordagens baseadas em embeddings pré-treinados e transfer learning. O LSTM, com uma acurácia de 93%, oferece um equilíbrio razoável entre desempenho e tempo de treinamento, mas o GRU se destaca por alcançar 92% de acurácia em um tempo significativamente menor (2 minutos). Já o modelo Bert, utilizando Transfer Learning, atinge a maior acurácia (95%), mas ao custo de um tempo de treinamento mais elevado (45 minutos). Assim, a

escolha do modelo ideal depende de uma análise entre a necessidade de alta acurácia e as restrições de tempo computacional.

Além dos resultados obtidos, é importante destacar que o desbalanceamento de classes pode ter influenciado negativamente na acurácia, especialmente nos reviews negativos, que foram menos representados nos dados. Esse desbalanceamento pode ter causado uma maior dificuldade para os modelos em identificar corretamente os reviews dessa classe, impactando na performance geral. Como trabalho futuro, seria interessante explorar técnicas de balanceamento de classes, como oversampling ou undersampling, para melhorar a precisão em todas as categorias e, potencialmente, aumentar mais ainda a acurácia geral.

Referências

KAGGLE**.** Brazilian E-Commerce Public Dataset by Olist. Disponível em: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce. Acesso em: 23 out. 2024.

NÚCLEO INTERINSTITUCIONAL DE LINGUÍSTICA COMPUTACIONAL (NILC). NILC Embeddings. Disponível em: http://nilc.icmc.usp.br/embeddings. Acesso em: 23 out. 2024.

NEURALMIND**.** BERT-base Portuguese Cased. Disponível em: https://huggingface.co/neuralmind/bert-base-portuguese-cased. Acesso em: 23 out. 2024.