UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Arielly Gonçalves Lima

1. Introdução

*O presente trabalho utiliza o conjunto de dados **B2W-Reviews01**, composto por mais de 130 mil avaliações de produtos realizadas por usuários da B2W Digital. O dataset inclui textos de reviews e rótulos associados, permitindo explorar duas tarefas de interesse: **polaridade** (positiva ou negativa, derivada das notas atribuídas) e **classificação** (notas de 1 a 5 estrelas). Além disso, o conjunto apresenta partições estratificadas em dez dobras, vocabulário amplo e distribuição desbalanceada entre as classes, características comuns em bases reais de opinião em língua portuguesa.*

*Neste trabalho, o dataset será utilizado para investigar como diferentes escolhas de **pré-processamento** e **extração de atributos** influenciam o desempenho de um modelo de classificação de texto. Na primeira etapa, são implementadas funções de preparação textual, incluindo normalização, remoção de stopwords, stemming e lematização. Na segunda, são desenvolvidos métodos de representação do texto(extração de atributos), como **CountVectorizer**, **TF-IDF** e **Word2Vec**. Por fim, na terceira etapa, um classificador único é aplicado a todos os cenários, permitindo comparar: (a) o desempenho com e sem pré-processamento, (b) as diferentes técnicas de extração de atributos e (c) lematização versus stemming.*

Assim, o dataset B2W serve como base prática para análise, comparação e compreensão dos impactos que técnicas de pré-processamento e representação textual exercem sobre modelos de aprendizado de máquina aplicados à análise de sentimentos em língua portuguesa.

2. Observações

*Embora o dataset disponibilize colunas contendo o texto já processado e tokenizado, optou-se por não utilizá-las diretamente. As etapas de pré-processamento foram realizadas integralmente a partir da coluna **review_text**, que contém o texto original das avaliações. Houve dúvidas iniciais sobre a possibilidade de aproveitar as versões prontas (**review_text_processed** e **review_text_tokenized**), porém, para atender aos objetivos da avaliação e garantir total controle sobre o fluxo de processamento, decidiu-se executar todo o tratamento manualmente.*

*A coluna **polarity** foi utilizada como rótulo (label), mantendo seu formato binário de 0 e 1. Avaliações com valores ausentes nessa coluna foram removidas para evitar inconsistências no treinamento e nas métricas finais.*

*Além disso, como o dataset completo possui um volume muito elevado de registros, trabalhou-se com uma **amostra aleatória de 7.000 linhas**, selecionada para tornar o processamento mais eficiente e viável dentro do escopo da atividade. Essa amostra preserva características importantes da base original, permitindo realizar os experimentos de forma consistente sem comprometer a representação dos dados.*

3. Resultados e discussão

Inicialmente, foi desenvolvido um processo completo de pré-processamento textual, cujo objetivo é preparar os dados brutos do dataset para a etapa de extração de atributos. As etapas aplicadas incluem a normalização do texto, remoção de elementos irrelevantes (como links, menções e números), tokenização, lematização, filtragem de stopwords e, quando necessário, aplicação de stemming. Esse conjunto de operações permite padronizar, limpar e reduzir o texto a seus elementos linguísticos essenciais, garantindo que os dados estejam consistentes, estruturados e adequados para os métodos de representação e classificação utilizados nas próximas fases do trabalho.

Em seguida, foi implementado um conjunto de métodos para extração de atributos textuais, reunidos em uma única classe para facilitar a organização e reutilização do código em outras etapas do trabalho. As técnicas contempladas incluem análise estatística básica dos textos, geração de representações vetoriais por meio de CountVectorizer e TF-IDF, e criação de embeddings semânticos com Word2Vec. Cada abordagem permite capturar diferentes aspectos do conteúdo textual — desde frequência de termos até relações semânticas entre palavras — garantindo múltiplas formas de representar os dados para posterior aplicação de algoritmos de classificação.

*Nesta etapa final, foi conduzido todo o processo experimental utilizando o classificador **Multinomial Naive Bayes**, comparando diferentes estratégias de pré-processamento e extração de atributos. Primeiro, avaliou-se o desempenho do modelo com e sem pré-processamento para todas as técnicas de representação textual; em seguida, identificou-se a melhor técnica entre elas considerando apenas os dados pré-processados; e, por fim, compararam-se diretamente lematização e stemming aplicadas à técnica vencedora. Esse fluxo permite observar de forma clara como cada escolha influencia a acurácia do NB no problema de classificação de textos.*

3.1 item a)

Foram aplicadas as mesmas técnicas de extração de atributos nos dois cenários (com e sem pré-processamento), possibilitando uma comparação direta.

Os resultados mostram que, em geral, as versões sem preprocessamento tiveram ligeiramente melhor desempenho. TF-IDF sem preprocessamento obteve a maior acurácia (0.905), seguido por CountVectorizer (0.902857). Já com preprocessamento, CountVectorizer liderou (0.898571), mas com uma leve queda em relação à versão sem preprocessamento. Word2Vec e Análise Estatística apresentaram os menores valores em ambas as condições.

Técnica	Preprocessamento	Acuracia
Analise Estatística	Sem	0.735000
CountVectorizer	Sem	0.902857
TF-IDF	Sem	0.905000
Word2Vec	Sem	0.774286
Analise Estatística	Com	0.730000
CountVectorizer	Com	0.898571
TF-IDF	Com	0.892857
Word2Vec	Com	0.748571

Os resultados mostram que o preprocessamento nem sempre melhora a acurácia. Técnicas como TF-IDF e CountVectorizer tiveram melhor desempenho sem preprocessamento, sugerindo que a preservação do texto original mantém nuances úteis para a classificação. Já Word2Vec teve desempenho inferior, possivelmente por não se adaptar bem ao modelo probabilístico utilizado.

3.2 item b)

*Em seguida, foram analisados apenas os resultados obtidos **com** pré-processamento, a fim de identificar qual técnica de representação textual entregou o melhor desempenho geral.*

A tabela abaixo foca apenas nos resultados com preprocessamento. CountVectorizer obteve a maior acurácia (0.898571), seguido por TF-IDF (0.892857), Word2Vec (0.748571) e Análise Estatística (0.730000). A diferença entre CountVectorizer e TF-IDF é pequena. Word2Vec e Análise Estatística ficaram bem abaixo das demais.

	Técnica	Preprocessamento	Acuracia
4	Analise Estatística	Com	0.730000
5	CountVectorizer	Com	0.898571
6	TF-IDF	Com	0.892857
7	Word2Vec	Com	0.748571

Com preprocessamento, CountVectorizer foi a técnica mais eficaz, seguida de perto por TF-IDF. Representações baseadas em contagem funcionam bem nesse cenário por gerarem atributos discretos. Word2Vec teve desempenho inferior, possivelmente por não se alinhar bem ao tipo de variáveis exigidas. Já a Análise Estatística, por ser mais simples, não capturou informações suficientes para uma boa classificação.

3.3 item c)

Por fim, realizou-se uma comparação direta entre dois métodos de redução morfológica:

- ***Lematização (Lemma)***
- ***Stemming***

A lematização obteve uma acurácia de 0.898571, enquanto o stemming ficou ligeiramente abaixo, com 0.892857. A diferença entre os dois métodos é pequena.

	Técnica	Método	Acuracia
0	bow	lemma	0.898571
1	bow	stem	0.892857

Os resultados indicam que, no uso do modelo BoW, a lematização apresenta uma pequena vantagem em relação ao stemming. Isso se deve ao fato de que a lematização conserva melhor o significado original das palavras ao reduzi-las à forma canônica, enquanto o stemming aplica cortes mais agressivos, gerando formas truncadas que nem sempre correspondem a palavras reais. Como o modelo depende da frequência e distinção entre termos, manter formas mais precisas pode favorecer a classificação. Apesar disso, a diferença é sutil, mostrando que ambos os métodos são viáveis, com leve superioridade da lematização.

4. Conclusões

*Os resultados foram condizentes com o esperado, considerando o classificador utilizado. Modelos baseados em frequência, como o **Bag-of-Words**, apresentaram bom desempenho. A acurácia ligeiramente superior sem preprocessamento indica que remover elementos do texto pode eliminar informações relevantes, como variações morfológicas e termos com nuances importantes.*

*A **lematização** superou o stemming, preservando formas canônicas e maior fidelidade semântica, enquanto o stemming pode gerar formas truncadas que prejudicam a precisão. O **Word2Vec** teve desempenho inferior, pois representações contínuas e semânticas não se ajustam bem ao modelo probabilístico, que depende de variáveis discretas. A **Análise Estatística**, por sua simplicidade, não capturou informações suficientes para competir.*

Em resumo, técnicas tradicionais de frequência — especialmente Bag-of-Words com CountVectorizer e lematização — são as mais adequadas neste cenário, confirmando as expectativas para o classificador utilizado e técnicas utilizadas.

5. Próximos passos

Com base nos resultados obtidos até aqui, algumas direções futuras podem fortalecer e ampliar a análise realizada:

- ***Aplicar outras métricas de avaliação***
  - *Além da acurácia, incluir métricas como **precisão, recall e F1-score**, que oferecem uma visão mais detalhada sobre o desempenho do modelo em classes desbalanceadas.*
  - *Utilizar **matriz de confusão** para identificar padrões de erro e compreender quais classes são mais difíceis de prever.*
- ***Experimentar classificadores mais robustos***
  - *Testar algoritmos como **Logistic Regression**, **Support Vector Machines (SVM)**, **Random Forest** e **XGBoost**, que podem capturar relações mais complexas entre atributos.*
  - *Avaliar também o uso de **redes neurais profundas**, especialmente em conjunto com embeddings modernos, para verificar ganhos em tarefas semânticas.*
  - *Comparar o desempenho desses modelos com o Naive Bayes, identificando cenários em que cada um se destaca.*
- ***Explorar métodos modernos de representação de texto***
  - *Incorporar embeddings pré-treinados em português, como **FastText**, **Word2Vec pré-treinado** e **BERTimbau**, que capturam melhor o contexto e significado das palavras.*

- *Comparar essas abordagens com as técnicas tradicionais (CountVectorizer e TF-IDF), verificando ganhos em termos de acurácia e generalização*

- ***Avaliar outras estratégias de pré-processamento***
  - *Investigar o impacto de manter ou remover elementos como **emojis, gírias, negações e sinais de pontuação**, que podem carregar informações semânticas relevantes.*
  - *Testar diferentes listas de **stopwords** e analisar se a remoção ou preservação delas influencia positivamente os resultados.*
  - *Explorar técnicas de **normalização específicas para português**, como tratamento de flexões verbais e variações regionais.*

Os passos citados permitem evoluir o projeto para um nível mais avançado, ampliando tanto a profundidade da análise quanto a aplicabilidade prática dos modelos de classificação de texto em língua portuguesa.