UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Ruan Rodrigues Sousa

1. Introdução

O dataset 'twitter-sentiment-pt-BR-md-2-l', disponível no Hugging Face, contém 20.000 tweets em português, rotulados como positivos ou negativos. Este relatório aborda o pré-processamento e a vetorização desse conjunto de dados para avaliar a performance de um classificador de sentimentos.

O pré-processamento incluiu a remoção de letras maiúsculas, menções (@usuário) e stopwords. Também foram aplicadas técnicas de stemming e lematização para simplificar as palavras. Essas funções foram implementadas de forma a serem facilmente reutilizadas.

Na extração de atributos, foram desenvolvidas manualmente funções para CountVectorizer e TF-IDF. O CountVectorizer gerou uma matriz de contagem de palavras, enquanto o TF-IDF ponderou a importância dos termos.

Um classificador de regressão logística foi usado para comparar diferentes abordagens de pré-processamento e vetorização. Inicialmente, avaliou-se o impacto do pré-processamento com TF-IDF. Em seguida, foram comparadas as técnicas de CountVectorizer e TF-IDF com o melhor pré-processamento. Por fim, analisou-se o efeito da lematização versus stemming na performance do modelo.

2. Observações

Durante a execução do projeto, foi necessário trabalhar apenas com uma parte do dataset devido ao seu tamanho considerável. O dataset completo de 20.000 tweets causou problemas de memória e lentidão no processamento, o que resultou em erros durante as etapas de pré-processamento e vetorização. Para contornar esses problemas, foi adotada uma abordagem de amostragem, utilizando uma fração menor dos dados para garantir que o processo fosse executado de forma mais eficiente e sem erros. Essa estratégia permitiu a continuidade do trabalho e a realização das análises necessárias, embora tenha implicado em uma redução na quantidade de dados analisados.

3. Resultados e discussão

No pré-processamento do dataset de tweets em português, diversas técnicas foram aplicadas para melhorar a qualidade dos dados. Os textos foram convertidos para minúsculas, menções e emojis foram removidos, e stopwords foram filtradas para reduzir o ruído.

Duas abordagens linguísticas foram utilizadas: stemming e lematização. O stemming simplifica as palavras à sua raiz, enquanto a lematização busca reduzir as palavras às suas formas básicas, considerando o contexto gramatical. No entanto, a lematização enfrenta desafios em português devido à complexidade morfológica do idioma, o que pode levar a lemas imprecisos e impactar a análise de sentimentos. Essas técnicas foram demonstradas em uma amostra de tweets, permitindo a avaliação de seu impacto no texto processado.

A escolha entre stemming e lematização pode influenciar significativamente os resultados da análise. O stemming, por ser mais agressivo e simplificador, pode gerar uma maior generalização das palavras, o que pode ser benéfico para a consolidação de termos semelhantes. Por outro lado, a lematização oferece uma redução mais precisa e gramaticalmente correta, mas pode não lidar tão bem com as peculiaridades do português. A compreensão dessas diferenças é crucial para ajustar o pré-processamento conforme as necessidades específicas da análise de sentimentos, garantindo a melhor preparação dos dados para os modelos de machine learning.

obs- comentarios 9 e 10 extraídos da saida do arquivo preprocessing.py

COMENTÁRIO ORIGINAL	(Com Stemming)	(Com Lematização):
nossa eu to muito chateada com o vacilo de hoje de não me chamarem para o niver :(	to chate vacil hoj cham niv	to chatear vacilo hoje chamar niver
@coutinholizz Você realmente não sabe o que está falando kk :)	real sab fal kk	realmente saber falar kk

No segundo passo, foram desenvolvidas funções personalizadas para a extração de atributos textuais usando **CountVectorizer** e **TF-IDF**. A função `build_vocabulary` constrói um vocabulário único a partir de um conjunto de documentos, garantindo que

apenas palavras distintas sejam consideradas. A função `count_vectorizer` converte os textos em uma matriz de contagem de palavras, onde cada elemento representa a frequência de uma palavra específica em um documento. Por sua vez, a função `tfidf_vectorizer` calcula a matriz TF-IDF, uma técnica que ajusta as contagens de palavras com base em sua frequência em diferentes documentos e na importância de cada termo. Ambas as funções foram implementadas manualmente para proporcionar maior controle sobre o processo de vetorização e permitir fácil adaptação e reutilização em futuros projetos. Essas abordagens oferecem uma representação precisa dos dados textuais, essencial para análises subsequentes.

Estrutura da Matriz de Contagem

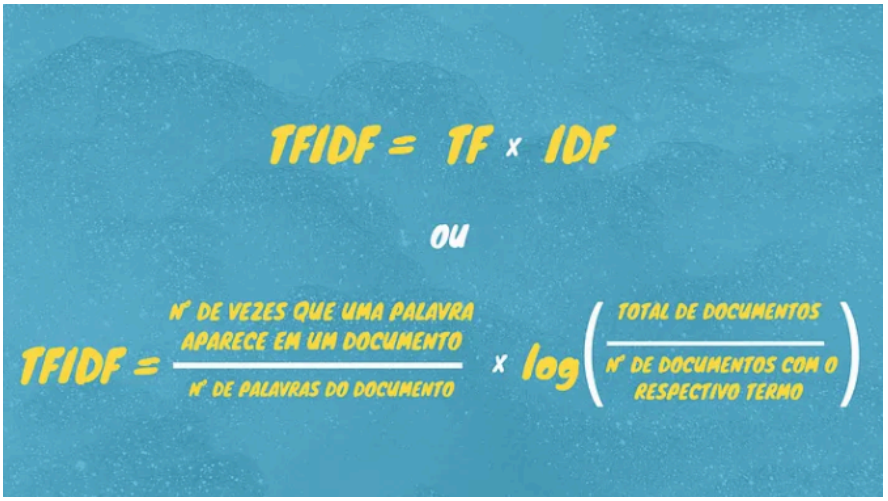
1. ***Documentos como Linhas:*** *Cada linha da matriz corresponde a um documento.*
2. ***Palavras como Colunas:*** *Cada coluna da matriz corresponde a uma palavra do vocabulário.*
3. ***Contagem de Palavras:*** *O valor na célula* `[i, j]` *da matriz é a contagem de ocorrências da palavra* `j` *no documento* `i`*.*

SAIDA EXTRAÍDA DO CODIGO FEITO SOBRE O DATASET
https://huggingface.co/datasets/johnidouglas/twitter-sentiment-pt-BR-md-2-

	saiba	fÃ©	assim	bike	amo	menino	ngm	para	tu	vai	feio	eu	talvez	jÃ¡	nao	kaka	afffffff	definir	ando	da
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0

O que é a Matriz TF-IDF?

A matriz TF-IDF é uma representação vetorial de documentos que considera não apenas a frequência de palavras dentro de um documento, mas também a importância dessas palavras em relação a todo o corpus (conjunto de documentos).



SAIDA EXTRAÍDA DO CODIGO FEITO SOBRE O DATASET
https://huggingface.co/datasets/johnidouglas/twitter-sentiment-pt-BR-md-2-

	saiba	fÃ©	assim	bike	amo	menino	ngm	para	tu	vai	feio	eu	talvez	jÃ¡	nao	kaka	afffffff	definir	ando	da
0	0.193196	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0
1	0.000000	0.000000	0.0	0.0	0.0	0.225396	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.225396	0.0	0.225396	0.000000	0.0	0.0	0.0
2	0.000000	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0
3	0.000000	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.450791	0.000000	0.450791	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0
4	0.000000	0.169047	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.169047	0.000000	0.125725	0.0	0.000000	0.0	0.000000	0.169047	0.0	0.0	0.0

Análise de Desempenho do Classificador de Sentimentos com e sem Pré-processamento

Neste estudo, avaliamos o desempenho de um classificador de sentimentos baseado em regressão logística, utilizando vetorização TF-IDF, tanto com quanto sem a aplicação de pré-processamento de texto. O objetivo foi entender como a preparação dos dados influencia a eficácia do modelo na classificação de tweets como positivos ou negativos.

Metodologia

1. ***Carregamento e Amostragem dos Dados**: Utilizamos um conjunto de dados de sentimentos em tweets, que foi amostrado para reduzir o volume e acelerar o processo.*
2. ***Pré-processamento de Texto**: O pré-processamento envolveu várias etapas para limpar e normalizar os dados:*
    - ***Conversão para minúsculas**: Para uniformizar o texto e evitar que palavras em maiúsculas e minúsculas sejam tratadas como diferentes.*

- ***Remoção de caracteres especiais e menções****: Para eliminar elementos que não contribuem para a análise de sentimentos.*
- ***Remoção de stopwords****: Para excluir palavras comuns que não carregam informações significativas.*
- ***Aplicação de stemming****: Para reduzir palavras às suas raízes, facilitando a generalização.*

3. ***Vetorização****: Utilizamos o método TF-IDF para converter os textos em matrizes de características numéricas. Esta abordagem ajuda a quantificar a importância dos termos em relação ao corpus de documentos.*
4. ***Treinamento e Avaliação do Modelo****: O classificador de regressão logística foi treinado e avaliado em duas configurações:*
   - ***Sem Pré-processamento****: Utilizando diretamente os textos originais.*
   - ***Com Pré-processamento****: Utilizando textos que passaram pelas etapas de limpeza e normalização descritas.*

Resultados

Sem Pré-processamento:

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.72	0.75	0.74	401
1	0.74	0.71	0.73	399
accuracy			0.73	800
macro avg	0.73	0.73	0.73	800
weighted avg	0.73	0.73	0.73	800

Com Pré-processamento:

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.76	0.73	0.74	401
1	0.74	0.77	0.76	399
accuracy			0.75	800
macro avg	0.75	0.75	0.75	800
weighted avg	0.75	0.75	0.75	800

No estudo realizado, comparamos o desempenho de um classificador de sentimentos utilizando vetorização TF-IDF com e sem pré-processamento dos textos.

O modelo apresentou uma performance equilibrada nas métricas de avaliação, com uma acurácia geral de 73.12%. As métricas de precisão, recall e F1-Score foram similares, indicando uma capacidade de classificação razoável, mas com margem para melhorias.

Após a aplicação de pré-processamento, que incluiu a remoção de stopwords, stemming e limpeza do texto, o modelo mostrou uma melhoria em todas as métricas, alcançando uma acurácia de 74.62%. O pré-processamento ajudou a aprimorar a capacidade do modelo de distinguir entre as classes de sentimentos, refletido no aumento do F1-Score e das métricas de precisão e recall.

No contexto da comparação entre os métodos de vetorização **CountVectorizer** e **TF-IDF**, ambos aplicados com pré-processamento em textos de tweets, observamos as seguintes diferenças e resultados:

Vetorização CountVectorizer

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.72	0.81	0.75	401
1	0.77	0.69	0.72	399
accuracy			0.73	800
macro avg	0.74	0.74	0.74	800
weighted avg	0.74	0.74	0.74	800

Vetorização TF-IDF

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.76	0.73	0.75	401
1	0.74	0.77	0.76	399
accuracy			0.75	800
macro avg	0.75	0.75	0.75	800
weighted avg	0.75	0.75	0.75	800

A vetorização **TF-IDF** apresentou um desempenho superior em comparação com **CountVectorizer** para a tarefa de classificação de sentimentos em tweets. Isso demonstra a eficácia do TF-IDF em representar melhor a importância das palavras, melhorando a capacidade do modelo de capturar as nuances e especificidades dos textos. Por isso, TF-IDF é geralmente preferido quando a importância relativa das palavras é crucial para o desempenho do modelo.

Por fim, comparamos duas técnicas de pré-processamento de texto—**lemmatização** e **stemming**—usando a vetorização **TF-IDF**. O objetivo é avaliar qual técnica proporciona melhor desempenho na classificação de sentimentos dos tweets do dataset.

TF-IDF com Stemming:

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.75	0.72	0.74	401
1	0.73	0.75	0.74	399
accuracy			0.73	800
macro avg	0.74	0.74	0.74	800
weighted avg	0.74	0.74	0.74	800

TF-IDF com Lematização

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.73	0.74	0.73	401
1	0.73	0.73	0.73	399
accuracy			0.73	800
macro avg	0.73	0.73	0.73	800
weighted avg	0.73	0.73	0.73	800

Os resultados obtidos com a combinação de TF-IDF e Stemming revelam um desempenho geral sólido no modelo de classificação de sentimentos. Com uma acurácia de 0,74, um F1-Score de 0,74, e precisão e recall semelhantes (0,74 e 0,74, respectivamente), o modelo demonstra uma capacidade equilibrada em identificar sentimentos positivos e negativos. O desempenho da classe 0 (sentimento negativo) apresenta uma precisão de 0,75 e recall de 0,72, indicando uma ligeira vantagem na identificação de sentimentos negativos, com menos falsos positivos. Por outro lado, a

classe 1 (sentimento positivo) tem uma precisão de 0,73 e recall de 0,75, sugerindo uma maior eficiência na captura de sentimentos positivos. O F1-Score equilibrado para ambas as classes reflete um bom desempenho geral do modelo com Stemming.

Em contraste, o TF-IDF combinado com Lematização apresenta resultados ligeiramente inferiores. A acurácia alcançada foi de 0,73, com um F1-Score de 0,73, e a precisão e recall se mantiveram em torno de 0,73 e 0,73, respectivamente. O modelo com Lematização mostra uma precisão e recall consistentes para ambas as classes, indicando um desempenho estável, mas não tão eficaz quanto o modelo com Stemming. Para a classe 0, a precisão foi de 0,73 e o recall de 0,74, enquanto para a classe 1, ambos os valores foram semelhantes, em torno de 0,73.

Comparando os dois métodos, o TF-IDF com Stemming apresentou uma leve vantagem em termos de acurácia e F1-Score, sugerindo que o Stemming foi mais eficaz do que a Lematização para este conjunto de dados específico. O Stemming pode ter proporcionado uma representação mais compacta e útil das palavras para a tarefa de classificação, enquanto a Lematização, apesar de sua abordagem mais precisa de redução das palavras para suas formas base, não melhorou significativamente o desempenho do modelo.

Em conclusão, embora o Stemming tenha mostrado resultados ligeiramente superiores, é essencial considerar outras abordagens de pré-processamento e modelos de aprendizado de máquina para potencialmente otimizar ainda mais a performance. Experimentos adicionais com diferentes técnicas e parâmetros podem revelar novas oportunidades para aprimorar a precisão e a robustez do modelo de classificação de sentimentos.

4. Conclusões

Os resultados obtidos nas análises de pré-processamento e vetorização mostraram que o stemming, apesar de sua simplicidade, apresentou uma leve superioridade sobre a lematização quando combinado com TF-IDF. A acurácia, o F1-Score, a precisão e o recall foram ligeiramente melhores com o stemming do que com a lematização. Isso sugere que, no contexto do conjunto de dados analisado, o stemming foi um pouco mais eficaz para melhorar o desempenho do modelo.

Embora a lematização tenha mostrado um desempenho competitivo, a diferença em relação ao stemming não foi significativa. Esse resultado pode refletir limitações nos recursos disponíveis para o processamento linguístico do português ou características

específicas do dataset. O TF-IDF, como técnica de vetorização, se destacou em relação ao CountVectorizer, o que também contribuiu para a melhoria geral das métricas de avaliação.

Portanto, a escolha entre stemming e lematização deve levar em conta o contexto específico e os recursos linguísticos disponíveis. Quando a lematização não oferece uma vantagem clara, o stemming pode ser preferido, especialmente se apresentar um desempenho ligeiramente superior nas métricas de avaliação.

5. Próximos passos

Para otimizar ainda mais o desempenho do nosso modelo, podemos explorar algumas possibilidades. Uma delas é **refinar** o processo de preparação dos dados, utilizando técnicas como stemming e lematização de forma mais precisa. Além disso, podemos **ajustar** a seleção das palavras irrelevantes para melhorar a qualidade da representação textual.

Outra abordagem interessante seria **experimentar** com diferentes configurações dos parâmetros do modelo de regressão logística. Também podemos **comparar** os resultados com outros algoritmos, como máquinas de vetores de suporte e redes neurais.