

# AV1 - NLP

by Marilia Oliveira

# Objectivos

01

Classificar avaliações como positivas ou negativas

02

Comparar métodos de pré-processamento

03

Avaliar diferentes formas de vetorização

04

Identificar quais combinações geram maior acurácia de vetorização

# Dataset

## Buscapé Sentiment Analysis Dataset

### Dados e Volume

01

O dataset utilizado é o Corpus Buscapé, um extenso corpus de avaliações de produtos em português, coletado em 2013, composto por 84.991 avaliações reais do Buscapé, um site de busca de produtos e preços, fornecendo um rico ambiente de teste para NLP.

Essas avaliações são originadas de usuários brasileiros, representando interações genuínas entre consumidores e produtos/serviços.

Sua dimensão robusta permite que modelos aprendam padrões linguísticos de forma representativa, aumentando a confiabilidade dos resultados.

### Natureza e Complexidade Textual

02

O texto dessas avaliações é marcado por informalidade, apresentando-se cheio de ruídos: gírias locais, erros ortográficos e repetições frequentes.

Tal complexidade eleva o grau de desafio para técnicas de processamento de linguagem natural, tornando o corpus ideal para avanços e testes específicos para o português brasileiro.

# Amostras do Dataset

## Dataset com todas as colunas

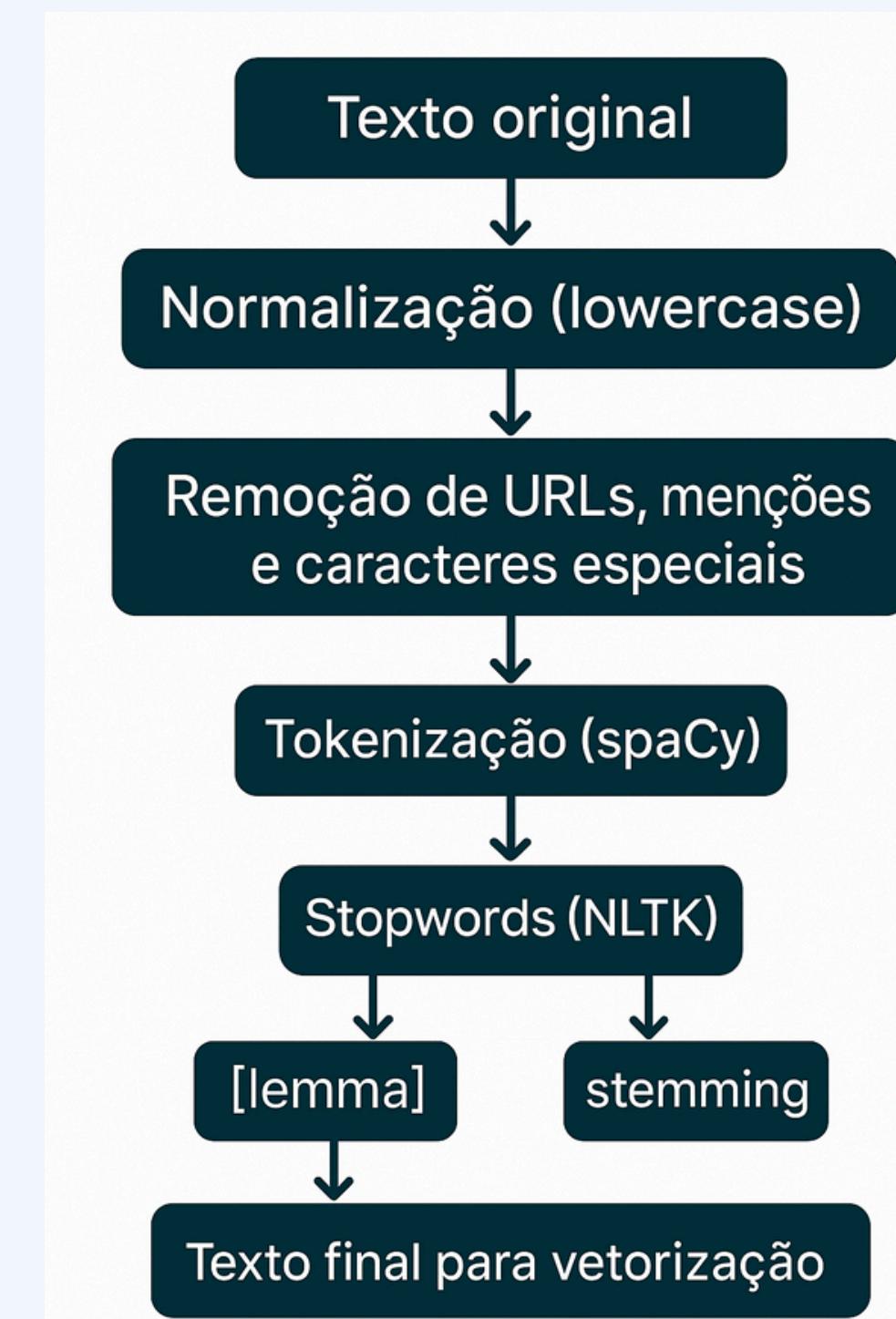
original_index	review_text	review_text_processed	review_text_tokenized	polarity	rating	kfold_polarity	kfold_rating
0 4_55516	Estou muito satisfeito, o visor é melhor do qu...	estou muito satisfeito, o visor e melhor do qu...	['estou', 'muito', 'satisfeito', 'visor', 'mel...', 'qu...']	1.0	4	1	1
1 minus_1_105339	"muito boa\n\nO que gostei: preco\n\nO que não...	"muito boa\n\nO que gostei: preco\n\nO que nao...	['muito', 'boa', 'que', 'gostei', 'preco', 'qu...', 'nao...']	1.0	5	1	1
2 23_382139	Rápida, ótima qualidade de impressão e fácil d...	rapida, otima qualidade de impressao e facil d...	['rapida', 'otima', 'qualidade', 'de', 'impres...', 'facil', 'd...']	1.0	5	1	1
3 2_446456	Produto de ótima qualidade em todos os quesito!	produto de otima qualidade em todos os quesito!	['produto', 'de', 'otima', 'qualidade', 'em', 'os', 'quesito!']	1.0	5	1	1
4 0_11324	Precisava comprar uma tv compatível com meu dv...	precisava comprar uma tv compativel com meu dv...	['precisava', 'comprar', 'uma', 'tv', 'compati...', 'com', 'meu', 'dv...', 'para', 'jogar', 'vídeo', 'jogos']	1.0	5	1	1

## Dataset com as colunas utilizadas

	review_text	polarity
0	Estou muito satisfeito, o visor é melhor do qu...	1.0
1	"muito boa\n\nO que gostei: preco\n\nO que não...	1.0
2	Rápida, ótima qualidade de impressão e fácil d...	1.0
3	Produto de ótima qualidade em todos os quesito!	1.0
4	Precisava comprar uma tv compatível com meu dv...	1.0
5	eu adorei este secador é muito bom,potente e d...	1.0
6	bom\n\nO que gostei: muitos aplicativos gratis...	NaN
7	positiva\n\nO que gostei: cumpriu com as espec...	1.0
8	Muito satisfeita com o telefone, atendeu as mi...	1.0
9	Estou muito contente com a utilização da maquin...	1.0

# Pré-processamento do Texto

# Fluxograma do Pré- processamento



# Descrição Detalhada das Etapas

## Tokenização e Stopwords

Tokenização é o processo de segmentar o texto em unidades básicas: palavras ou tokens, facilitando o tratamento individual das unidades textuais.

A seguir, a remoção de stopwords elimina palavras comuns como "e", "de", "a", que são frequentes mas carregam pouco valor semântico.

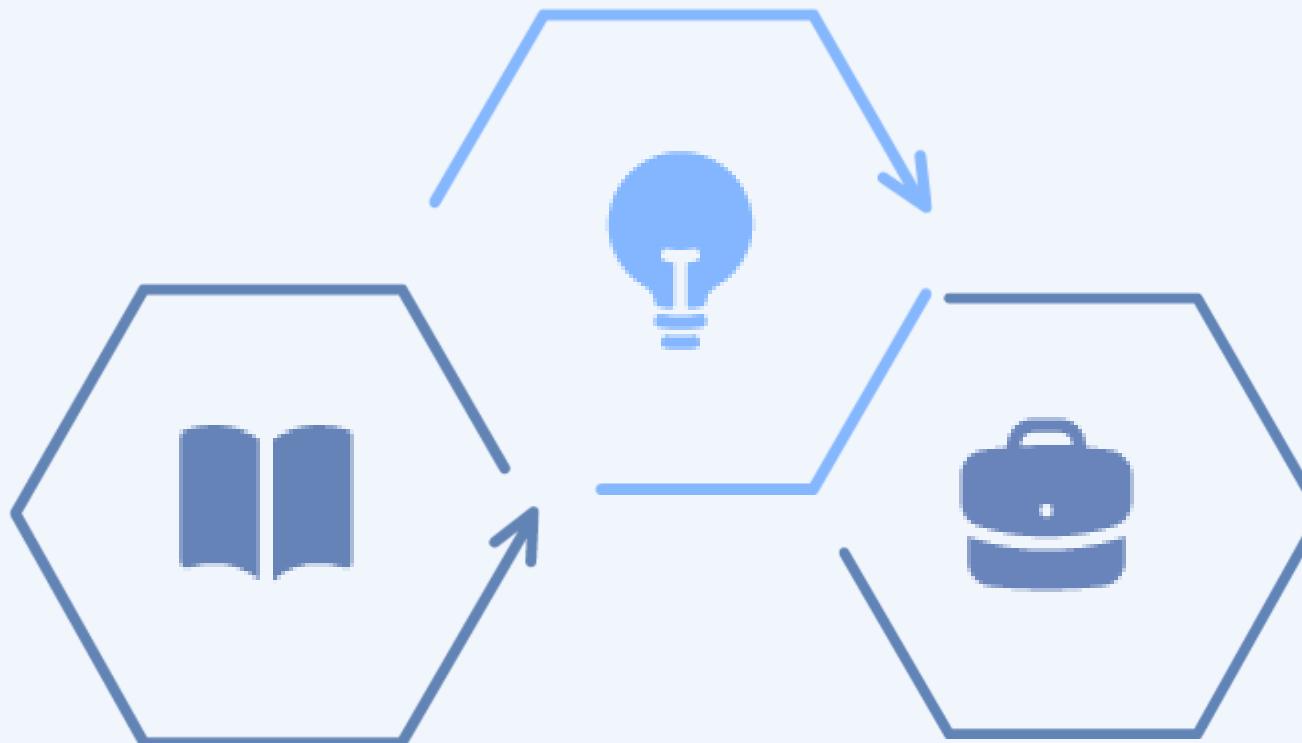
Este processo ajuda a focar apenas em termos que possuem impacto significativo para a classificação.

## Normalização e Limpeza

Normalização consiste em transformar todas as palavras para minúsculas, garantindo que "Ótimo" e "ótimo" sejam tratados igual.

Limpeza implica na eliminação de elementos que não acrescentam valor para a análise, como URLs, menções e demais símbolos.

Este passo reduz ruídos que poderiam distorcer a representação textual e comprometer a performance do modelo.



## Lematização e Stemming

Lematização reduz as palavras à sua forma base ou canônica, respeitando o contexto gramatical (ex: "correndo" vira "correr").

Stemming, por outro lado, elimina sufixos ou prefixos para encontrar o radical, sendo uma técnica mais agressiva e menos precisa (ex: "correndo" vira "corr").

Ambas as técnicas ajudam a diminuir a dimensionalidade do texto, melhorando o treinamento do classificador ao lidar com menos variantes da mesma palavra.

# Vetorização dos Textos

# ★ Métodos Utilizados ★

## Bag-of-Words (BoW)

01

O método Bag-of-Words transforma o texto em uma matriz de frequências, contabilizando quantas vezes cada palavra aparece.

Essa abordagem é simples, rápida e eficaz, mas não capta relações semânticas entre palavras.

Útil para detectar presença e intensidade de termos, porém pode gerar grandes matrizes esparsas.

## TF-IDF

02

A transformação TF-IDF pondera a frequência de palavras pela sua raridade no corpus, valorizando palavras mais discriminatórias.

Isso permite destacar termos que têm maior poder de diferenciação entre avaliações positivas e negativas.

Resulta em vetores que refletem a importância contextual dos termos.

# • Métodos Utilizados •

03

## TF-IDF com Chi<sup>2</sup>

A inclusão do Chi<sup>2</sup> atua como um método estatístico de seleção de features, escolhendo as palavras mais informativas para a separação das classes.

Essa combinação reduz dimensionalidade e melhora o desempenho do modelo ao retirar ruídos desnecessários.

Garante um conjunto mais limpo e relevante de atributos para classificação.

04

## Word2Vec

Word2Vec cria vetores densos e contínuos para palavras, capturando relações semânticas e contexto em um espaço vetorial.

É uma técnica avançada que permite que palavras similares estejam próximas no espaço vetorial, enriquecendo a análise.

Apesar de poderosa, pode requerer maior poder computacional e ajustes específicos.

# Classificação e Avaliação

# Modelo Utilizado

01

## Regressão Logística

Foi escolhido o modelo de Regressão Logística, conhecido por sua simplicidade e eficiência para problemas binários.

Esse modelo calcula a probabilidade de uma avaliação pertencer à classe positiva ou negativa com base nas features obtidas.

02

## Divisão do Conjunto de Dados

O dataset foi dividido em 80% para treino e 20% para teste, garantindo avaliação confiável do modelo em dados não vistos.

A divisão foi estratificada, ou seja, manteve a proporção original de avaliações positivas e negativas em ambos os conjuntos.

Esse procedimento é fundamental para evitar viés e garantir representatividade adequada de classes.

03

## Métrica de Avaliação

A métrica utilizada para avaliar o desempenho foi a acurácia, que mede a proporção de previsões corretas em relação ao total.

Embora simples, a acurácia é intuitiva e indicadora da eficácia geral do modelo, especialmente em datasets balanceados.

Outras métricas poderão ser utilizadas em futuros trabalhos para análise mais detalhada do desempenho.

# Resultados da Classificação

# Impacto do Pré-processamento

## Comparação com e sem Pré-processamento

A tabela compara a acurácia obtida com as features utilizando lematização (Lema) vs sem pré-processamento:

Features	Lema	Sem pré-proc
Count (Bag-of-Words)	0.9436	0.9476
TF-IDF	0.9474	0.95199
TF-IDF + Chi <sup>2</sup>	0.9471	0.9504
Word2Vec	0.9318	0.9303

Observa-se que o melhor resultado geral foi obtido com TF-IDF sem pré-processamento, alcançando 95,19% de acurácia. Esse comportamento evidencia que o pré-processamento nem sempre melhora a performance, especialmente em datasets grandes e ricos em variações linguísticas, onde elementos considerados “ruído” podem carregar sinais importantes para o modelo. Esse resultado reforça a importância de testar diferentes combinações de pré-processamento e vetorização, em vez de assumir que mais limpeza sempre leva a um melhor desempenho.

# Comparação entre Técnicas de Vetorização

## Resultados por Método

Método	Acurácia
TF-IDF	0.9474
TF-IDF + Chi <sup>2</sup>	0.9471
Count (Bag-of-Words)	0.9436
Word2Vec	0.9320

A técnica TF-IDF obteve o melhor desempenho geral, sendo a mais consistente para classificação das avaliações.

A seleção TF\_IDF com Chi<sup>2</sup> teve performance próxima, indicando boa relevância na escolha das melhores features.

Bag-of-Words e Word2Vec apresentaram resultados ligeiramente inferiores, mostrando que vetorização baseada em frequência e importância tem vantagem nesse cenário.

# Comparação entre Lematização e Stemming

Texto original:

Estou muito satisfeito, o visor é melhor do que eu imaginava, boas imagens, desing ultra fino. Pelo preço é um exelente aparelho.

O que gostei: Desing exelente, display, custo beneficio.

O que não gostei: Não tem como adicionar mais papeis de parede, bateria dura pouco.

Pré-processado (lemma):

satisfierto visor melhor imaginar bom imagem desing ultra fino preço exelente aparelho gostar desing exelente display custo beneficio gostar adicionar papel parede bateria

Pré-processado (stem):

satisfiit vis melhor imagin boa imag desing ultr fin preç exel aparelh gost desing exel display cust benefici gost adicon papal pared bat dur pouc

## Impacto do Pós-processamento

Pré-proc	Acurácia
Lemma	0.94744
Stem	0.948119

Os resultados indicam que o Stemming superou a lematização levemente, demonstrando que reduzir radicalmente as palavras pode trazer benefícios para esse tipo de classificação.

A diferença, embora pequena, merece consideração em projetos futuros, especialmente em contextos de textos informais.

# Conclusões

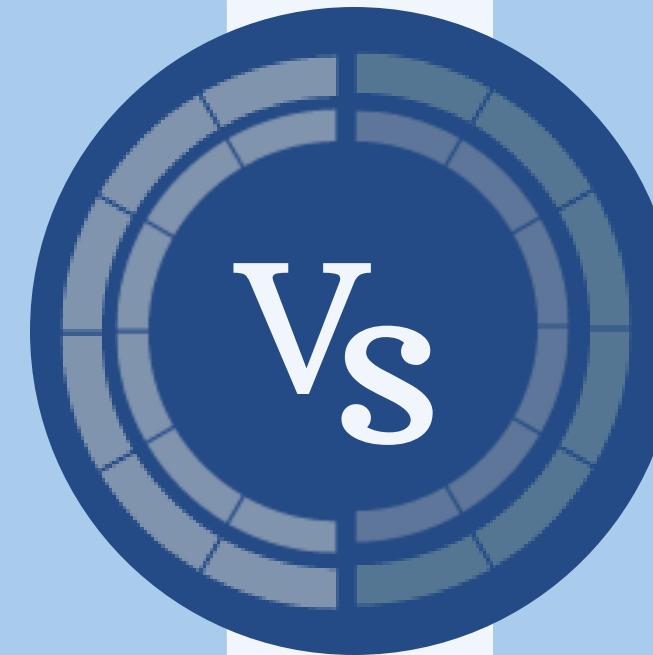
# Aprendizados Principais

## Destaque dos Resultados

O melhor desempenho foi atingido pelo modelo usando TF-IDF sem pré-processamento, com impressionantes 95,19% de acurácia.

Isso revela que o pré-processamento nem sempre é benéfico e pode, em certos casos, remover informações úteis para classificação.

A técnica de Stemming apresentou leve vantagem sobre a lematização, sugerindo que formas mais agressivas de redução podem ser eficazes para dados ruidosos.

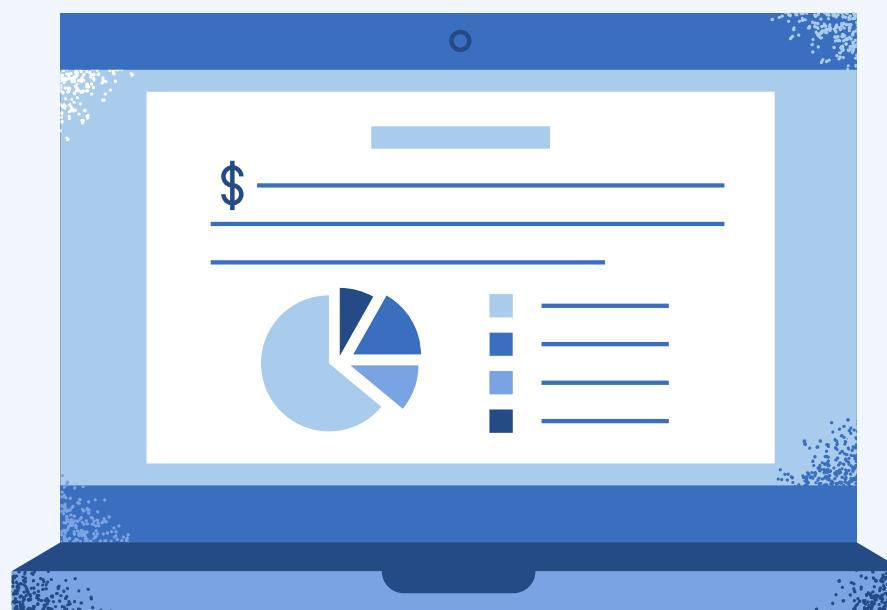


## Eficiência da Vetorização

O método TF-IDF mostrou-se o mais consistente e estável, sendo excelente para transformar texto em dados para modelos de classificação.

A escolha correta da vetorização é fundamental para garantir performance elevada, principalmente em textos informais e ruidosos.

# Próximos Passos



- 01 Testar modelos: SVM, Random Forest, XGBoost
- 02 Usar embeddings modernos: BERTimbau, mBERT, fastText
- 03 Adicionar métricas: F1-score, matriz de confusão
- 04 Criar API (Flask/FastAPI) ou dashboard (Gradio/Streamlit)

# Obrigada!

by Marilia Oliveira