

Escola Politécnica da Universidade de São Paulo
PECE - Programa de Educação Continuada
IAD-017 - Processamento de Linguagem Natural



Classificação de textos do Twitter "X" com spaCy

Prof. Marcos Lopes

Jorge Gomes da Silva Filho - 6259604

São Paulo
Maio de 2025

Objetivos: *O objetivo deste trabalho é aplicar técnicas de Processamento de Linguagem Natural (PLN) para classificar automaticamente o sentimento de tweets "X" escritos em português. A partir da análise supervisionada, busca-se distinguir textos com polaridade positiva e negativa, avaliando a eficácia de pré-processamentos linguísticos e do modelo spaCy como ferramenta de classificação textual.*

Palavras-chave: processamento de linguagem natural, classificação de sentimentos, spacy

Sumário

| | | |
|----------|---|-----------|
| 0 | INTRODUÇÃO | 3 |
| 1 | MÉTODOS USADOS | 4 |
| 1.1 | Corpus e Preparação dos Dados | 4 |
| 1.2 | Pré-processamento Textual com spaCy | 4 |
| 1.3 | Visualização e Inspeção dos Dados | 4 |
| 1.4 | Treinamento com spaCy TextCategorizer | 6 |
| 1.5 | Avaliação do Modelo | 7 |
| 2 | CONCLUSÕES | 9 |
| 3 | REPOSITÓRIO | 10 |

0 INTRODUÇÃO

Com o crescimento acelerado das redes sociais como canais de comunicação entre empresas e consumidores, plataformas como o Twitter ("X") se consolidaram como vitrines públicas de experiências, opiniões e críticas. Diariamente, milhares de usuários expressam sua satisfação ou frustração com produtos, serviços e marcas em mensagens curtas e de grande visibilidade. Nesse contexto, a capacidade de monitorar, analisar e interpretar automaticamente essas interações se tornou uma vantagem competitiva para organizações que buscam uma atuação mais proativa e estratégica.

A análise de sentimentos aplicada aos comentários em redes sociais permite identificar padrões emocionais nas falas dos clientes, classificando-as como positivas ou negativas. Essa classificação automatizada não apenas economiza recursos com análises manuais, mas também fornece insumos valiosos para tomadas de decisão em tempo real. A partir dos dados extraídos:

Times de marketing podem compreender melhor a percepção da marca, identificar embaixadores espontâneos e ajustar campanhas com base no humor predominante dos consumidores.

Áreas operacionais podem detectar falhas recorrentes nos serviços ou produtos, agindo preventivamente para evitar crises de reputação.

Gestores de risco podem mapear insatisfações que indiquem possíveis perdas de clientes, escaladas públicas negativas ou ameaças à imagem da empresa.

Ao empregar técnicas de Processamento de Linguagem Natural (PLN) e modelos de aprendizado supervisionado, como o implementado neste trabalho, é possível transformar grandes volumes de dados não estruturados em indicadores acionáveis. Assim, empresas que investem em inteligência analítica aplicada às redes sociais conseguem se posicionar com mais agilidade, empatia e assertividade frente às demandas do mercado digital.

1 MÉTODOS USADOS

A construção da análise de sentimentos seguiu um pipeline clássico de Processamento de Linguagem Natural (PLN), composto por cinco etapas principais: preparação dos dados, pré-processamento textual, visualização exploratória, treinamento do modelo e avaliação de desempenho. Cada fase foi executada com ferramentas modernas da biblioteca spaCy, amplamente reconhecida pela sua eficiência no processamento de textos em linguagem natural.

1.1 Corpus e Preparação dos Dados

Os dados utilizados foram extraídos do repositório público “Portuguese Tweets for Sentiment Analysis” disponível no Kaggle. O corpus contém mensagens publicadas por usuários no Twitter em língua portuguesa, previamente rotuladas como positivas (1) ou negativas (0).

Foram utilizadas duas bases:

Treinamento: com 50 mil tweets balanceados.

Teste: com cerca de 5 mil tweets para validação do modelo.

Antes do treinamento, as colunas irrelevantes (como *id*, *data do tweet* e *query usada*) foram removidas, mantendo-se apenas o conteúdo textual e o rótulo de sentimento.

1.2 Pré-processamento Textual com spaCy

O pré-processamento tem papel fundamental na redução do ruído e na normalização da linguagem informal típica do Twitter. As transformações aplicadas incluem:

Normalização: transformação de todo o texto para letras minúsculas, removendo variações de capitalização.

Remoção de elementos não informativos: menções de usuários (*@usuário*), URLs, espaços múltiplos e emoticons (que foram convertidos em palavras-chave como *emocao-positiva* e *emocaonegativa*).

Tokenização e Lematização: com o modelo *en_core_web_sm* da spaCy, as palavras foram transformadas em suas formas canônicas (lemmas), o que ajuda a reduzir a dimensionalidade do vocabulário.

Stopwords e pontuação: palavras funcionais como “de”, “com”, “que” foram removidas com o auxílio da lista de stopwords da *spaCy.lang.pt*, assim como sinais de pontuação.

Esse conjunto de técnicas garante que os textos estejam limpos e estruturados, prontos para a vetorização e classificação.

1.3 Visualização e Inspeção dos Dados

A inspeção visual dos dados foi realizada utilizando bibliotecas como Pandas, Seaborn e Matplotlib, com o objetivo de entender a distribuição dos rótulos, detectar possíveis valores ausentes e validar o balanceamento das classes.

Gráficos gerados:

Gráfico de barras com a distribuição de sentimentos.

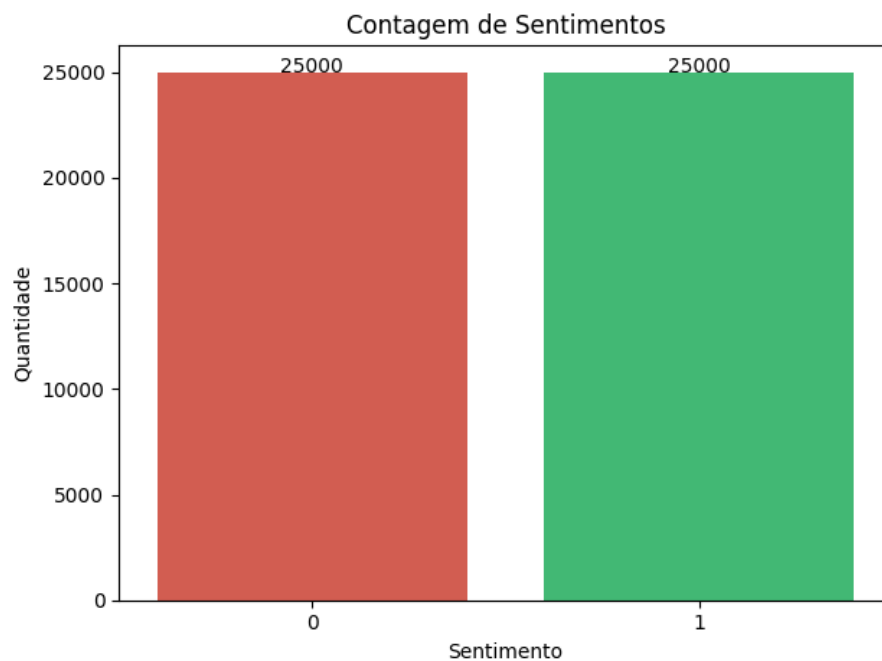


Figura 1: Distribuição dos sentimentos rotulados como negativos (0) e positivos (1)

Heatmaps de valores nulos para validação da integridade das bases.

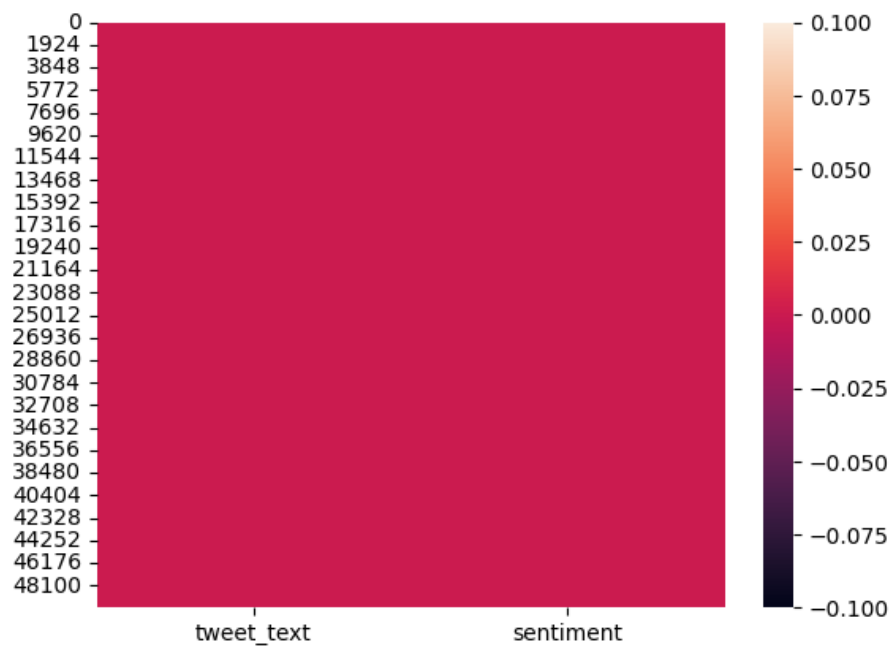


Figura 2: Ausencia de dados ausentes nos atributos tweet_text e sentiment do conjunto de dados de treino e de teste

Essas análises garantem que os dados estejam em condição ideal para treinar um modelo confiável.

1.4 Treinamento com spaCy TextCategorizer

O modelo de classificação foi construído com o componente TextCategorizer da spaCy, que permite a criação de um pipeline de classificação de texto binário. Esse componente transforma o texto em representações vetoriais internas e aplica uma camada densa para prever a probabilidade de cada classe (positiva ou negativa).

Etapas realizadas:

- Criação do pipeline com `nlp.add_pipe("textcat")`.
- Definição das labels "POSITIVE" e "NEGATIVE".
- Treinamento com batches iterativos utilizando o método `update()` e função de perda (loss).

Registro da curva de erro ao longo dos batches, demonstrando a convergência do modelo.

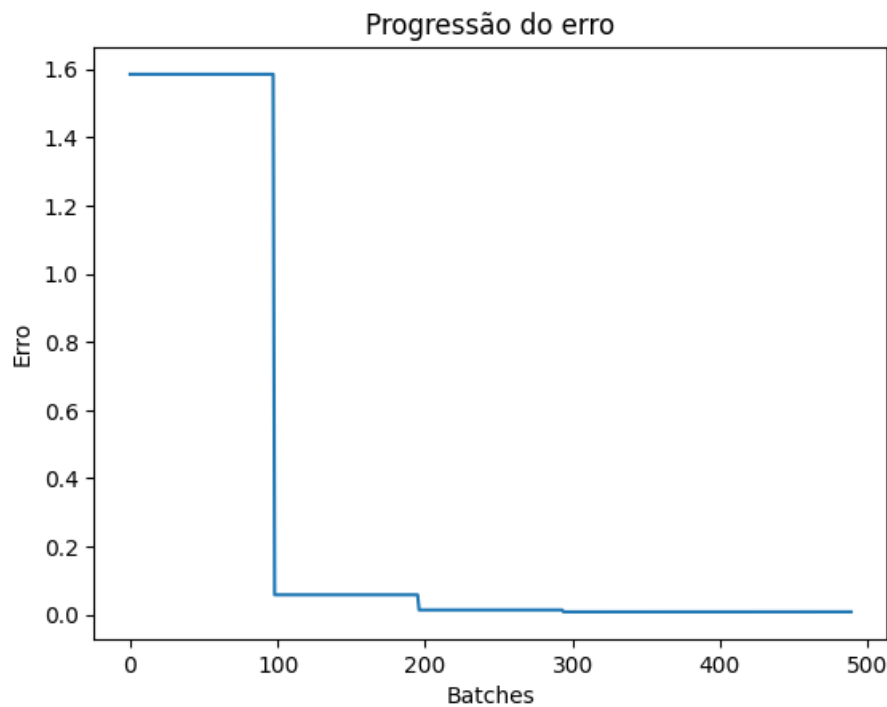


Figura 3: Progressão do erro durante o treinamento

Além das métricas quantitativas, a curva de progressão do erro durante o treinamento do modelo indica uma aprendizagem eficaz. Observa-se uma queda abrupta no erro após cerca de 100 batches, com estabilização próxima de zero até o final do processo.

Isso sugere que o modelo rapidamente aprendeu a distinguir os padrões entre os tweets positivos e negativos, o que reforça a eficiência da combinação entre o pré-processamento aplicado e o classificador baseado em spaCy.

No entanto, a ausência de oscilação pode indicar overfitting leve, especialmente se os dados de teste forem substancialmente diferentes dos de treino. Como melhoria futura, recomenda-se:

- Avaliação com k-fold cross-validation.
- Inclusão de dados mais variados e desbalanceados para teste de robustez.
- Uso de regularização ou dropout, caso redes neurais profundas sejam aplicadas.

1.5 Avaliação do Modelo

Após o treinamento, o modelo foi avaliado com o conjunto de teste, utilizando métricas clássicas de classificação:

Matriz de Confusão: representando verdadeiros positivos, negativos e erros.

Acurácia: proporção de classificações corretas.

Precisão, Recall e F1-Score: para entender o equilíbrio entre acertos e erros.

A matriz abaixo representa os resultados da classificação binária dos tweets em positivo (1) e negativo (0):

| | Prev. Negativo | Prev. Positivo |
|---------------|----------------|----------------|
| Real Negativo | 2500 | 7 |
| Real Positivo | 4 | 2500 |

Tabela 1: Matriz de Confusão

Interpretação:

Verdadeiros Negativos (TN): 2500 — tweets negativos corretamente classificados como negativos.

Falsos Positivos (FP): 7 — tweets negativos classificados incorretamente como positivos.

Falsos Negativos (FN): 4 — tweets positivos classificados incorretamente como negativos.

Verdadeiros Positivos (TP): 2500 — tweets positivos corretamente classificados como positivos.

O desempenho foi avaliado com base em métricas clássicas de classificação:

- **Acurácia:**

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{2500 + 2500}{2500 + 2500 + 7 + 4} \approx 0.9978 \quad (99.78\%)$$

- **Precisão (classe 1):**

$$\frac{TP}{TP + FP} = \frac{2500}{2500 + 7} \approx 0.9972 \quad (99.72\%)$$

- **Recall (classe 1):**

$$\frac{TP}{TP + FN} = \frac{2500}{2500 + 4} \approx 0.9984 \quad (99.84\%)$$

- **F1-Score:**

$$2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \approx 0.9978 \quad (99.78\%)$$

2 CONCLUSÕES

A análise de sentimentos realizada com spaCy e PLN mostrou-se eficaz para classificar sentimentos em tweets em português, especialmente após um bom pré-processamento.

O modelo foi capaz de identificar padrões linguísticos relevantes para distinguir entre sentimentos positivos e negativos.

Pontos fortes:

- Pipeline claro e bem estruturado com spaCy;
- IPré-processamento eficaz para dados ruidosos como tweets.

O que poderia ser melhorado:

- Adoção de embeddings mais robustos (ex: word2vec ou transformers como BERT-Timbau);
- Inclusão de uma terceira classe (neutra);
- Experimentar sumarização ou análise de tópicos para dar mais contexto às decisões do modelo.

3 REPOSITÓRIO

Repositório no GitHub contendo o corpus utilizado para treino e teste, além do notebook com o modelo de classificação desenvolvido: https://github.com/jgfilho/Classifica-o-Textos-do-Twitter-com-spaCy_IA/tree/main