**UNIVERSIDADE PRESBITERIANA MACKENZIE**

BRENNO MONTEIRO DE OLIVEIRA
BRUNO FERNANDES MASCARINI
LUCAS CAMARGO SPINELI
MARINA CAMARGO SPINELI

PROJETO APLICADO II

# ANÁLISE DAS PERCEPÇÕES DO CONTEÚDO DO TWITTER

São Paulo
2025

BRENNO MONTEIRO DE OLIVEIRA
BRUNO FERNANDES MASCARINI
LUCAS CAMARGO SPINELI
MARINA CAMARGO SPINELI

# PROJETO APLICADO II

# ANÁLISE DAS PERCEPÇÕES DO CONTEÚDO DO TWITTER

Projeto Aplicado II, do terceiro período do curso de Tecnologia em Ciência de Dados da Universidade Presbiteriana Mackenzie.

Prof. Dr. Felipe Albino dos Santos

São Paulo
2025

**SUMÁRIO**

## 1. INTRODUÇÃO

O suporte ao cliente nas redes sociais tem se tornado uma das principais formas de interação entre empresas e consumidores, e o X (Twitter), com sua natureza dinâmica e interativa, se destaca como uma das plataformas mais utilizadas para esse fim. Com mais de 3 milhões de tweets e respostas de grandes marcas, o conjunto de dados analisado neste estudo oferece uma oportunidade de explorar a comunicação entre empresas e consumidores, além de fornecer insights sobre a eficácia das práticas de suporte ao cliente modernas.

Este trabalho se propõe a investigar esses dados, visando compreender melhor as interações em tempo real, a análise da percepção do consumidor e o impacto das respostas oferecidas pelas empresas. A análise desses dados é relevante não apenas para aprimorar as estratégias de atendimento ao cliente, mas também para contribuir com a evolução da compreensão da linguagem natural e o desenvolvimento de modelos conversacionais. Ao longo deste estudo, exploraremos as características do conjunto de dados, suas vantagens em relação a outras corpora conversacionais e as possibilidades de aplicação, incluindo a previsão de respostas, o agrupamento de tópicos e a identificação de novos problemas que possam surgir no contexto do atendimento online.

Toda a documentação, ficará disponível no github, no link a seguir:
https://github.com/Brennu/Projeto-Aplicado-II

## 2. CRONOGRAMA

O cronograma a seguir ilustra as etapas a serem realizadas neste projeto. Cada integrante do grupo está representado por uma cor, indicando as respectivas responsabilidades de cada um; o "X" marca a expectativa da semana do mês de realização de cada parte do trabalho.

| LEGENDA | | | | | |
|---|---|---|---|---|---|
| | BRENNO MONTEIRO DE OLIVEIRA | | LUCAS CAMARGO SPINELI | | TODOS |
| | BRUNO FERNANDES MASCARINI | | MARINA CAMARGO SPINELI | | |

| CRONOGRAMA - PROJETO APLICADO II | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Lista de Atividades | RESP. | FEVEREIRO | | | | MARÇO | | | | ABRIL | | | | MAIO | | | | |
| Etapa 1 | Definição da Organização | | | X | | | | | | | | | | | | | | | |
| | Apresentação do Dados (Metadados) | | | X | | | | | | | | | | | | | | | |
| | Objetivos e Metas | | | X | | | | | | | | | | | | | | | |
| | Cronograma de Atividades | | | | X | | | | | | | | | | | | | | |
| | Estrutura do Documento | | | | X | X | | | | | | | | | | | | | |
| | Link para o Github | | | | X | | | | | | | | | | | | | | |
| Etapa 2 | Definição da Linguagem de Programação | | | | | | X | | | | | | | | | | | | |
| | Análise Exploratória da Base de Dados | | | | | | | X | X | | | | | | | | | | |
| | Tratamento da Base de Dados | | | | | | | X | | | | | | | | | | | |
| | Definição e Descrição das Bases Teóricas dos Métodos | | | | | | | | | X | | | | | | | | | |
| | Definição e Descrição da Acurácia | | | | | | | | | X | | | | | | | | | |
| Etapa 3 | Aplicação do Método Analítico | | | | | | | | | | X | | | | | | | | |
| | Medidas de Acurácia | | | | | | | | | | | X | | | | | | | |
| | Descrição dos Resultados Preliminares | | | | | | | | | | | | X | | | | | | |
| | Esboço do Storytelling | | | | | | | | | | | | | X | | | | | |
| Etapa 4 | Relatório Técnico | | | | | | | | | | | | | | X | | | | |
| | Apresentação do Storytelling | | | | | | | | | | | | | | | X | X | | |
| | Repositório no GitHub | | | | | | | | | | | | | | | | X | | |
| | Vídeo da Apresentação | | | | | | | | | | | | | | | | X | X | |

## 3. OBJETIVOS DO ESTUDO

O presente estudo tem como objetivo desenvolver um modelo capaz de analisar e classificar sentimentos em tweets, identificando padrões e tendências nas opiniões expressas pelos usuários. Para isso, será realizada uma investigação detalhada dos dados, incluindo a limpeza e o pré-processamento dos textos, a aplicação de algoritmos de aprendizado de máquina e a avaliação do desempenho dos modelos. Ao final, pretende-se construir uma ferramenta eficiente para prever a polaridade dos sentimentos em novas postagens no Twitter, contribuindo para uma melhor compreensão do comportamento dos usuários na plataforma.

## 4. CONTEXTO DE ESTUDO

Com o crescimento das redes sociais, o Twitter tornou-se uma das principais plataformas onde os usuários compartilham suas opiniões sobre uma ampla variedade de assuntos, como produtos, serviços, eventos e figuras públicas. A grande quantidade de postagens diárias cria um ambiente rico em informações, mas ao mesmo tempo desafiador para análise manual. Diante desse cenário, a análise automatizada de sentimentos surge como uma ferramenta essencial para compreender as percepções dos usuários de forma rápida e eficiente.

Empresas e organizações utilizam a análise de sentimentos para monitorar a recepção de seus produtos, avaliar a satisfação do público e até mesmo antecipar crises de imagem. Ao identificar padrões nas emoções expressas nos tweets, é possível obter insights valiosos que auxiliam na tomada de decisões estratégicas.

Além disso, com o avanço do processamento de linguagem natural e aprendizado de máquina, é possível identificar nuances emocionais complexas, como sarcasmo e ironia. Isso torna a análise de sentimentos mais precisa e permite que empresas respondam rapidamente a problemas de clientes no Twitter. Assim, a análise de sentimentos se torna uma ferramenta estratégica, ajudando as organizações a adaptar suas estratégias e melhorar a experiência do cliente.

## 5. APRESENTAÇÃO DA EMPRESA

O X, anteriormente conhecido como Twitter, é uma plataforma de mídia social dedicada à comunicação em tempo real. Fundada em 2006 por Jack Dorsey, Biz Stone e Evan Williams, a empresa tem sede em São Francisco, Califórnia, EUA.

**Missão:** A missão do X é promover e proteger a conversa pública — ser a praça pública da internet.

**Visão:** Ser o melhor aplicativo para tudo que gira em torno de áudio, vídeo, mensagens, pagamentos e serviços bancários

**Valores:** Os valores do X refletem seu compromisso com a comunicação aberta e inclusiva. A empresa incentiva a comunicação sem medo para construir confiança, garantindo um ambiente onde os usuários possam se expressar livremente. Além disso, busca defender e respeitar a voz do usuário, promovendo um espaço de diálogo autêntico.

**Segmento de atuação:** O X opera no setor de tecnologia e redes sociais, oferecendo um espaço para debates, notícias, entretenimento e interação entre usuários, empresas e governos. Seu modelo de negócios inclui publicidade digital, assinaturas e monetização de conteúdo, atendendo a um público global.

**Market Share:** O X sofreu uma desvalorização de aproximadamente 79% desde a aquisição por Elon Musk, reduzindo seu valor de mercado para cerca de US$ 9,4 bilhões. Relatórios indicam que o número de usuários ativos mensais do X caiu para aproximadamente 335 milhões em 2024, uma redução em relação aos 368 milhões anteriores.

**Iniciativas em Ciência de Dados:** O X utiliza técnicas de ciência de dados para analisar sentimentos e tendências nas postagens dos usuários, auxiliando empresas a entenderem a percepção de suas marcas e produtos.

## 6. APRESENTAÇÃO DO DATASET

O dataset utilizado neste estudo "Twitter Sentiment Analysis" contém informações extraídas do Twitter e apresenta quatro tipos de dados principais. Cada entrada no conjunto de dados possui um identificador único que diferencia os tweets entre si. Além disso, há uma coluna que representa a categoria ou tópico principal abordado na postagem, como o nome de um produto, jogo ou assunto específico discutido pelos usuários.

Outro dado relevante presente no dataset é a classificação do sentimento expresso no tweet. Cada postagem é rotulada de acordo com a emoção predominante, podendo ser positiva, negativa, neutra ou irrelevante. Por fim, o conjunto de dados inclui o próprio texto do tweet, que contém a opinião do usuário sobre o tema abordado. Esses dados permitirão a aplicação de técnicas de Processamento de Linguagem Natural para compreender e classificar automaticamente os sentimentos expressos nas postagens.

## 7. SELEÇÃO DE BIBLIOTECAS EM PYTHON

Nós escolhemos Python para nossa análise exploratória e tratamento dos dados porque é uma linguagem poderosa, flexível e amplamente utilizada na área de ciência de dados. Com bibliotecas como Pandas e NumPy, conseguimos manipular e limpar os dados de forma eficiente, enquanto Matplotlib e Seaborn nos ajudaram a visualizar padrões e insights de maneira clara. Além disso, utilizamos a Scikit-Learn, uma biblioteca fundamental para a implementação de algoritmos de machine learning e avaliação do desempenho do modelo, fornecendo métricas como acurácia, precisão e F1-score. Outro ponto importante é a integração com ferramentas de banco de dados e aprendizado de máquina, o que nos permitiu trabalhar desde a preparação dos dados até análises mais avançadas, tudo no mesmo ambiente. Por fim, a grande comunidade e os recursos disponíveis online foram essenciais para resolver desafios que surgiram durante o projeto.

## 8. MÉTODOS ANALÍTICOS

Durante as escolhas dos métodos optamos por uma abordagem tradicional, mas bem fundamentada: vetorização via CountVectorizer com n-grams e classificação com LogisticRegression da biblioteca Scikitlearn. Essa decisão levou em conta fatores como interpretabilidade, tempo de execução e facilidade de manutenção do modelo.

Visualizações com Seaborn e WordCloud foram usadas para revelar padrões linguísticos e insights por classe (ex: frequência de termos em tweets negativos), enquanto o desempenho foi avaliado pela métrica de acurácia.

### 8.1 Carregamento de Dados

Os dados foram importados de arquivos CSV e divididos em dois conjuntos: train_data para o treinamento do modelo e val_data para sua validação. As colunas foram renomeadas para id, information, type e text, com o objetivo de padronizar a estrutura e facilitar a manipulação nas etapas subsequentes. A coluna type contém a categoria de sentimento atribuída a cada tweet e será utilizada como variável alvo na modelagem. Após a separação, os dados foram inspecionados visualmente para garantir que o carregamento e a estruturação ocorreram corretamente, assegurando que os conjuntos estejam prontos para o pré-processamento e desenvolvimento do modelo de classificação.

### 8.2. Pré-processamento de Texto

O pré-processamento textual foi realizado para padronizar e limpar os dados, reduzindo ruídos e garantindo maior eficiência na etapa de modelagem. Inicialmente, todos os textos foram convertidos para letras minúsculas, eliminando distinções desnecessárias entre palavras com a mesma grafia em diferentes capitalizações. Em seguida, assegurou-se que todos os registros fossem do tipo string, prevenindo erros causados por entradas numéricas ou formatos inconsistentes.

Na etapa seguinte, aplicou-se uma limpeza com expressões regulares, removendo caracteres especiais, pontuações, emojis, URLs e outros elementos que não contribuem semanticamente para a análise. Essa filtragem resultou em textos mais uniformes e relevantes para a tarefa de classificação. Ao final, os

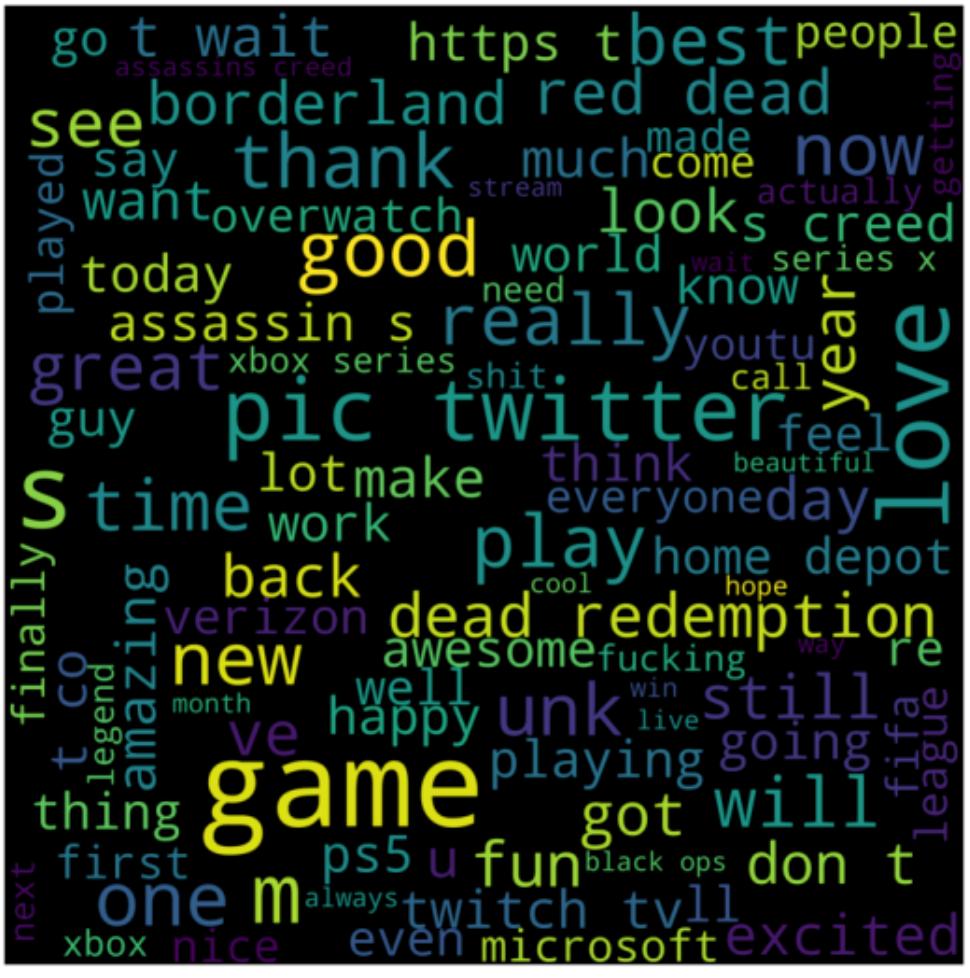
dados textuais apresentaram-se consistentes e prontos para as fases de vetorização e construção do modelo de análise de sentimentos.

### 8.3. Visualizações (Palavras e Distribuição por Marca)

Para identificar padrões linguísticos e analisar o engajamento por marca, foram geradas visualizações a partir dos tweets classificados. A etapa foi dividida em duas abordagens: nuvens de palavras por tipo de sentimento, e gráficos de distribuição por marca e tipo. As nuvens de palavras permitem observar os termos mais recorrentes por categoria de sentimento, enquanto os gráficos facilitam a análise comparativa entre marcas.

### 8.4. Positive

Gerada a nuvem de palavras para tweets classificados como Positive, a partir do texto previamente limpo e padronizado. Termos como "love", "game", "good", "play" e "happy" foram os mais frequentes, indicando associação com experiências positivas e satisfação. A variedade lexical observada sugere ampla diversidade na forma de expressar sentimentos positivos.
Nuvem:



### 8.5. Negative

A nuvem de palavras dos tweets Negative evidenciou alta frequência de termos como "fuck", "shit", "problem" e "fix", além de menções a marcas como "facebook" e "verizon". Os resultados indicam forte presença de reclamações e insatisfação, principalmente relacionadas a falhas técnicas ou serviços.

Nuvem:

### 8.6. Irrelevant

Na categoria Irrelevant, destacaram-se palavras como "ban", "player", "fuck" e "game". A semelhança lexical com a categoria Negative pode gerar confusão na modelagem, exigindo maior rigor na distinção entre essas classes durante o pré processamento ou rotulagem.

Nuvem:

### 8.7. Neutral

A nuvem de palavras dos tweets Neutral apresentou alta incidência de termos relacionados a tecnologia e mídia, como "game", "twitch", "tv", "amazon" e "microsoft", além de links e referências visuais ("https", "pic", "twitter"). Os dados indicam um conteúdo majoritariamente informativo e descritivo, sem polaridade emocional clara.

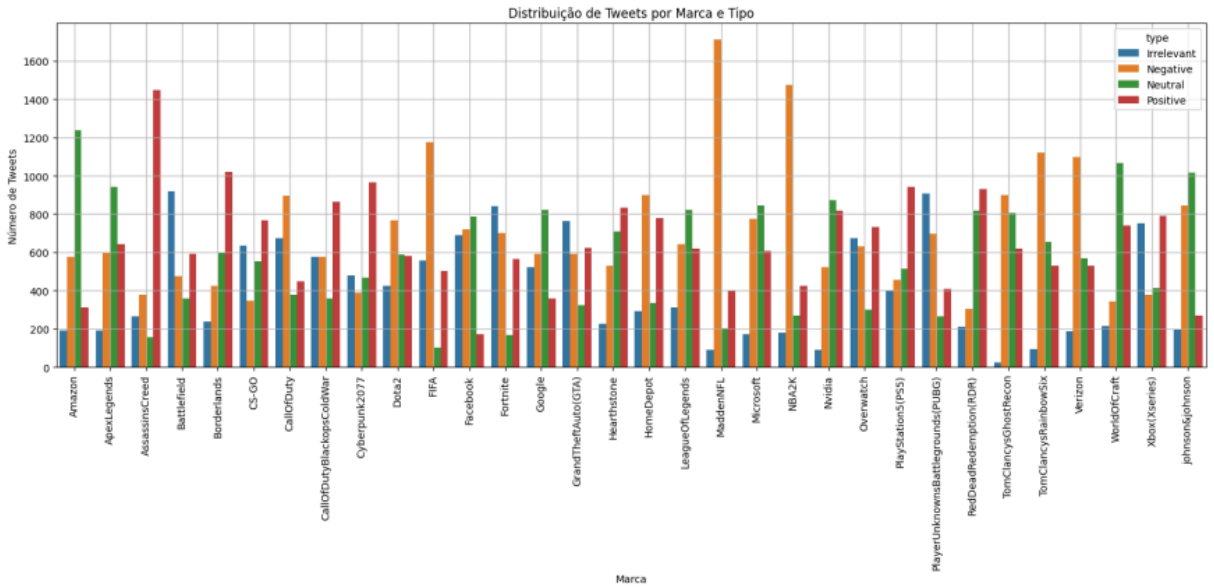
Nuvem:

### 8.8. Distribuição por Marca

Foi realizado o agrupamento de tweets por marca, considerando a contagem de registros únicos (id). A marca Amazon apresentou predominância de tweets Neutral (1236), seguida por Negative (576) e Positive (312), sugerindo foco em conteúdos informativos e promocionais, mas com presença relevante de críticas. Outras marcas, como ApexLegends, mostraram distribuição mais concentrada em categorias específicas.

### 8.9. Distribuição por Marca e Tipo

A visualização em barras revelou padrões distintos entre marcas. MaddenNFL e NBA2K apresentaram alta incidência de tweets negativos, sugerindo maior insatisfação. Marcas como Amazon, Google e Microsoft mostraram distribuição mais equilibrada entre os tipos. A categoria Neutral foi predominante na maioria das marcas, indicando tendência a postagens descritivas ou informativas.



### 8.10. Tokenização

O texto dos tweets foi segmentado em palavras individuais (tokens). Inicialmente, cada tweet foi transformado em uma lista de tokens. Em seguida, todas as listas foram unificadas para análise do vocabulário total, resultando em 30.436 tokens únicos, evidenciando alta dimensionalidade. Por fim, foram inspecionadas saídas tokenizadas como exemplo da estrutura que será utilizada nas etapas seguintes de pré-processamento e modelagem.

### 8.11. Modelo de Regressão Logística

Nesta etapa, foi desenvolvido um modelo de classificação de sentimentos utilizando Regressão Logística. A estratégia adotada incluiu a vetorização dos textos por meio da técnica Bag of Words com n-grams de até 4 palavras, seguida pela divisão dos dados e treinamento do modelo. As etapas foram segmentadas em três partes: configuração e vetorização, divisão e análise das classes, e treinamento com avaliação dos resultados.

#### 8.11.1. Configuração Inicial e Vetorização

Foi adotada a abordagem Bag of Words com n-grams de 1 a 4 palavras, sem remoção de stopwords, para preservar contextos relevantes. A vetorização foi aplicada exclusivamente ao conjunto de treino, evitando vazamento de informação. O resultado gerou uma matriz esparsa com 14.937 tweets e 1.427.378 atributos únicos, totalizando mais de 803 mil ocorrências, caracterizando um espaço vetorial de alta dimensionalidade.

#### 8.11.2. Divisão dos Dados e Análise de Classes

Os dados foram divididos em treino (80%) e teste (20%) com estratificação, assegurando a proporcionalidade das classes. Os rótulos foram definidos com base nos tipos de sentimento. A distribuição revelou leve desbalanceamento: Negative (29,9%), Positive (28,2%), Neutral (24,6%) e Irrelevant (17,2%). Esse cenário pode afetar o desempenho em métricas específicas, como recall para classes minoritárias.

### 8.11.3. Treinamento e Avaliação do Modelo

Foi treinado um modelo de Regressão Logística com regularização (C = 0.9) e limite de 1500 iterações. Nos dados de teste, o modelo alcançou acurácia de 90,79%. Em validação externa, a acurácia foi de 98,6%, sugerindo boa performance, com possibilidade de sobreajuste dependendo da composição do conjunto de validação. O modelo demonstrou alta eficácia na classificação de sentimentos utilizando Bag of Words com n-grams.

## 8.12 Código dos métodos utilizados

```
import numpy as np

import pandas as pd

pd.options.mode.chained_assignment = None

# Bibliotecas para visualização

from wordcloud import WordCloud

import matplotlib.pyplot as plt

import seaborn as sns


# Bibliotecas para NLP e modelagem

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score

from sklearn.preprocessing import LabelEncoder

import re
```

```
import nltk

from nltk import word_tokenize

nltk.download('stopwords')

nltk.download('punkt_tab')

# Carregar dataset de validação

val=pd.read_csv("https://raw.githubusercontent.com/Brennu/Projeto-Aplicado-II/refs

heads/main/Dataset/twitter_validation.csv", header=None)


# Carregar dataset de treinamento

train=pd.read_csv("https://raw.githubusercontent.com/Brennu/Projeto-Aplicado-II/refs
/heads/main/Dataset/twitter_training.csv", header=None)


# Renomeando colunas

train.columns = ['id', 'information', 'type', 'text']

val.columns = ['id', 'information', 'type', 'text']

train_data=train

val_data=val


# Visualizar primeiras linhas

print("Dados de Treino:")

display(train_data.head())
```

```
print("\nDados de Validação:")

display(val_data.head())

#Transforma todo o texto para lowercase para padronização

#Isso evita diferenciação entre palavras com caixas diferentes

train_data["lower"]=train_data.text.str.lower()

val_data["lower"]=val_data.text.str.lower()


# Converte todos os valores para string, incluindo números isolados (como '2')

# Isso é necessário pois alguns tweets podem conter apenas números

train_data["lower"]=[str(data) for data in train_data.lower]

val_data["lower"]=[str(data) for data in val_data.lower]

#Remove caracteres especiais, pontuações e símbolos

#Mantém apenas letras, números e espaços

#Importante para tweets que podem conter erros de digitação ou formatação

train_data["lower"]=train_data.lower.apply(lambda x: re.sub('[^A-Za-z0-9 ]+', ' ', x))

val_data["lower"]=val_data.lower.apply(lambda x: re.sub('[^A-Za-z0-9 ]+', ' ', x))


#Palavras mais frequentes em tweets positivos incluem termos como "love" e "game", além de outras palavras associadas a sentimentos positivos.

# A diversidade lexical é maior nesta categoria.

word_cloud_text = ''.join(train_data[train_data["type"]=="Positive"].lower)
```

```
wordcloud = WordCloud(

max_font_size=100,

max words=100,

background_color="black",

scale=10,

width=800,

height=800

).generate(word_cloud_text)

plt.figure(figsize=(10,10))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis("off")

plt.show()

# Tweets negativos apresentam palavrões com frequência, além de menções a empresas/games específicos como 'facebook' e 'eamaddennfl'.

# Isso pode indicar reclamações direcionadas a essas marcas.

word_cloud_text = ''.join(train_data[train_data["type"]=="Negative"].lower)

wordcloud = WordCloud(

 max_font_size=100,

max_words=100,

 background_color="black",

scale=10,
```

```
width=800,
height=800
).generate(word_cloud_text)

plt.figure(figsize=(10,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()

# Padrão similar aos tweets negativos, o que pode impactar a performance do modelo,sugerindo possível sobreposição entre essas categorias.
word_cloud_text = ''.join(train_data[train_data["type"]=="Irrelevant"].lower)
wordcloud = WordCloud(
max_font_size=100,
max_words=100,
background_color="black",
scale=10,
width=800,
height=800
).generate(word_cloud_text)
```

```
plt.figure(figsize=(10,10))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis("off")

plt.show()

# Apresenta perfil lexical distinto, com quase nenhum palavrão e palavras-chave diferentes das outras categorias, indicando maior neutralidade.

word_cloud_text = ''.join(train_data[train_data["type"]=="Neutral"].lower)

wordcloud = WordCloud(

max_font_size=100,

max_words=100,

background_color="black",

scale=10,

width=800,

 height=800

).generate(word_cloud_text)

plt.figure(figsize=(10,10))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis("off")

plt.show()

# Pré-processamento dos dados

# Agrupamento por marca e tipo para contagem, usando 'id' como referência
```

```
plot1 = train.groupby(by=["information","type"]).count().reset_index()

plot1.head()


# Mostra distribuição desigual de sentimentos entre marcas:

# MaddenNFL e NBA2K têm predominância de tweets negativos

# Outras marcas apresentam distribuição mais balanceada

# Neutral é geralmente a categoria mais frequente


plt.figure(figsize=(20,6))

sns.barplot(data=plot1,x="information",y="id",hue="type")

plt.xticks(rotation=90)

plt.xlabel("Marca")

plt.ylabel("Número de Tweets")

plt.grid()

plt.title("Distribuição de Tweets por Marca e Tipo");


# Transforma cada tweet em uma lista de palavras individuais (tokens)

tokens_text = [word_tokenize(str(word)) for word in train_data.lower]


# Achata a lista de listas em uma única lista e calcula elementos únicos
```

```
# O tamanho do vocabulário (30,436 tokens) indica alta dimensionalidade, o que pode impactar a performance do modelo
tokens_counter = [item for sublist in tokens_text for item in sublist]
print("Número de tokens únicos: ", len(set(tokens_counter)))
# Demonstra como o texto foi dividido em unidades linguísticas básicas
tokens_text[1]

# N-grams de 1 a 4 palavras (captura frases e contextos)
# Sem remoção de stopwords (pode preservar informações contextuais)
bow_counts = CountVectorizer(
tokenizer=word_tokenize,
ngram_range=(1,4)  # Captura uni-, bi-, tri- e four-grams
)

# Split estratificado (80% treino, 20% teste)
# random_state=0 garante reprodutibilidade
reviews_train, reviews_test = train_test_split(train_data, test_size=0.2,
random_state=0)

# Aprende o vocabulário apenas com dados de treino
# Aplica a mesma transformação nos dados de teste
```

```
X_train_bow = bow_counts.fit_transform(reviews_train.lower)  # Treino + vocabulário

X_test_bow = bow_counts.transform(reviews_test.lower)

# Formato (n_tweets, n_palavras_únicas) com contagem de ocorrências

X_test_bow

y_train_bow = reviews_train['type']

y_test_bow = reviews_test['type']


#Mostra desbalanceamento moderado com predominância de tweets Negativos e Positivos

y_test_bow.value_counts() / y_test_bow.shape[0]


# C=0.9: Regularização ligeiramente maior

# max_iter=1500: Garantir convergência

model = LogisticRegression(C=0.9, solver="liblinear", max_iter=1500)

model.fit(X_train_bow, y_train_bow)

test_pred = model.predict(X_test_bow)

print("Acurácia: ", accuracy_score(y_test_bow, test_pred) * 100)

y_val_bow = val_data['type']

X_val_bow = bow_counts.transform(val_data["lower"])

Val_pred = model.predict(X_val_bow)

print("Acurácia: ", accuracy_score(y_val_bow, Val_pred) * 100)
```

## 9. CÁLCULO DE ACURÁCIA

Para avaliar o desempenho do modelo de classificação de sentimentos em tweets, será utilizada a métrica de acurácia. A acurácia é definida como a proporção de previsões corretas feitas pelo modelo em relação ao total de amostras analisadas. Sua fórmula é dada por:

Nosso conjunto apresenta categorias (positivo, negativo, neutro, irrelevante) com contagens similares, então a acurácia fornece uma visão geral confiável do desempenho.

Aqui queremos maximizar o número total de classificações corretas antes de aprofundar em erros específicos. Como nosso foco inicial é avaliar a taxa global de acertos em múltiplas classes balanceadas, a acurácia foi definida como métrica principal.

Acurácia = Número de previsões corretas / Número total de previsões

Para calcular essa métrica, será utilizada a função *accuracy_score* da biblioteca Scikit-Learn, que compara os rótulos reais dos sentimentos com as previsões geradas pelo modelo. O cálculo será realizado conforme o código abaixo:

```
from sklearn.metrics import accuracy_score

# Rótulos reais (valores esperados)

y_true = ["positive", "neutral", "negative", "positive", "negative", "neutral", "irrelevant"]

# Previsões do modelo

y_pred = ["positive", "neutral", "negative", "neutral", "negative", "positive", "irrelevant"]

# Cálculo da acurácia

acuracia = accuracy_score(y_true, y_pred)

# Exibir o resultado formatado

print(f"Acurácia do modelo: {acuracia:.4f}")
```

## 10. ANÁLISE EXPLORATÓRIA DE DADOS

Segundo a análise exploratória dos dados detalhada no github, as palavras mais comuns em tweets positivos incluem termos de apreço e empolgação, enquanto tweets negativos frequentemente mencionam marcas específicas, sugerindo que são reclamações diretas. Além disso, a categoria "Irrelevant" apresenta certa semelhança com a negativa, o que pode impactar a precisão da classificação.

A modelagem com Regressão Logística demonstrou um desempenho satisfatório, com uma acurácia de 90,79% nos dados de treinamento e 98,6% na validação. No entanto, a distribuição das classes mostrou um leve desbalanceamento, o que pode influenciar a performance do modelo. A implementação de outras técnicas, como TF-IDF ou redes neurais, poderia ser explorada para aprimorar os resultados.

Durante o desenvolvimento do projeto, a equipe se deparou com alguns desafios técnicos e metodológicos significativos. Notavelmente, a considerável semelhança semântica entre as classes de sentimento "irrelevante" e "negativa" dificultou a distinção precisa entre opiniões genuinamente negativas e conteúdos que, embora pudessem usar linguagem similar, eram na verdade fora de contexto. Adicionalmente, observou-se um leve desbalanceamento na distribuição das classes de sentimento nos dados, o que, apesar de moderado, apresentava o risco de enviesar a métrica de acurácia, potencialmente favorecendo o desempenho do modelo nas categorias com maior número de exemplos. Outro obstáculo foi a alta dimensionalidade do vocabulário, com mais de 30 mil tokens únicos identificados, impondo desafios consideráveis tanto para a performance computacional quanto para o tempo necessário para o treinamento do modelo. Por fim, embora o modelo de regressão logística tenha apresentado um bom desempenho geral, foram identificadas limitações em sua capacidade de capturar nuances emocionais mais complexas, como o sarcasmo e a ironia, que exigem uma compreensão contextual mais profunda.

## 11. POSSÍVEL PRODUTO

Com base nessa análise, poderia ser desenvolvida uma ferramenta de monitoramento de sentimentos em tempo real para o Twitter. O produto permite que empresas visualizem a percepção da sua marca/produto por meio de um dashboard interativo, que exibe gráficos de polaridade (positivo, negativo, neutro), tópicos mais mencionados e tendências ao longo do tempo. Além disso, a ferramenta gera alertas automáticos quando há picos de menções negativas, permitindo uma resposta rápida a possíveis crises de imagem.

Os planos seriam estruturados em tiers mensais, com valores escalonados conforme o volume de tweets analisados. Por exemplo, um plano básico poderia oferecer a análise de até 10 mil tweets mensais por R$ 500, enquanto planos mais robustos, com capacidades superiores de processamento e recursos adicionais, teriam preços progressivamente mais altos.

## 12. CONSIDERAÇÕES FINAIS

O projeto teve início com a definição clara do objetivo: desenvolver um modelo de análise de sentimentos para classificar tweets com base nas emoções expressas. A equipe organizou um cronograma de execução dividido por tarefas e integrantes, o que permitiu uma estruturação eficiente das etapas.

O dataset utilizado provinha de uma base pública com mais de 3 milhões de tweets classificados por sentimento e marca. Após a coleta, a equipe realizou um extenso pré-processamento textual, utilizando bibliotecas como Pandas, Numpy e Regex, padronizando os dados e aplicando técnicas de vetorização BoW com n-grams (1 a 4 palavras). A limpeza rigorosa dos textos incluiu a remoção de links, símbolos e palavras irrelevantes para garantir maior qualidade na entrada do modelo.

O projeto cumpriu seu objetivo principal, demonstrando que, com técnicas clássicas de PLN e machine learning, é possível construir um modelo eficiente de análise de sentimentos em tempo real. A validação com altíssima acurácia mostra o potencial de aplicação prática, especialmente em ambientes corporativos.

A equipe propôs um produto escalável, um dashboard de monitoramento de sentimentos em redes sociais, com planos pagos, alertas de crises e análise de tendências. Os próximos passos incluem a adoção de métodos mais avançados (TF-IDF, redes neurais), ampliação para outras plataformas (ex: Instagram, TikTok) e melhorias na detecção de nuances emocionais.

## REFERÊNCIAS

KAGGLE. *Twitter Entity Sentiment Analysis*. Disponível em: https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis. Acesso em: 27 fevereiro 2025.