

CURSO: Ciência de Dados

Disciplina: Engenharia de Dados


ANÁLISE DE SENTIMENTOS: TWITTER

04/12/2023

TURMA: Grupo 10


MEMBROS:

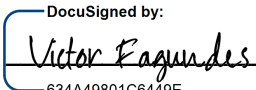
Alison de Almeida Sales:(Ass) 
DocuSigned by:
284A8FCA3187487...

Beatriz Monteiro:(Ass) 

Giovanna Paola Lunetta:(Ass) 

Leonardo Moreno:(Ass) 

Mashara Arambasic:(Ass) 
DocuSigned by:
C54F3D19A71B41E...

Victor Fagundes:(Ass) 
DocuSigned by:
634A49801C6449E...

Yago Angelini Candido:(Ass) _____

Sumário

Introdução.....	3
Resumo	3
Abstract	3
Objetivos.....	3
Descrição geral do sistema	3
Revisão do Tema.....	Error! Bookmark not defined.
Arquitetura	4
Processos	4
Visão Geral	4
Fontes dos dados	4
Base "train.csv"	4
Base "Twitter_Data.csv"	5
Base "twitter_training.csv"	6
Técnicas de Processamento Aplicadas	6
Consolidação de arquivos	6
Padronização de classes	7
Conversão de Tipo	7
Transformação de Caixa de Texto.....	7
Pontuação	7
Stopwords.....	8
Vetorização	8
TF-IDF	9
Treino/Teste	9
Pipeline	9
Análise.....	10
Sistemas de análise	10
Python	10
SciKit Learn	10
Jupyter Notebooks.....	10
Resultados.....	11
Conclusões.....	12
Referências	12
Bases de dados.....	12
GitHub	12

1. Introdução

Resumo

O projeto de Análise de Sentimentos de Textos no Twitter se origina da necessidade de compreender as opiniões e emoções expressas em um dos maiores repositórios de informações em tempo real disponíveis - o Twitter. Identificar sentimentos em tweets pode ser vital para diversas aplicações, desde monitorar a satisfação do cliente até entender tendências de mercado e reações a eventos em tempo real.

Abstract

The Text Sentiment Analysis project on Twitter originates from the need to understand the opinions and emotions expressed in one of the largest real-time information repositories available - Twitter. Identifying sentiments in tweets can be crucial for various applications, from monitoring customer satisfaction to understanding market trends and reactions to real-time events.

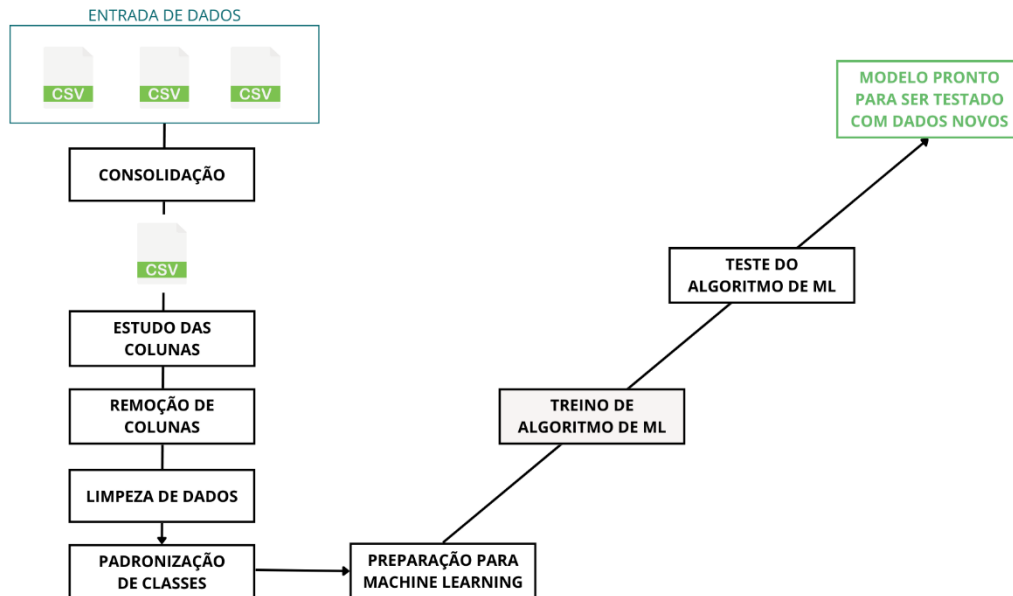
2. Objetivos

O objetivo principal deste projeto é desenvolver um sistema de análise de sentimentos que seja capaz de classificar textos do Twitter em duas categorias principais: negativo ou positivo. Isso permitirá uma avaliação abrangente do sentimento geral associado a tópicos específicos, marcas, eventos ou tendências no Twitter.

3. Descrição geral do sistema

Faremos a coleta de um grande volume de tweets do Twitter, abrangendo uma ampla gama de tópicos e fontes, então realizaremos tarefas de limpeza, transformação e remoção de ruídos para preparar os dados para análise. Desenvolveremos e treinaremos um modelo de machine learning, para classificar os tweets em categorias de sentimento. Avaliaremos a precisão do modelo usando métricas de desempenho, como precisão, recall e F1-score.

4. Arquitetura



5. Processos

Visão Geral

Iniciamos coletando tweets do Twitter e selecionando informações-chave, como ID do tweet, texto e sentimentos (positivo, negativo, neutro ou irrelevante). Em seguida, realizamos uma limpeza detalhada dos dados, incluindo tratamento de nulos e normalização do texto. Na fase de modelagem, dividimos os dados, escolhemos e treinamos um modelo de machine learning para prever sentimentos. A avaliação do modelo se deu através de métricas como precisão e recall. Finalmente, aplicamos o modelo a dados não vistos, interpretando e comunicando os resultados, destacando padrões de sentimentos na base de tweets. Este processo proporciona uma compreensão abrangente dos sentimentos presentes nos dados do Twitter, permitindo insights valiosos.

Fontes dos dados

Base "train.csv"

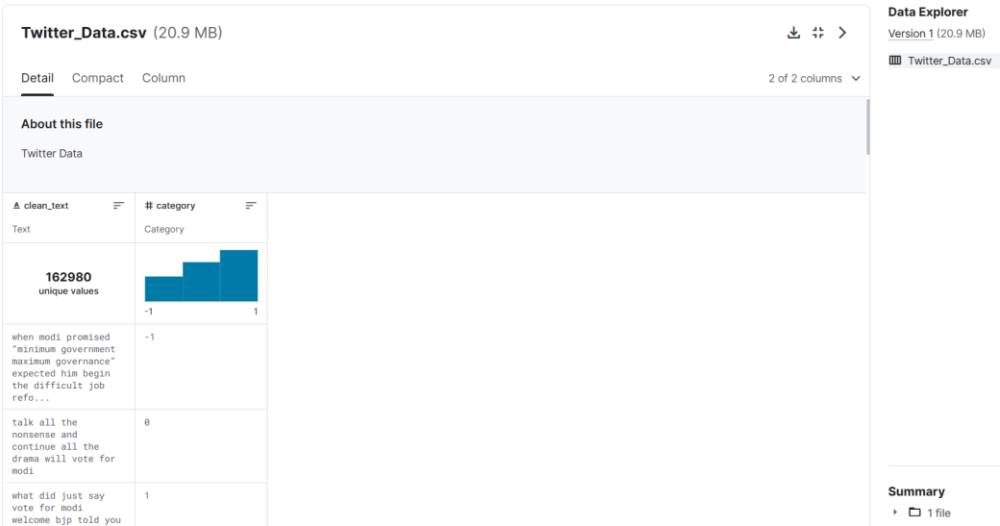
- Arquivo de valores separados por vírgulas que nos apresenta alguns tweets;
- A base tem as seguintes informações:
 - textID (ID do texto; texto);
 - text (o Tweet; texto);

- selected_text (o texto selecionado que está sendo levado em consideração; texto);
- sentiment (o sentimento que o texto está passando, neutro, negativo ou positivo; texto);
- Time of Tweet (em que horário do dia o tweet foi publicado, manhã, tarde, noite; texto);
- Age of User (idade do usuário que publicou o tweet; intervalo numérico);
- Country (País de origem do tweet; texto);
- Population - 2020 (a população do país em 2020; número);
- Land Area (Km²) (área do país; número);
- Density (P/Km²) (Densidade do país, população dividida pela área; número).

train.csv (4.64 MB)										Data Explorer	
Detail Compact Column										Version 9 (150.82 MB)	
10 of 10 columns										test.csv testdata.manual.2009.06.14.csv train.csv training.1600000.processed.noemoticon.csv	
textID	text	selected_text	sentiment	Time of Tweet	Age of User	Country					
27481 unique values	27481 unique values	22431 unique values	neutral 40% positive 31% Other (7781) 28%	morning 33% noon 33% Other (9160) 33%	0-20 17% 21-30 17% Other (18320) 67%	19 unique					
cb774db8d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral	morning	0-20	Afghanistan					
549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative	noon	21-30	Albania					
888c6ef138	my boss is bullying me...	bullying me	negative	night	31-45	Algeria					
9642c863ef	what interview! leave me alone	leave me alone	negative	morning	46-60	Andorra					
358bd9e861	Sons of ****, why couldn't they put them on the releases we already bought	Sons of ****,	negative	noon	60-70	Angola					
28b57f3998	http://www.dotheboun cy.com/saf - some shameless plugging for the best Rangers forum on earth	http://www.dotheboun cy.com/saf - some shameless plugging for the best Rangers forum on earth	neutral	night	70-100	Antigua and					
6e8c6d75b1	2am feedings for the hubb are fun when ha	fun	positive	morning	0-20	Argentina					
										Summary	
										4 files	

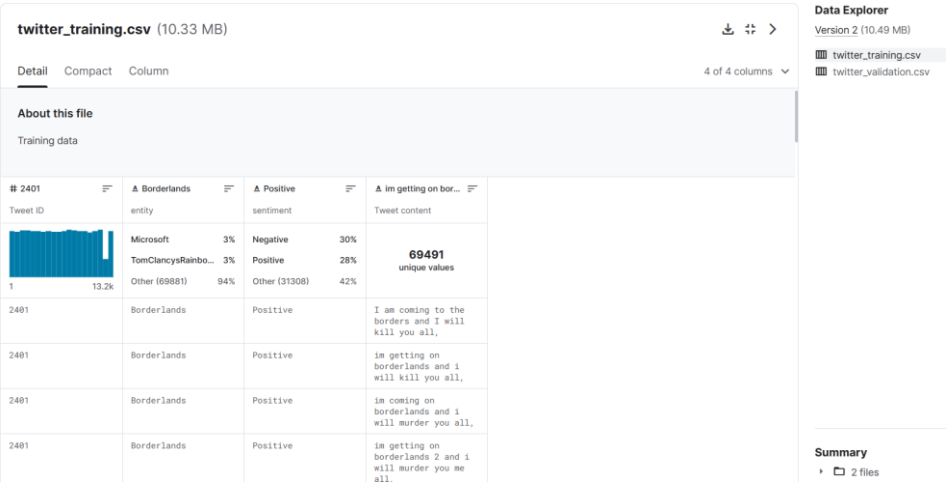
Base "Twitter_Data.csv"

- Arquivo de valores separados por vírgulas que nos apresenta alguns tweets;
- A base tem as seguintes informações:
 - clean_text (o tweet inteiro; texto);
 - category (sentimento do texto, se é positivo, negativo ou neutro, sendo 1 positivo, 0 neutro e -1 negativo; número).



Base "twitter_training.csv"

- Arquivo de valores separados por vírgulas que nos apresenta alguns tweets que são as opiniões de usuários sobre alguns jogos.
- A base tem as seguintes informações:
 - TweetId (código de 4 números; texto);
 - Entity (nome da entidade em foco no tweet; texto);
 - Sentiment (sentimento do tweet, se é positivo, negativo, neutro ou irrelevante; texto);
 - TweetContent (tweet na íntegra; texto).



Técnicas de Processamento Aplicadas

Consolidação de arquivos

Usando Python e Pandas, consolidamos vários arquivos Excel, ou seja, diferentes bases de dados em uma única base de dados. Usando uma espécie de loop para percorrer os arquivos no diretório, lendo cada arquivo

com `pd.read_excel`, e concatenando os DataFrames usando `pd.concat`. Por fim, ajustamos os formatos conforme necessário.

Padronização de classes

Na padronização de classes, foi fundamental uniformizar as categorias em uma variável de destino. Utilizando Python e Pandas, esse processo envolveu a identificação das classes existentes, opcionalmente o mapeamento para rótulos padronizados, e a aplicação da padronização usando a função `replace`. Este método simplifica e organiza as classes, facilitando análises subsequentes e garantindo consistência nos dados. A verificação final e ajustes refinados podem ser realizados conforme necessário para obter um resultado preciso e coerente.

Conversão de Tipo

Foi feita a alteração dos formatos de dados de uma variável para adequá-la às necessidades analíticas. Efetuamos esse processo por meio da função `astype`, permitindo a transformação de tipos de dados, como de texto para numérico. Essa conversão foi crucial para garantir a consistência e a precisão dos dados, facilitando operações matemáticas e análises estatísticas. No entanto, foi importante realizar verificações cuidadosas para evitar perda de informação ou erros durante a conversão, adaptamos de acordo com as exigências específicas do conjunto de dados que possuímos

Transformação de Caixa de Texto

Envolveu a padronização da capitalização das palavras em uma variável de texto. Realizamos essa operação usando métodos como `str.lower()` para converter todo o texto em minúsculas. Essa abordagem foi útil para garantir consistência nos dados, facilitando comparações e análises, independentemente da capitalização original. Ao aplicar a transformação de caixa de texto, foi essencial considerar a natureza específica do conjunto de dados, ajustando conforme necessário para manter a integridade das informações e otimizar a eficácia das operações subsequentes.

Pontuação

A remoção de pontuação foi essencial no pré-processamento. Realizamos a remoção através do método `str.replace()` combinando com expressões regulares para substituir caracteres de pontuação por espaços ou removê-los completamente. Essa operação foi valiosa para garantir consistência e melhorar a eficácia das análises textuais, removendo esses elementos não essenciais. No entanto, ao aplicar essa remoção, foi importante avaliar cuidadosamente o impacto sobre os dados, ajustando o método conforme necessário para preservar o significado do texto original.

Stopwords

A manipulação feita válida de citação foi a remoção de Stopwords. As stopwords são palavras comuns que geralmente são removidas ao processar textos, pois não contribuem significativamente para o significado em uma análise textual.

Exemplos de stopwords incluem "e", "de", "para", "o", "a", entre outras. A remoção de stopwords é comumente usada ao processar texto para análise de sentimentos, classificação de documentos, entre outros.

Para realizar a remoção de stopwords:

- Precisa ter uma lista de palavras que deseja remover, geralmente obtida de bibliotecas de processamento de linguagem natural.
- Em seguida, você divide o texto em palavras individuais.
- Para cada palavra, você verifica se ela está na lista de stopwords.
- Se a palavra for uma stopwords, ela é removida do texto; caso contrário, é mantida.

Essa técnica ajuda a reduzir o ruído e a dimensionalidade do texto, focando nas palavras mais significativas para a análise.

Remoção de Tweets Nulos, Irrelevantes e Duplicados

Para realizar a limpeza da base de dados de tweets em Python utilizando Pandas, começamos removendo tweets nulos com `dropna()`. Em seguida, eliminamos tweets irrelevantes, exemplificado aqui por aqueles com menos de 5 caracteres. Para finalizar, aplicamos `drop_duplicates()` para remover tweets duplicados com base no texto do tweet. Esses passos simplificados asseguram que a base de dados estejam livre de valores nulos, tweets irrelevantes e duplicatas, proporcionando um conjunto mais coeso e apropriado para análises futuras.

Normalização de Classes

Para normalizar as classes, inicialmente, foi calculada a contagem de amostras por classe. Em seguida, determinamos a quantidade desejada de amostras por classe, usamos a quantia igual à menor contagem. A normalização é realizada utilizando `groupby()` e `sample()`, mantendo apenas o número desejado de amostras por classe. O resultado é um DataFrame normalizado, equilibrando a distribuição de classes na base de dados. Essa abordagem é valiosa quando há desequilíbrio nas classes, buscando assegurar uma representação mais uniforme para análises subsequentes.

Vetorização

A vetorização envolveu a transformação dos textos em representações numéricas para alimentar modelos de machine learning. Utilizando a classe `CountVectorizer` ou `TfidfVectorizer` do Scikit-Learn, é possível converter os textos em vetores de contagens ou termos ponderados, respectivamente. O processo inclui a tokenização, remoção de stop words e atribuição de valores numéricos a cada palavra. Esses vetores numéricos resultantes podem então ser usados como entrada para modelos de aprendizado de máquina, permitindo a análise e previsão de sentimentos com base nos tweets. A vetorização foi uma etapa crucial na preparação de dados textuais para análises computacional ser eficaz.

TF-IDF

Para aplicar a transformação TF-IDF (Term Frequency-Inverse Document Frequency) utilizamos a classe `TfidfVectorizer`. Este processo envolve a tokenização e contagem da frequência de termos nos documentos (tweets) e, em seguida, a aplicação do esquema de ponderação TF-IDF, que leva em consideração a frequência do termo em um tweet específico e a raridade do termo em toda a base de dados. O resultado é uma matriz numérica em que cada coluna representa um termo ponderado pelo seu impacto relativo. Essa representação foi valiosa para análises de texto e alimentação de modelos de machine learning, uma vez que captura a importância dos termos em relação ao contexto global da base de dados.

Treino/Teste

A divisão entre conjuntos de treino e teste é essencial para avaliar a performance de modelos de machine learning. Após o pré-processamento dos dados, a função `train_test_split` foi empregada para separar a base em dois conjuntos independentes: o conjunto de treino, usado para treinar o modelo, e o conjunto de teste, utilizado para avaliar sua eficácia em dados não vistos anteriormente. Essa divisão permitiu uma avaliação mais realista do desempenho do modelo, verificando sua capacidade de generalização para além dos dados de treinamento. Usamos a proporção de 80% para treino e 20% para teste.

Pipeline

Organizamos e automatizamos o fluxo de trabalho. A classe `Pipeline` permitiu encadear diversas etapas, como pré-processamento, extração de características e treinamento de modelos, em uma única estrutura. Cada etapa da pipeline é definida com um par de nome e estimador, facilitando a execução sequencial de tarefas. Isso não apenas simplificou o código,

tornando-o mais modular e legível, mas também possibilita a busca por hiperparâmetros de maneira integrada. Uma vez construída, a pipeline pode ser ajustada ao conjunto de treinamento e utilizada para previsões no conjunto de teste, proporcionando um processo mais eficiente e organizado na implementação de modelos de machine learning.

6. Análise

Sistemas de análise

Para a análise de sentimentos em tweets, o código fornecido utiliza principalmente a linguagem de programação Python e diversas bibliotecas especializadas em processamento de texto, análise de dados e machine learning. Vamos destacar as principais ferramentas e sistemas utilizados:

Python

A linguagem de programação Python é a espinha dorsal do projeto. Ela é amplamente utilizada para desenvolvimento de aplicações de análise de dados e machine learning devido à sua sintaxe clara, vasta comunidade de desenvolvedores e uma variedade de bibliotecas especializadas.

SciKit Learn

O código faz extenso uso da biblioteca SciKit Learn para implementar tarefas relacionadas à análise de sentimentos. Isso inclui a criação de pipelines para o pré-processamento de texto, vetorização, treinamento de modelos e avaliação de desempenho.

Jupyter Notebooks

Jupyter Notebooks são usados para a execução do código de forma interativa, permitindo a visualização imediata dos resultados e facilitando a análise exploratória de dados.

Algoritmo de ML: Naive Bayes

A implementação de um algoritmo de Machine Learning Naive Bayes em Python, foi utilizado através da biblioteca Scikit-Learn, envolvendo algumas etapas chave. Primeiramente, foi necessário vetorizar os textos dos tweets utilizando TF-IDF. Em seguida, a base de dados foi dividida em conjuntos de treino e teste. O modelo Naive Bayes foi então treinado no conjunto de treino, onde as probabilidades condicionais de ocorrência de termos são estimadas

assumindo independência condicional entre eles. Posteriormente, o modelo é avaliado no conjunto de teste, e métricas como precisão, recall e F1-score foram utilizadas para avaliar seu desempenho na previsão dos sentimentos dos tweets. A simplicidade e eficiência do Naive Bayes o tornam uma escolha comum para tarefas de classificação de texto, incluindo a análise de sentimentos em tweets.

Resultados

Classification Report

```
from sklearn.metrics import classification_report
# Substitua y_true pelos rótulos verdadeiros e y_pred pelas previsões do seu modelo
y_true = [...] # Seus rótulos verdadeiros y_pred = [...] # As previsões do seu modelo
# Crie o relatório de classificação class_report = classification_report(y_true, y_pred)
# Imprima o relatório de classificação print("Relatório de Classificação:")
print(class_report)
```

Matriz de Confusão

Verdadeiro Positivo (VP): Classificações corretas como positivas.

Falso Positivo (FP): Classificações incorretas como positivas.

Falso Negativo (FN): Classificações incorretas como negativas.

Verdadeiro Negativo (VN): Classificações corretas como negativas.

O código fornecido realiza uma análise abrangente de dados de sentimentos em tweets, utilizando um modelo de classificação Naive Bayes Multinomial com o SciKit Learn em Python. A etapa de pré-processamento inclui a normalização do texto, remoção de pontuações e stopwords, além da tokenização dos tweets. A distribuição do comprimento dos tweets revela uma média de aproximadamente 107 caracteres, com uma variação significativa entre mensagens mais curtas e mais longas. A análise de sentimentos por meio do modelo treinado demonstra uma precisão global de cerca de 61%, sendo que as métricas detalhadas, como precisão, recall e F1-score, variam para cada classe de sentimento.

Ao explorar a distribuição de sentimentos, observa-se uma relativa equidade entre as classes, indicando que o modelo não possui grandes desequilíbrios na capacidade de prever diferentes tipos de sentimentos. O código também destaca mensagens extremamente curtas, algumas com apenas um caractere, sugerindo a presença de dados atípicos que podem ser examinados mais detalhadamente. Como sugestões de melhorias, o código ressalta a possibilidade de ajustes nos parâmetros do modelo, a exploração de diferentes algoritmos de aprendizado de máquina, e a realização de uma Análise Exploratória de Dados (EDA) mais aprofundada para obter insights adicionais sobre o conjunto de dados.

Em termos práticos, os resultados indicam que o modelo treinado é capaz de realizar uma classificação razoável dos sentimentos presentes nos tweets, com

margem para aprimoramentos. Considerando o contexto do projeto, ajustes adicionais no pré-processamento de texto e na seleção do modelo podem contribuir para melhorar ainda mais a eficácia da classificação de sentimentos em dados do Twitter.

7. Conclusões

A detecção de sentimentos negativos em tweets possibilita a identificação precoce de sinais de saúde mental, facilitando intervenções e apoio apropriado. A análise do sentimento em torno de questões sociais e políticas oferece insights preciosos sobre o engajamento cívico e as perspectivas da juventude em relação a temas relevantes. Monitorar sentimentos em tweets é fundamental para compreender as dinâmicas culturais emergentes, ampliando a voz da juventude e oferecendo insights para o desenvolvimento de políticas e estratégias alinhadas com suas perspectivas. Identificar sentimentos ligados a questões sociais cruciais, como igualdade, diversidade e inclusão, desempenha um papel fundamental em impulsionar campanhas de conscientização e promover mudanças positivas na sociedade.

8. Referências

Bases de dados

- Twitter_Data.csv: [Twitter Sentiment Dataset \(kaggle.com\)](#)
- Train.csv: [Sentiment Analysis Dataset \(kaggle.com\)](#)
- Twitter_training.csv: [Twitter Sentiment Analysis \(kaggle.com\)](#)

GitHub

- https://github.com/leomoreno11/trabalho_engenhariaDeDados/tree/main