Projeto de disciplina

Análise de sentimentos no Twitter

Docente: Fabio Silveira Vidal
Discentes: Jefté Lopes, Edivaldo Araujo, Lucas Lopes
Disciplina: Inteligência Artificial
Período: 6º

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
TOCANTINS
Campus Paraíso do Tocantins

Problema

Classificar os sentimentos do Twitter como Negativo ou positivo

O objetivo desse estudo de caso é criar um modelo que analisa um ou mais Tweets para prever o sentimento(Positivo ou Negativo) presente em cada Tweet.

Possíveis trabalhos futuros

A Inteligência artificial e a analise de sentimentos baseadas em aprendizado de maquina é crucial para empresas, visto que, os insight revelado pela analise visa indicar o grau de qualidade dos serviços e/ou produtos da empresa de acordo com os clientes.

Esse projeto é diretamente aplicável a praticamente qualquer empresa que disponha de meios online(Twitter, Instagram, Facebook, Website) para interagir com seus clientes.

Os algoritmos podem ser usados para detectar e possivelmente sinalizar automaticamente tweets de ódio e racismo.

1º Solução

- ❑ Base de dados
  - ✓ Idioma inglês
  - ✓ Utilizando tweets
  - ✓ Baixada no kaggle
- ❑ NLP(Natural Language Processing)
- ❑ Bag of Word
- ❑ Naive Bayes
- ❑ Visualização dos Resultado

2º Solução

- ❑ Base de dados
  - ✓ Idioma português-Brasil
  - ✓ Utilizando tweets
  - ✓ Criada
  - ✓ NLP(Natural Language Processing)
- ❑ Bag of Word
- ❑ Naive Bayes
- ❑ Visualização dos Resultado

1º Solução

- ❑ Entender a Declaração do Problema e o caso de negócios.
- ❑ Importar bibliotecas e conjuntos de dados.
- ❑ Executar a análise exploratória dos dados.
- ❑ Plotar a nuvem de palavras.
- ❑ Executar a limpeza de dados - remover pontuação.
- ❑ Executar a limpeza de dados - remover palavras de parada(stop words).
- ❑ Executar vetorização de contagem (Tokenization).
- ❑ Criar um pipeline para remover palavras irrelevantes, pontuação e realizar tokenização.
- ❑ Treinar um Classificador Naive Bayes.
- ❑ Avaliar o desempenho do modelo treinado.

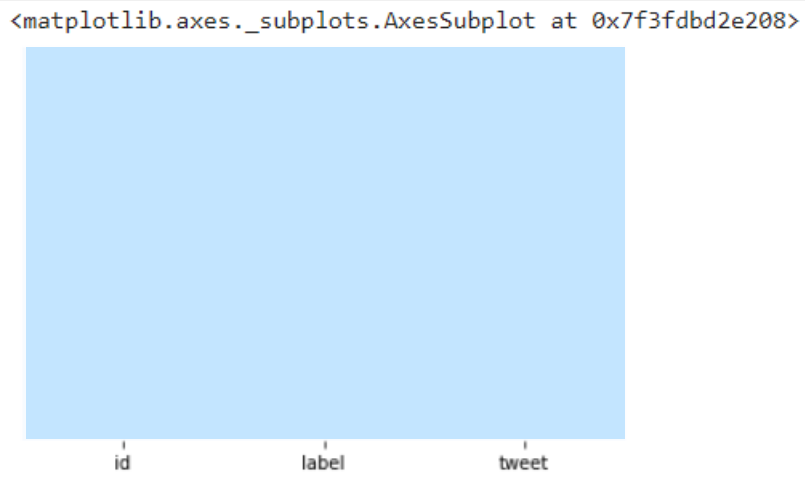
2º Solução

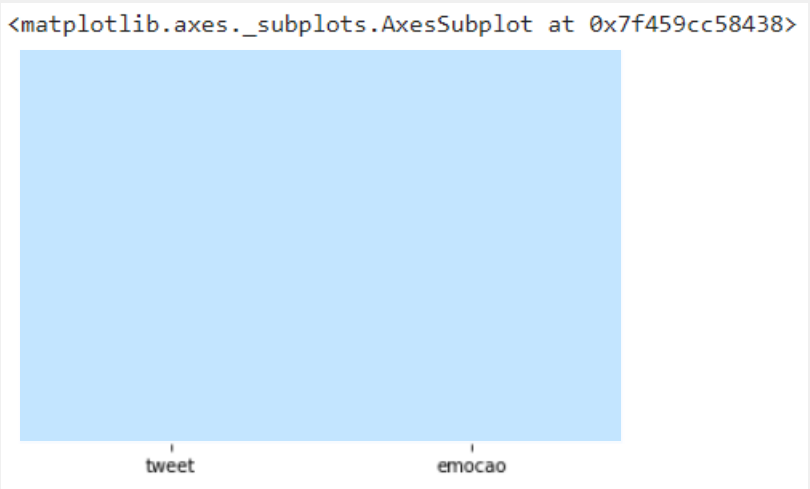
- ❑ Entender a Declaração do Problema e o caso de negócios.
- ❑ Importar bibliotecas e conjuntos dos dados.
- ❑ Executar a análise exploratória de dados.
- ❑ Plotar a nuvem de palavras.
- ❑ Executar vetorização de contagem (Tokenization).
- ❑ Treinar um Classificador Naive Bayes.
- ❑ Avaliar o desempenho do modelo treinado.
- ❑ Salvar o modelo treinado.

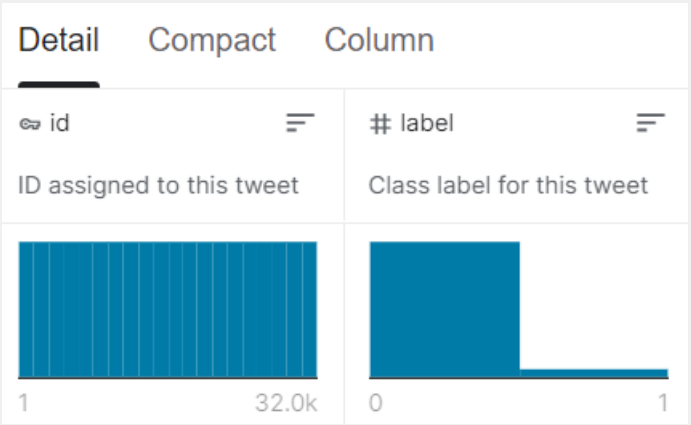
1º Solução

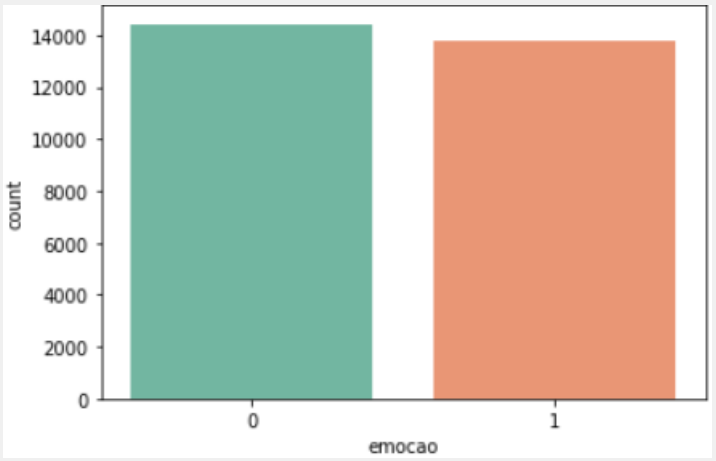
2º Solução

Análise exploratória dos dados









1º Solução

2º Solução

Base de Dados

BASE DO KAGGLE

- Base de dados criada a partir de uma busca de tweets racistas e sexistas.
- Sem acesso a como foi criada e quais termos utilizados para sua criação.
- Alto desbalanceamento dos dados positivos e negativos.
- Necessidade de limpeza dos dados.

BASE CRIADA UTILIZANDO A API TWEEPY

- Criada a partir de buscas por termos que referenciam em sua grande parte um sentimento negativo ou positivo.
- Base balanceada entre a quantidade de tweets positivos e negativos.
- Enquanto a base é criada também é feita a limpeza dos dados, excluindo dados redundantes, eliminação de pontuações, links e usuários.

:) feliz	:(triste
:) alegre	:-(triste
\o/ empolgado	:(chateado
:) amor	:(mau
\o/confiante	:(morrer
:) apaixonado	
\o/ otimista	
resiliência	

1º Solução

2º Solução

Processamento de Linguagem Natural

- Na primeira solução foi utilizado o processamento de linguagem natural após o download da base de dados no kaggle.
- Remoção de:
  - Pontuação
  - Números
  - Stop Words(NLTK)
  - Entre outros caracteres indesejados

- Na segunda solução foi utilizado o processamento de linguagem natural durante a criação da base de dados.
- Remoção de:
  - Pontuação
  - Números
  - Links
  - Usuários
  - Entre outros caracteres indesejados

1º Solução

2º Solução

Bag of Words(sklearn)

```
"""
[
  'Porcaria de produto',
  'Obrigado pelo retorno, obrigado mesmo OBRIGADO OBRIGADO oBriGaDO',
  'Otimo atendimento'
]

['atendimento', 'de', 'mesmo', 'obrigado', 'otimo', 'pelo', 'porcaria', 'produto', 'retorno']

{'porcaria': 6, 'de': 1, 'produto': 7, 'obrigado': 3, 'pelo': 5, 'retorno': 8, 'mesmo': 2, 'otimo': 4, 'atendimento': 0}

[[0 1 0 0 0 0 1 1 0]
 [0 0 1 5 0 1 0 0 1]
 [1 0 0 0 1 0 0 0 0]]
"""
```

1º Solução

2º Solução

Naive Bayes

Para criar o modelo de classificação foi utilizado o algoritmo de Naïve Bayes.

```
# Separação dos dados de treino e teste
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

```
[ ]  # Utilização do algoritmo de Naive Bayes para o treinamento
     from sklearn.naive_bayes import MultinomialNB

     NB_classifier = MultinomialNB()
     NB_classifier.fit(X_train, y_train)
```

1º Solução

2º Solução

Resultados

```
[ ] y_predict_test = NB_classifier.predict(X_test)
    cm = confusion_matrix(y_test, y_predict_test)
    cm
    # sns.heatmap(cm, annot=True)
```

```
array([[5773,  185],
       [ 198,  237]])
```

```
[ ] print(classification_report(y_test, y_predict_test))
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	5958
1	0.56	0.54	0.55	435
accuracy			0.94	6393
macro avg	0.76	0.76	0.76	6393
weighted avg	0.94	0.94	0.94	6393

```
[39] y_predict_test = NB_classifier.predict(X_test)
     cm = confusion_matrix(y_test, y_predict_test)
     cm
     # sns.heatmap(cm, annot=True)
```

```
array([[2790,   76],
       [ 170, 2606]])
```

```
[33] print(classification_report(y_test, y_predict_test))
```

	precision	recall	f1-score	support
0	0.94	0.98	0.96	2881
1	0.98	0.94	0.96	2761
accuracy			0.96	5642
macro avg	0.96	0.96	0.96	5642
weighted avg	0.96	0.96	0.96	5642

Google Colaboratory – (Notebooks)

GitHub

- Os notebooks utilizados juntamente com as bases de dados e o modelo de classificação estão disponibilizados no repositório do GitHub abaixo.

  - ✓ https://github.com/JefteLG/Twitter_Sentiment_Analysis