

TwitterScraper.ipynb

File Edit View Insert Runtime Tools Help Last saved at 05:00

Comment Share Settings

Connect Editing

Code Text

⏮ ⏪ ⏩ ⏭ 🔍 ⚙️ 📄 📌 📋 📧

```
!pip install tweepy
import tweepy
import datetime
import pytz

!pip install stop_words --use-pep517
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.util import ngrams
from stop_words import get_stop_words
import re

!pip install vaderSentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from nltk import bigrams

Requirement already satisfied: tweepy in c:\python311\lib\site-packages (4.12.1)
Requirement already satisfied: oauthlib<4,>=3.2.0 in c:\python311\lib\site-packages (from tweepy) (3.2.2)
Requirement already satisfied: requests<3,>=2.27.0 in c:\python311\lib\site-packages (from tweepy) (2.28.1)
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in c:\python311\lib\site-packages (from tweepy) (1.3.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\python311\lib\site-packages (from requests<3,>=2.27.0->tweepy) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\python311\lib\site-packages (from requests<3,>=2.27.0->tweepy) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python311\lib\site-packages (from requests<3,>=2.27.0->tweepy) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in c:\python311\lib\site-packages (from requests<3,>=2.27.0->tweepy) (2022.12.7)
Requirement already satisfied: stop_words in c:\python311\lib\site-packages (2018.7.23)
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\abhit\AppData\Roaming\nltk_data...
[nltk_data] package stopwords is already up-to-date!
Requirement already satisfied: vaderSentiment in c:\python311\lib\site-packages (3.3.2)
Requirement already satisfied: requests in c:\python311\lib\site-packages (from vaderSentiment) (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\python311\lib\site-packages (from requests->vaderSentiment) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\python311\lib\site-packages (from requests->vaderSentiment) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python311\lib\site-packages (from requests->vaderSentiment) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in c:\python311\lib\site-packages (from requests->vaderSentiment) (2022.12.7)
```

For a User Input on stock

```
[ ] # Set up the API with your Twitter developer account's credentials
consumer_key = ""
consumer_secret = ""
access_token = ""
access_token_secret = ""

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

[ ] stock_ticker = str(input('Enter the stock code'))
stock_ticker_full = str(input('Enter the stock name'))
stock_hashtag = '#' + stock_ticker

Enter the stock code ZOMATO
Enter the stock name Zomato

[ ] data = []
usernames = ["vsgj", "Reuters", "business", "CNBC", "WSJmarkets", "liveint",
              "moneycontrolcom", "NDTVProfit", "bsindia", "FinancialTimes",
              "ETMarkets", "bqprime", "Investopedia", "BloombergTV", "FinancialXpress",
              "NSEIndia", "TOIBusiness", "IIFL_Live", "elearnmarkets", "mystockedge"]

for username in usernames:
    tweets = api.user_timeline(screen_name=username, count=100)
    for tweet in tweets:
        if stock_ticker in tweet.text or stock_ticker_full in tweet.text:
            data.append(tweet.text)

tweets = api.search_tweets(q=stock_hashtag, count=100)
for tweet in tweets:
    data.append(tweet.text)

[ ] stop_words = list(stopwords.words('english')) + list(get_stop_words('en'))
pattern = '[0-9]'
cleaned_data = []

def generate_clean_ngrams(tweet_text, n):
    tweet_text = tweet_text.lower()
    tweet_text = re.sub(r'[^a-zA-Z0-9\s]', ' ', tweet_text)
    tweet_text = re.sub(pattern, 'http', tweet_text)

    tokens = [token for token in tweet_text.split(' ')
               if token != ''
               and token not in stop_words
               and 'http' not in token]

    output = list(ngrams(tokens, n))

    return output

for d in data:
    cleaned_data.append(generate_clean_ngrams(d, 2))

[ ] analyzer = SentimentIntensityAnalyzer()
score = []

for data in cleaned_data:
    for d in data:
        bigram_text = " ".join(d)
        scores = analyzer.polarity_scores(bigram_text)
        score.append(scores['compound'])

[ ] print(f'Sentiment Score for the stock {stock_ticker}: {sum(score)/len(score)}')

Sentiment Score for the stock ZOMATO: 0.088237821716649428
```

For NSE top 500

```
[ ] import pandas as pd
import numpy as np

[ ] nselist= pd.read_csv('C:/Users/abhit/Desktop/COLLEGE/SEM X/SEMX_Git/cc/ind_nifty500list.csv')
nselist.head()
```

	Company Name	Symbol
0	3M India Ltd.	3MINDIA
1	ABB India Ltd.	ABB
2	ACC Ltd.	ACC
3	AIA Engineering Ltd.	AIAENG
4	APL Apollo Tubes Ltd.	APLAPOLLO

```
[ ] # Set up the API with your Twitter developer account's credentials
consumer_key = ""
consumer_secret = ""
access_token = ""
access_token_secret = ""

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
```

```
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

```
[ ] nselist['Company Name'] = nselist['Company Name'].str.replace('.', '')
nselist['Company Name'] = nselist['Company Name'].str.replace('Ltd', '')
```

```
C:\Users\Public\Documents\Wondershare\CreatorTemp\ipykernel_26284\571671230.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will "not
nselist['Company Name'] = nselist['Company Name'].str.replace('.', '')
```

```
[ ] data = []
usernames = ["usj", "Reuters", "business", "CNBC", "WSJmarkets", "livemint",
             "moneycontrolcom", "NDTVProfit", "bsindia", "FinancialTimes",
             "ETMarkets", "bqprime", "Investopedia", "BloombergTV", "FinancialXpress",
             "NSEIndia", "TOIIBusiness", "IIFL_Live", "elearnmarkets", "mystockedge"]
```

```
[ ] nselist['Sentiment'] = 0
for index in nselist.index:
    data = []
    stock_ticker = nselist['Symbol'][index]
    stock_ticker_full = nselist['Company Name'][index]
    stock_hashtag = '#' + stock_ticker

    for username in usernames:
        tweets = api.user_timeline(screen_name=username, count=100)
        for tweet in tweets:
            if stock_ticker in tweet.text or stock_ticker_full in tweet.text:
                data.append(tweet.text)

    tweets = api.search_tweets(q=stock_hashtag, count=100)
    for tweet in tweets:
        data.append(tweet.text)

    cleaned_data = []
    def generate_clean_ngrams(tweet_text, n):
        stop_words = list(stopwords.words('english')) + list(get_stop_words('en'))
        pattern = '[0-9]'

        tweet_text = tweet_text.lower()
        tweet_text = re.sub(r'[a-zA-Z0-9\s]', ' ', tweet_text)
        tweet_text = re.sub(pattern, 'http', tweet_text)

        tokens = [token for token in tweet_text.split(' ')
                  if token != ''
                  and token not in stop_words
                  and 'http' not in token]

        output = list(ngrams(tokens, n))
        return output

    for d in data:
        cleaned_data.append(generate_clean_ngrams(d, 2))

    analyzer = SentimentIntensityAnalyzer()

    score = []
    for data in cleaned_data:
        for d in data:
            bigram_text = " ".join(d)
            scores = analyzer.polarity_scores(bigram_text)
            score.append(scores['compound'])

    average_score = sum(score) / len(score)
    nselist['Sentiment'][nselist['Symbol'] == stock_ticker] = average_score
    #nselist['Sentiment'] = np.where(nselist['Symbol'] == stock_ticker, average_score, 0)
    print('Index: ', index, ' ', stock_ticker, ': ', average_score)
```

```
C:\Users\Public\Documents\Wondershare\CreatorTemp\ipykernel_26284\2694785966.py:49: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
nselist['Sentiment'][nselist['Symbol'] == stock_ticker] = average_score
```

```
Index: 0 3MINDIA : 0.019135483870967734
Index: 1 ABB : 0.022285214285714297
Index: 2 ACC : 0.0576765865502183
Index: 3 AIAENG : 0.05084448970918384
Index: 4 APIAPOLLO : 0.04137269147801474
Index: 5 AURANK : 0.0370515690376569
Index: 6 AARTIDRUGS : -0.024600000000000018
Index: 7 AAVAS : 0.03237320261437907
Index: 8 ABBOTINDIA : 0.015310675381263622
Index: 9 ADANIEMI : 0.04225381365986827
Index: 10 ADANIGREEN : 0.033898474059083085
Index: 11 ADANIPORES : 0.02559132149901383
Index: 12 ATGL : 0.027436094674556214
Index: 13 ADANITRANS : 0.008110079575596809
Index: 14 AML : 0.04625295095594346
Index: 15 ABCAPITAL : 0.028575046551724128
Index: 16 ABFRL : 0.03922133971291869
Index: 17 ABSLAWC : 0.03495357142857142
Index: 18 AEGISCHEM : 0.004453086635944701
Index: 19 AETHER : 0.03286536312849162
Index: 20 AFFLE : 0.01797924528301887
Index: 21 AJANTPHARM : -0.03176744180604512
Index: 22 ADLLTD : -0.0008052265081027107
Index: 23 ALKEM : 0.0031595744680851068
Index: 24 ALKYLAMINE : -0.051416279069767444
Index: 25 ALOKINDS : -0.051249626865671624
Index: 26 AMARAJABAT : 0.009024871794871795
Index: 27 AMBER : 0.03085716694723447
Index: 28 AMBUJACHE : -0.0063200000000000003
Index: 29 ANGELONE : 0.04651289782244549
Index: 30 AMURAS : -0.024071337579617845
Index: 31 APOLLOHOSP : 0.028403222222222232
Index: 32 APOLLOTYRE : 0.003254948301329391
Index: 33 APTUS : 0.014155681818181816
Index: 34 ASAHINDIA : 0.032622222222222226
Index: 35 ASHOKLEY : -0.02335208325806455
Index: 36 ASIAMPAIN : 0.010243815331010452
Index: 37 ASTERDM : 0.0014357142857142871
Index: 38 ASTRAZEN : 0.014817599999999995
Index: 39 ASTRAL : 0.0207678060605175
Index: 40 ATUL : 0.001817083379530934
Index: 41 AUROPHARMA : -0.004356900212314237
Index: 42 AVANTIFEED : -0.0469908476190476194
Index: 43 DMART : 0.04136704653371313
Index: 44 AXISBANK : 0.05116705290013234
Index: 45 BASF : -0.00993883089703548
Index: 46 BHEL : 0.0004993749999999999
Index: 47 BSE : 0.0431970401539088
Index: 48 BAJAJ-AUTO : 0.022621012006061077
Index: 49 BAJAJELEC : -0.045533333333333335
Index: 50 BAJAFINANCE : 0.002631155778894473
Index: 51 BAJAJFINSV : 0.014950497017892618
```

```
[ ] nselist['Sentiment Group'] = np.where(nselist['Sentiment'] >= 0.05, 'Positive Sentiment', nselist['Sentiment'] <= -0.05, 'Negative Sentiment', 'Neutral Sentiment')
```

```
[ ] nselist.to_csv('C:/Users/abhit/Desktop/COLLEGE/SEM X/SEM_X_Git/CC/submission.csv', index = False)
```

```
[ ]
```

