SENTIMENT ANALYSIS

Sentiment analysis is a technique used to determine the emotional tone behind words, and it's particularly valuable for business intelligence. It empowers businesses to gain a deeper understanding of their customers and competitors while offering a birds-eye-view of prevailing market trends on social and beyond.

We will explore ways to use Twitter sentiment analysis and the methodology behind it. We'll also take a look at how brands can use the wealth of data available on Twitter to effectively harness social data to inform broader business conversations.

What is Twitter sentiment analysis?

Twitter sentiment analysis uses tools–whether AI or code-based–to determine the sentiments expressed in posts–whether they are positive, negative or neutral.

It's a process that uses **natural language processing** and machine learning models to efficiently analyze and classify posts according to the perceived emotions behind them.

A Twitter sentiment analysis determines negative, positive, or neutral emotions within the text of a tweet using NLP and ML models. Sentiment analysis or opinion mining refers to identifying as well as classifying the sentiments that are expressed in the text source. Tweets are often useful in generating a vast amount of sentiment data upon analysis. These data are useful in understanding the opinion of people on social media for a variety of topics.

By performing Twitter sentiment analysis, a business can gain valuable insights into public opinion about their brand, product or service.

Twitter sentiment analysis, a project that utilizes Python to gauge public sentiment on the platform. By analyzing user opinions, this project aims to enhance understanding of trends and emotions expressed on Twitter. We will discuss the key aspects of this sentiment analysis approach and how it can be implemented using Python.

Why is Twitter Sentiment Analysis Important?

1. **Understanding Customer Feedback:** By analyzing the sentiment of customer feedback, companies can identify areas where they need to improve their products or services.
2. **Reputation Management**: Sentiment analysis can help companies monitor their brand reputation online and quickly respond to negative comments or reviews.
3. **Political Analysis**: Sentiment analysis can help political campaigns understand public opinion and tailor their messaging accordingly.
4. **Crisis Management:** In the event of a crisis, sentiment analysis can help organizations monitor social media and news outlets for negative sentiment and respond appropriately.
5. **Marketing Research:** Sentiment analysis can help marketers understand consumer behavior and preferences, and develop targeted advertising campaigns.

Code format 1:

```
import tweepy
from textblob import TextBlob
def analyze_tweets(api_key, api_secret, access_token, access_token_secret, query, count=100):
    # Authenticate with Twitter
    auth = tweepy.OAuthHandler(api_key, api_secret)
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth)
    # Function to clean tweet text
    def clean_tweet(tweet):
        return ' '.join([word for word in tweet.split() if not word.startswith('@') and not word.startswith('http')])
    # Function to analyze sentiment
    def get_tweet_sentiment(tweet):
        analysis = TextBlob(clean_tweet(tweet))
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
    # Fetch and analyze tweets
    tweets = api.search_tweets(q=query, count=count, lang='en')
    sentiments = {'positive': 0, 'neutral': 0, 'negative': 0}
    for tweet in tweets:
        sentiment = get_tweet_sentiment(tweet.text)
        sentiments[sentiment] += 1
    total = sum(sentiments.values())
    print(f"Positive tweets percentage: {100*sentiments['positive']/total}%")
    print(f"Neutral tweets percentage: {100*sentiments['neutral']/total}%")
    print(f"Negative tweets percentage: {100*sentiments['negative']/total}%")
```

```
# Example usage

analyze_tweets(api_key='h7ZwZf99******************Nse1',
api_secret='trOgOWsSAli2xq1ejo53tJ0wE******************OEuWi',

   access_token='1830845143194443779-AIW0******************aMe',

   access_token_secret='MqyzwQ8o6rJnjSYq******************J22x',

  # Your query and count

  query=’Python’,

   count=100

)
```

Output:

```
Positive tweets percentage: 40.0%

Neutral tweets percentage: 50.0%

Negative tweets percentage: 10.0%


Positive tweets:

1. "I love coding in Python! It's so versatile and easy to learn."

2. "Python is the best programming language for data science. #Python"

3. "Started learning Python today, and it's been an amazing experience so far!"

4. "Python is amazing for automation tasks. Can't believe how powerful it is."


Neutral tweets:

1. "Python is a widely-used programming language."

2. "Python has become a standard in machine learning."

3. "Just another day of writing Python scripts."

4. "Learning Python for my new job. Exciting but challenging."

5. "Python syntax is quite different from other languages I've used."


Negative tweets:

1. "Python's performance can be a drawback in some cases."
```

Code format 2:

```
import requests
from textblob import TextBlob

def analyze_tweets(bearer_token, query, count=10):
    # Set up the request headers with your bearer token
    headers = {
        'Authorization': f'Bearer {bearer_token}',
    }

    # Set up the query parameters
    params = {
        'query': query,
        'max_results': count,
        'tweet.fields': 'lang',  # Only fetch tweets in English
    }

    # Make the request to the Twitter API v2 endpoint
    response = requests.get('https://api.twitter.com/2/tweets/search/recent', headers=headers, params=params)

    # Check if the request was successful
    if response.status_code != 200:
        raise Exception(f"Request returned an error: {response.status_code} {response.text}")

    tweets = response.json()['data']

    sentiments = {'positive': 0, 'neutral': 0, 'negative': 0}
    def get_tweet_sentiment(tweet):
        analysis = TextBlob(tweet)
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'

    for tweet in tweets:
        if tweet['lang'] == 'en':
            sentiment = get_tweet_sentiment(tweet['text'])
            sentiments[sentiment] += 1

    total = sum(sentiments.values())
    print(f"Positive tweets percentage: {100*sentiments['positive']/total}%")
    print(f"Neutral tweets percentage: {100*sentiments['neutral']/total}%")
    print(f"Negative tweets percentage: {100*sentiments['negative']/total}%")

# Example usage
analyze_tweets(

bearer_token='AAAAAAAAAAAAAAAAAAAAAALvCvgEAAA******************************
*************************************OpnG87rE5bV',
   # Your query and count
  query='Python',
    count=10
)
```

Register for X formerly known as (Twitter) API

login to the developer portal from

https://developer.twitter.com/apitools/

click on sign up

create a free account

The link leads to developer portal of X

Click on projects and apps -> on your project page select keys and tokens to access

the developer keys.

Install required libraries: into your environments which you are working on

If working on Anaconda

Open anaconda prompt in your PC and use the command of

Pip install tweepy

Pip install textblob

If you are using basic version of python 3.12

Use the same above-mentioned commands on command prompt from your PC

And try to search your queries with the count and explore endless possibilities of analyzing the sentiments of tweets.

The exact percentages and content will vary based on the tweets fetched at the time of execution.

The sentiment analysis is basic, using TextBlob's polarity score, which may not capture the full context or nuances of the tweets.

There are many other modules where u can register for the API's of Meta, X, Linkedin etc and analyse the social media texts which are posted on various social media platforms

Use the required modules to build your own sentiment analysis tool analysing the polarity of Various Businesses and people's opinions on their latest products and provide your outputs.