A

project Report

On

“Twitter Sentiment Analysis”

B.Sc. (Computer Science)
SemVI Examination

Submitted By:

Shubham Dhole

Roll no: 2025007

Department of Computer Science

R.K.TALREJACOLLEGE OF ARTS, SCIENCE AND COMMERCE

ULHASNAGAR-3

UNIVERSITY OF MUMBAI

2020-21

PREFACE

Over the last few years, Computer already had a considerable impact on many aspects of our society. A few organizations are claiming to offer AI-based sentiment analysis solutions to companies, explicitly their marketing and product development divisions. Indeed, probably the biggest tech organizations are offering these solutions for medium and huge enterprises. We found these solutions are expected to assist organizations to solve many problems.

Sentiment analysis is an ability of natural language processing, a sort of artificial intelligence. It could permit organizations to look through social media, the overall web, and their excess of client support tickets for what their prospects and clients think about their brand and products. This can thus permit the organization to make advertisements and products that their prospects and customers will like, hence expanding the conversion rates of marketing campaigns.

INDEX

ABSTRACT

This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users [24] - out of which 100 million are active users and half of them log on twitter on a daily basis - generating nearly 250 million tweets per day [20]. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets. Analysing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream

Sentiment analysis Overview:

In the field of social media data analytics, one popular area of research is the sentiment analysis of twitter data. Twitter is one of the most popular social media platforms in the world, with 330 million monthly active users and 500 million tweets sent each day. By carefully analyzing the sentiment of these tweets—whether they are positive, negative, or neutral, for example—we can learn a lot about how people feel about certain topics.

Understanding the sentiment of tweets is important for a variety of reasons: business marketing, politics, public behavior analysis, and information gathering are just a few examples. Sentiment analysis of twitter data can help marketers understand the customer response to product launches and marketing campaigns, and it can also help political parties understand the public response to policy changes or announcements.

However, Twitter data analysis is no simple task. There are something like ~6000 tweets released every second. That's a lot of Twitter data! And though it's easy for humans to interpret the sentiment of a tweet, human sentiment analysis is simply not scalable.

In this Project, we're going to look at building a scalable system for Twitter sentiment analysis, to help us better understand the role of machine learning in social media data analytics.

Technology used

- Jupyter Notebook(Python IDE)

Important libraries

- Textblob
- Wordcloud
- Tweepy
- Pandas
- Numpy
- Re
- Matplot lib

Requirements Specification

Hardware requirement

- I5 processor.
- 8 GB of RAM recommended
- GPU required

Software requirement

- Windows 10 version (64 bits)
- Jupyter Notebook(Python IDE)

What is sentiment analysis?

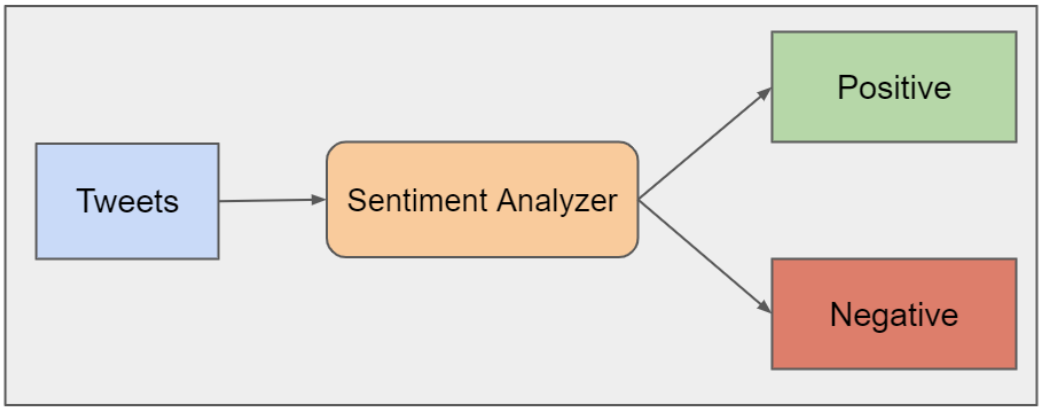
- Sentiment analysis is the automated process of identifying and classifying subjective information in text data. This might be an opinion, a judgment, or a feeling about a particular topic or product feature.
- It is the process of classifying text as either *positive*, *negative*, or *neutral*. Machine learning techniques are used to evaluate a piece of text and determine the sentiment behind it.
- Sentiment analysis uses Natural Language Processing (NLP) to make sense of human language, and machine learning to automatically deliver accurate results.
- The most common type of sentiment analysis is 'polarity detection' and involves classifying statements as *positive*, *negative* or *neutral*. A polarity sentiment analysis model, for example, automatically tags this tweet as *positive*:

Sentiment analysis using twitter

- Twitter allows businesses to engage personally with consumers. However, there's so much data on Twitter that it can be hard for brands to prioritize mentions that could harm their business.
- That's why sentiment analysis, a tool that automatically monitors emotions in conversations on social media platforms, has become a key instrument in social media marketing strategies.
- Carefully listening to voice of the customer on Twitter using sentiment analysis allows companies to understand their audience, keep on top of what's being said about their brand – and their competitors – and discover new trends in the industry.

Problem: Identifying Negative Sentiment in Tweets

- In this Project, we'll learn how to identify tweets with a negative sentiment. To do so, we'll create a sentiment analyzer to classify positive and negative tweets in text format. Though we'll be using our classifier for Twitter data analysis, it can also be used to analyze text data from other sources as well.



- Through the course of the Project, we are going to look at datasets, various text processing, and embedding techniques, and then employ a machine learning model to process our data.

How to Perform Sentiment Analysis on your Twitter Data

Performing sentiment analysis on Twitter data involves five steps:

1. Gather relevant Twitter data
2. Clean your data using pre-processing techniques
3. Create a sentiment analysis machine learning model
4. Analyze your Twitter data using your sentiment analysis model
5. Visualize the results of your Twitter sentiment analysis

Process model

Dataset file : config.csv- tweeter authentication token file

To get access to tweeter account e.g wwe(world wide wrestling entertainment)

We have taken record of 1000.

1. Creating dataframe
2. Reading config.csv file
3. Establish tweeter connection
4. Loading data from tweeter account

Twitter Sentiment Analysis Dataset

Let's start with our Twitter data. We will use the open-source Twitter Tweets Data for Sentiment Analysis dataset.

The target variable for this dataset is 'label', which maps negative tweets to 1, and anything else to 0. Think of the target variable as what you're trying to predict. For our machine learning problem, we'll train a classification model on this data so it can predict the class of any new tweets we give it.

A snapshot of the data is presented in the image below.

	Tweets
0	Before @TBARRetribution & @MACEtheWRESTLER...
1	#TheBoss always bounces back. #SmackDown\n\n@S...
2	From swinging @UNBESIEGBAR_ZAR 40 times to cat...
3	Cast your vote now for #WrestleMania 36 in @Th...
4	RT @WWENetwork: So ... which victories made @m...

Text Processing

Data usually comes from a variety of different sources and is often in a variety of different formats. For this reason, cleaning your raw data is an essential part of preparing your dataset. However, cleaning is not a simple process, as text data often contain redundant and/or repetitive words. This is especially true in Twitter sentiment analysis, so processing our text data is the first step towards our solution.

The fundamental steps involved in text processing are:

- A. Cleaning of Raw Data
- B. Tokenization

A. Cleaning of Raw Data

This phase involves the deletion of words or characters that do not add value to the meaning of the text. Some of the standard cleaning steps are below:

- Lowering case
- Removal of mentions

- Removal of special characters
- Removal of hyperlinks
- Removal of numbers
- Removal of whitespaces

Lowering Case: Lowering the case of text is essential for the following reasons: The words, 'Tweet', 'TWEET', and 'tweet' all add the same value to a sentence. Lowering the case of all the words helps to reduce the dimensions by decreasing the size of the vocabulary.

Removal of mentions: Mentions are very common in tweets. However, as they don't add value for interpreting the sentiment of a tweet, we can remove them. Mentions always come in the form of '@mention', so we can remove strings that start with '@'.

Removal of special characters: This text processing technique will help to treat words like 'hurray' and 'hurray!' in the same way. At this stage, we remove all punctuation marks.

Removal of hyperlinks: Now we can remove URLs from the data. It's not uncommon for tweets to contain URLs, but we won't need to analyze them for our task.

B. Tokenization

Tokenization is the process of splitting text into smaller chunks, called tokens. Each token is an input to the machine learning algorithm as a feature. NLTK (Natural Language Toolkit) provides a utility function for tokenizing data.

There is a huge amount of data in text format. Analyzing text data is an extremely complex task for a machine as it's difficult for a machine to understand the semantics behind text. At this stage, we're going to process our text data into a machine-understandable format using word embedding.
Word Embedding is simply converting data in a text format to numerical values (or vectors) so we can give these vectors as input to a machine, and analyze the data using the concepts of algebra.

However, it's important to note that when we perform this transformation there could be data loss. The key then is to maintain an equilibrium between conversion and retaining data.

Here are two commonly used terminologies when it comes to this step.

- Each text data point is called a **Document**
- An entire set of documents is called a **Corpus**

SENTIMENT ANLAYSIS BASED ON SUBJECTIVITY AND POLARITY

Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information.

Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Sentiment polarity for an element defines the orientation of the expressed **sentiment**, i.e., it determines if the **text** expresses the positive, negative or neutral **sentiment** of the user about the entity in consideration.

WORDCLOUD:

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analysing data from social network websites.

Code and Implementation

```
#importing essential libraries
import textblob
from textblob import TextBlob
import tweepy
from wordcloud import WordCloud
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
config = pd.read_csv("C:/Users/shubh/OneDrive/Desktop/configuration.csv")
#get data
twitterApiKey= config['twitterApiKey'][0]
twitterApiSecret= config['twitterApiSecret'][0]
twitterApiAcessToken= config['twitterApiAcessToken'][0]
twitterApiAcessTokenSecret= config['twitterApiAcessTokenSecret'][0]
#Twitter API Credentials and establishing connection throught twitter api by
providing the keyword for which u want to search relate tweets

auth = tweepy.OAuthHandler(twitterApiKey,twitterApiSecret)
auth.set_access_token(twitterApiAcessToken,twitterApiAcessTokenSecret)
twitterApi= tweepy.API(auth,wait_on_rate_limit=True)
twitterAccount= "WWE"
tweets = tweepy.Cursor(twitterApi.user_timeline,
                       screen_name= twitterAccount,
                       count=None,
```

```
                since_id=None,

max_id=None,trim_user=True,exclude_replies=True,contributor_details=False,
                include_entities=False
               ).items(1000);
#creating a DataFrame with acolumn called tweets
df =pd.DataFrame(data=[tweet.text for tweet in tweets],columns=['Tweets'])

#show the first 5 rows of data
df.head()
```

	Tweets
0	Just how distracted IS @NiaJaxWWE? 😬\n\n#WWERa...
1	#USChampion @WWESheamus offered HIS version of...
2	Before @TBARRetribution & @MACEtheWRESTLER...
3	#TheBoss always bounces back. #SmackDown\n\n@S...
4	From swinging @UNBESIEGBAR_ZAR 40 times to cat...

```
#Clean the text
#Creating a function to clean the tweets
def cleanTxt(text):
  text = re.sub(r'@[A-Za-z0-9]+', '',text) #Remove @metions
  text = re.sub(r'#','',text)#Removing the '#' symbol
  text = re.sub(r'RT[\s]+','',text)#Removing RT
  text = re.sub(r'https?:\/\/\S+','',text)#Removing the Hyperlink
  text = re.sub(r':','',text)#Removing the extra semicolons
```

```
 return text
df['Tweets']= df['Tweets'].apply(cleanTxt)
#Show The Clean Data
df
```

	Tweets
0	Just how distracted IS ? 😬\n\nWWERaw
1	USChampion offered HIS version of the Open Ch...
2	Before & dished out further damage on , ...
3	TheBoss always bounces back. SmackDown\n\n 💙💙
4	From swinging _ZAR 40 times to catching oppone...
...	...
995	Congrats champ 😈 _WWE ! WrestleMania
996	WWENXT ROYALTY. 😈 🐻\n\nCouldn't be more proud...
997	Hell yeah! Congrats champ. _WWE 🤘\n(Can I bor...
998	WrestleMania NXTTakeOver\n\nWeAreNXT ❤️🤍
999	Yaaaas _WWE !!!! Congrats wifey!!! 💞😘

1000 rows × 1 columns

```
# Creating a Funtion to get the subjectivity
def getSubjectivity(text):
  return TextBlob(text).sentiment.subjectivity

#Creating a Function to get the Polarity
def getPolarity(text):
  return TextBlob(text).sentiment.polarity

#Creating two new columns
df['Subjectivity'] =df['Tweets'].apply(getSubjectivity)
df['Polarity'] = df['Tweets'].apply(getPolarity)

#Show the new dataframe with thh new columns
Df
```

	Tweets	Subjectivity	Polarity
0	Just how distracted IS ? 😬\n\nWWERaw	0.00	0.000000
1	USChampion offered HIS version of the Open Ch...	0.50	0.000000
2	Before & dished out further damage on , ...	0.50	0.000000
3	TheBoss always bounces back. SmackDown\n\n 🤍🤍	0.00	0.000000
4	From swinging _ZAR 40 times to catching oppone...	0.45	0.300000
...	...	...	...
995	Congrats champ 😻 _WWE ! WrestleMania	0.00	0.000000
996	WWENXT ROYALTY. 😻 😻\n\nCouldn't be more proud...	0.50	0.433333

	Tweets	Subjectivity	Polarity
997	Hell yeah! Congrats champ. _WWE 🤘\n(Can I bor...	0.00	0.000000
998	WrestleMania NXTTakeOver\n\nWeAreNXT ❤♡	0.00	0.000000
999	Yaaaas _WWE !!!! Congrats wifey!!! 💞😘	0.00	0.000000

1000 rows × 3 columns

Plot the WorldCloud

allWords = ' '.join([twts for twts in df['Tweets']])

wordCloud = WordCloud(width =500,height=300,random_state =21,max_font_size=119).generate(allWords)

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

plt.axis('off')

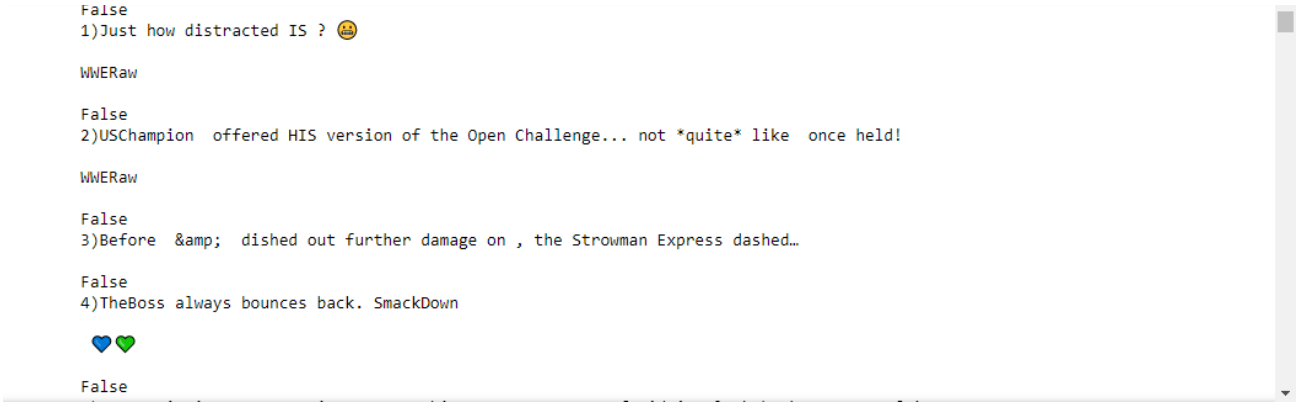
plt.show()

```
#Create a function to compute the negative,neutral and positive analysis
def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
df['Analysis']= df['Polarity'].apply(getAnalysis)
#Show the  dataframe
df
```

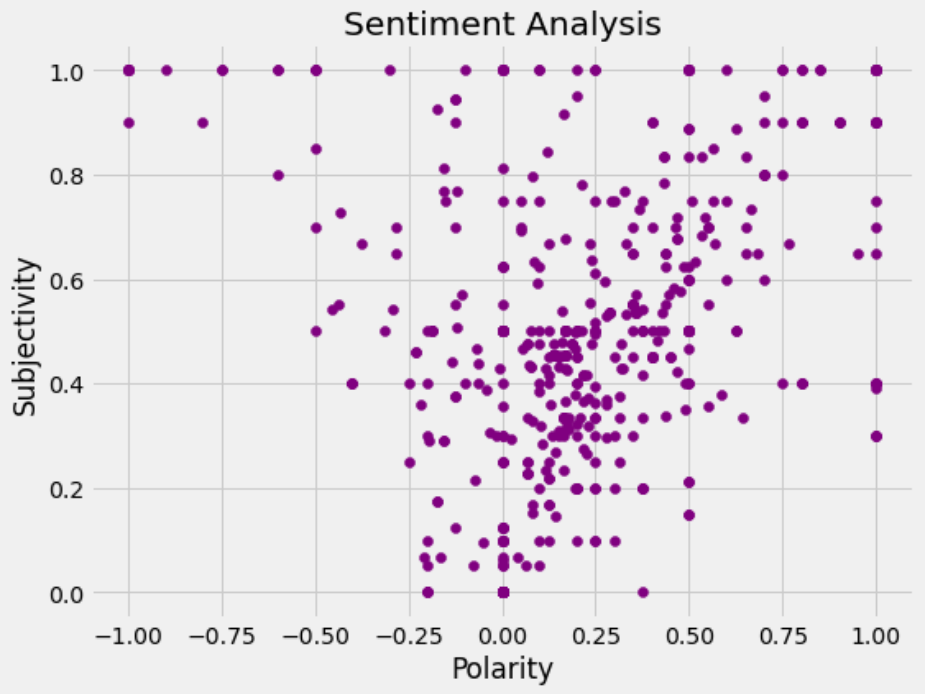
	Tweets	Subjectivity	Polarity	Analysis
0	Just how distracted IS ? 😬\n\nWWERaw	0.00	0.000000	Neutral
1	USChampion offered HIS version of the Open Ch...	0.50	0.000000	Neutral
2	Before & dished out further damage on , ...	0.50	0.000000	Neutral
3	TheBoss always bounces back. SmackDown\n\n 💟💟	0.00	0.000000	Neutral
4	From swinging _ZAR 40 times to catching oppone...	0.45	0.300000	Positive
...	...	...	...	...
995	Congrats champ 😻 _WWE ! WrestleMania	0.00	0.000000	Neutral
996	WWENXT ROYALTY. 😻 😁\n\nCouldn't be more proud...	0.50	0.433333	Positive
997	Hell yeah! Congrats champ. _WWE 🤘\n(Can I bor...	0.00	0.000000	Neutral
998	WrestleMania NXTTakeOver\n\nWeAreNXT ❤️💟	0.00	0.000000	Neutral
999	Yaaaas _WWE !!!! Congrats wifey!!! 💞😘	0.00	0.000000	Neutral

1000 rows × 4 columns

```
# Print all of the positive tweets
 j=1
 sortedDF = df.sort_values(by=['Polarity'])
 for i in range(0,sortedDF.shape[0]):
   if (sortedDF['Analysis'][i]=='Positive'):
    print(str (j)+') '+sortedDF['Tweets'][i])
    print()
    j =j+1
```




```
#plot the polarity and subjectivity
plt.figure(figsize=(8,6))
for i in range(0,df.shape[0]):
  plt.scatter(df['Polarity'][i],df['Subjectivity'][i],color='purple')
plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```



#get the percentage of the positive tweets

positive = df[df.Analysis == 'Positive']

print(str(positive.shape[0]/(df.shape[0])*100)+"% 0f postitve tweets")

pos=positive.shape[0]/df.shape[0]*100

```
37.5% 0f postitve tweets
```

#get the percentage of the negative tweets

negative = df[df.Analysis == 'Negative']

print(str(negative.shape[0]/(df.shape[0])*100)+"% 0f negative tweets")

neg=negative.shape[0]/df.shape[0]*100

```
8.4% 0f negative tweets
```

#get the percentage of the neutral tweets

neutral = df[df.Analysis == 'Neutral']

print(str(neutral.shape[0]/(df.shape[0])*100)+"% 0f neutral tweets")

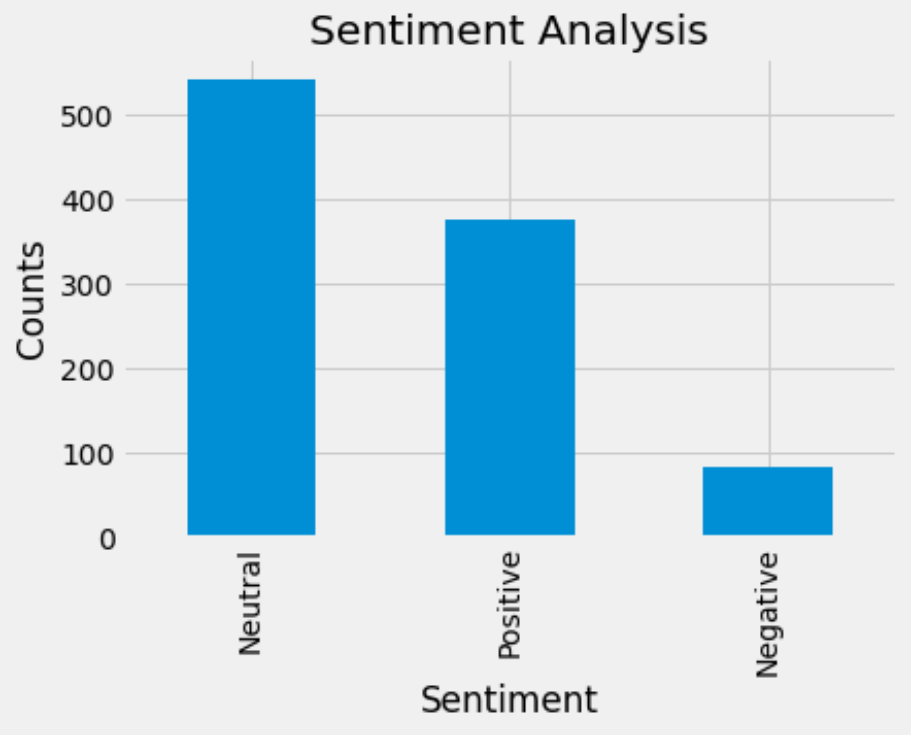
neut=neutral.shape[0]/df.shape[0]*100

```
54.1% 0f neutral tweets
```

#Showing the value counts

```
df['Analysis'].value_counts()


#plot and visualize the counts
plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
df['Analysis'].value_counts().plot(kind='bar')
plt.show()
```



Conclusion:

Advanced analytics tools joined with social listening can be utilized for real-time experimentation and better comprehension of customer sentiment about product and service attributes. The capacity to cut up structured and unstructured data empowers marketers to think of micro-targeting strategies and a chance to quickly engage with and delight customers.

While there is still space for innovativeness, it is not, at this point a unique competitive differentiator. Progressively, leading companies are known for harnessing the power of information to tune in to their clients intently, comprehend them better than other people and react in manners that make their clients feel significant and connected with the brand.

Reference and Bibliography

Many projects and books have been referred to develop this project, but out of those many resources only a few of them have been very handy in the project development.

A few references are as follows:-

- **Mastering Jupyter Notebook.**
- **Star Python**
- **Microsoft Office**
- **https://medium.com/seek-blog/your-guide-to-sentiment-analysis-344d43d225a7**
- **https://unamo.com/blog/social/sentiment-analysis-social-media-monitoring**
- **https://www.revuze.it/blog/sentiment-analysis/**