Group A8

Sentiment Analysis of Twitter Data

Project Report of CMPE 257 Spring 2018

Under the Guidance of: Prof. Bhavana Bhasker

By: Bohan Liu (011456237)
Karthikeya Rao (011448437)
YuYu Chen (007855939)

1. Abstract

Social media has changed the world in many ways. It allows people from all over the world to interact with each other and made it possible to connect with friends and family who are halfway across the world. Twitter, for example allows the user to send a maximum of 140 or 280 characters per tweet about their current status in realtime. It can also be used for micro-blogging or even as a platform for news consumption. Twitter has also become a platform for users to express their opinion freely, allowing them to exercise free-speech. While there are many positives opinions, expressing opinions freely has also encouraged a lot of negative opinions, such as trash-talking, cyber-bullying, and misinformation.

With the knowledge acquired in machine learning, sentiment analysis will be performed on Twitter data in order to differentiate positive and negative tweets. Twitter data will be collected from multiple sources on twitter and the dataset will be cleaned for feature extraction. The model will be trained and evaluated on a new test dataset. In the end one shall be able to see the accuracy of the test result, also be able to see the positive and negative labels of each individual tweets.

There are difficulties along the way while we implementing the algorithms, especially in term of successfully utilizing algorithms to correctly classify and predict the data sets. Through the careful study of the theories and times of troubleshoot, we were able to carry out the prediction result.

2. Problem Statement

We have huge chunks of data coming online every second. Twitter is one such platform which has millions of users around the globe who tweets frequently about the happenings around them and information they gained and their thoughts. If we want to collectively know thoughts of people who are tweeting on a particular topic or based in a particular region of the globe or on language and other aspects, it would be very difficult.

The Problem to know what people think about particular topic on a global scale is very difficult. However assuming people use twitter as a platform to express their feelings on various topics frequently and twitter having a large user base has enabled us to know opinion of a section of people about a topic to certain extent. Machine Learning approach to analyze these tweets enable us to determine the overall thinking of the users via their tweets.

3. Existing System

Twitter gives us it's API's (Application Programming Interface) and python has a library called tweepy to do analytics mentioned above using twitter. However there are other numerous python libraries and other tools to analyze twitter data, tweepy being most used and reliable the tech society used this the most. We get access to use twitter data on agreeing some terms and generating keys on our twitter account.

Post this we do analysis on the twitter data using various approaches in a methodical way of data tapping, data ingestion, data cleansing, introduction of data to Machine Learning model, Building the model, testing and verifying. However, it is mostly built for a particular scenario in most systems.

There are also existing libraries in Python called Natural Language Toolkit (NLTK) and Textblob that allows the user to perform text analytics. NLTK library already has a database of texts that enables the developers build a machine learning model as fast as possible, such as categorized positive and negative tweets, as well as positive and negative opinion lexicons. Textblob to is built on top of the NLTK library, so it abstracts away the implementation details, hence enabling developers to quickly text analytic models. There already exists API in Textblob that allows the user to determine the polarity and objectivity of a text by passing the text as a parameter.

4. Proposed System

The proposed system consists of training and testing. The training portion of the system is trained using a supervised learning approach where large dataset of tweets and label is fed into the machine learning algorithm. This is done by vectorizing the dataset, which also does the preprocessing, tokenizing, and removal of stopwords. The output of vectorizing is a list of feature vectors that will be fed into the machine learning algorithm.

In our model we test the performance of the model comparing with different algorithms such as SVM, Naive Bayes and others and considering the algorithm which is giving best results for the train dataset. Once the Algorithm is trained with 75% of the training dataset, we test it with 25% of the remaining training dataset and consider the algorithm giving us highest accuracy.

Later we run this model saved in the pickle file with the live stream data of the twitter, filtered based on the search term. This model gives us the sentiment of the tweets along with the most used keywords in the tweets and classifying its used case more in a positive or in a negative way.

Since we tested the trained model on a live data, no labels were given. An unsupervised learning approach was used to label the live tweets as a positive or negative tweet. The unsupervised algorithm used was the Pointwise Mutual Information - Information Retrieval algorithm (PMI) [21]. The following two equations are used to determine the semantic orientation of words.

$$\mathrm{PMI}(\mathit{word}_1, \mathit{word}_2) = \log_2 \left[\frac{\mathrm{p}(\mathit{word}_1\ \&\ \mathit{word}_2)}{\mathrm{p}(\mathit{word}_1)\ \mathrm{p}(\mathit{word}_2)} \right] \quad (1)$$

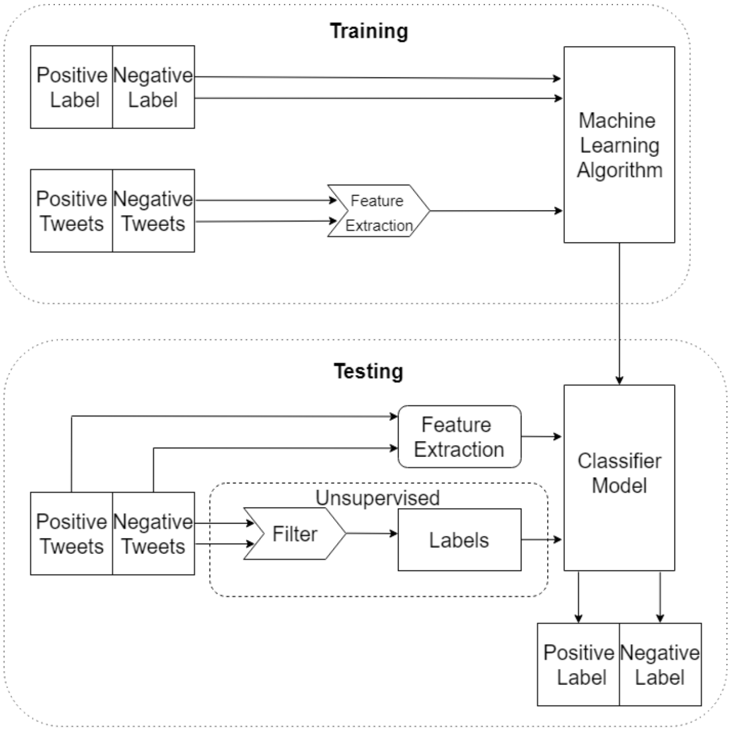
$$\begin{aligned} \mathrm{SO}(\mathit{phrase}) = {} & \mathrm{PMI}(\mathit{phrase}, \text{“excellent”}) \\ & - \mathrm{PMI}(\mathit{phrase}, \text{“poor”}) \end{aligned} \quad (2)$$

The PMI of word1 and word2 is calculated by the log of the probability of word1 and word2 occurring together divided by the probability of word1 occurring multiplying the probability of word2 occurring. The semantic orientation of a word is calculated by the summation of PMI of word1 and word2, where word2 is a dictionary of positive lexicon, minus the summation of PMI of word1 and word2, where word2 is a dictionary of negative lexicon. The semantic orientation of a word is positive if it is greater than zero, negative if it is less than zero, and neutral if it is zero. The weight of the orientation is determined by a higher number (being associated with positive words, ie. 100) or a negative number (being associated with negative words, ie. -100). The semantic orientation of a tweet is calculated by the summing all weights of a words that exist in a tweet.

5. Architecture/ Computation

Below diagram represents are Architecture of our system involving training and testing cases.

In the training phase, we load the nltk data to our machine learning algorithm to perform supervised learning using SVM. Later we test it on the same data as well. As nltk data is cleaned, tokenized and normalized, we need not perform these operations on this dataset. However once the Algorithm has learnt, we load to onto testing phase from the stored pickle file, where we run live stream twitter data after cleaning, tokenizing, normalizing and predicting the sentiment before for each tweet using other methods and later load it to ML algorithm for it’s prediction of positive and negative tweets.



In the above diagram, the filtering includes removal of stop words, tokenizing and normalizing, and Labels being the attached sentiment to each tweet using the two approaches.

6. Testing/Experiments

Data exploration was done on training data in order to determine if the dataset is bias towards a label. It is also done for preprocessing, tokenizing, and removal of stopwords, as well as finding bigrams and trigrams.

Testing was done on the training dataset by splitting it into 75% training data and 25% testing data. Different machine learning algorithms were used, such as SVM, Bernoulli Naive Bayes, and Multinomial Naive Bayes, in order to benchmark its accuracy.

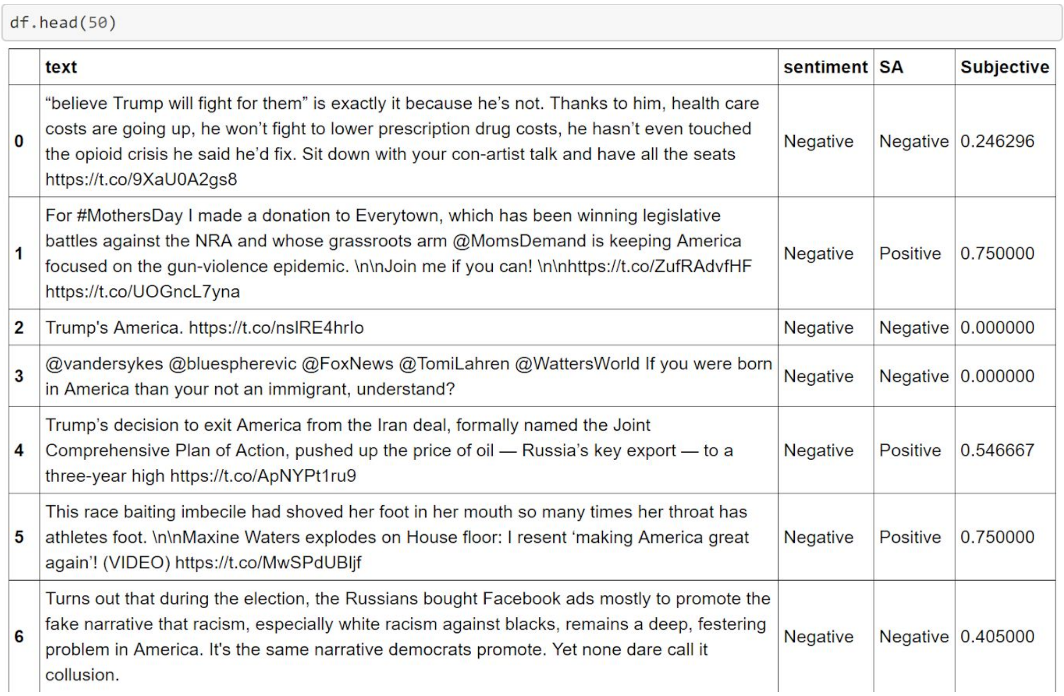
Benchmarking:

Benchmarking was done in order to determine which model to use on the live stream twitter data. Out of the three mentioned, SVM performed the best with an accuracy of almost 80%. Bernoulli Naive Bayes scored an accuracy of around 76% and Multinomial Naive Bayes scored an accuracy of around 77%. Therefore, SVM was chosen and saved to a pickle file.

7. Results

After saving the model in a pickle file, inference test was done by loading the model onto a new file where testing on live twitter data was done. Since the live dataset are unlabeled, PMI was used to determine the semantic orientation. Our own implementation was compared against Text blob's implementation of semantic orientation.

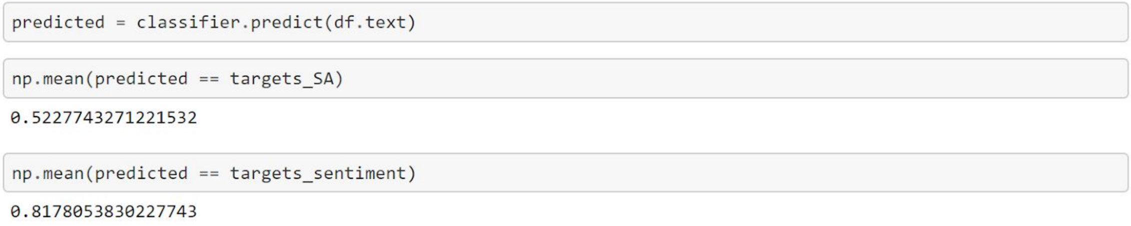
In the figure below, the "sentiment" column is the PMI implementation and "SA and Subject" column is Text blob's implementation.



```
df.head(50)
```

	text	sentiment	SA	Subjective
0	"believe Trump will fight for them" is exactly it because he's not. Thanks to him, health care costs are going up, he won't fight to lower prescription drug costs, he hasn't even touched the opioid crisis he said he'd fix. Sit down with your con-artist talk and have all the seats https://t.co/9XaU0A2gs8	Negative	Negative	0.246296
1	For #MothersDay I made a donation to Everytown, which has been winning legislative battles against the NRA and whose grassroots arm @MomsDemand is keeping America focused on the gun-violence epidemic. \n\nJoin me if you can! \n\nhttps://t.co/ZufRAdvfHF https://t.co/UOGncL7yna	Negative	Positive	0.750000
2	Trump's America. https://t.co/nslRE4hrIo	Negative	Negative	0.000000
3	@vandersykes @bluespherevic @FoxNews @TomiLahren @WattersWorld If you were born in America than your not an immigrant, understand?	Negative	Negative	0.000000
4	Trump's decision to exit America from the Iran deal, formally named the Joint Comprehensive Plan of Action, pushed up the price of oil — Russia's key export — to a three-year high https://t.co/ApNYPt1ru9	Negative	Positive	0.546667
5	This race baiting imbecile had shoved her foot in her mouth so many times her throat has athletes foot. \n\nMaxine Waters explodes on House floor: I resent 'making America great again'! (VIDEO) https://t.co/MwSPdUBljf	Negative	Positive	0.750000
6	Turns out that during the election, the Russians bought Facebook ads mostly to promote the fake narrative that racism, especially white racism against blacks, remains a deep, festering problem in America. It's the same narrative democrats promote. Yet none dare call it collusion.	Negative	Negative	0.405000

In the figure below, we can see that PMI implementation has an accuracy of around 81%, almost 30% better compared to Text blob's implementation.



```
predicted = classifier.predict(df.text)
```

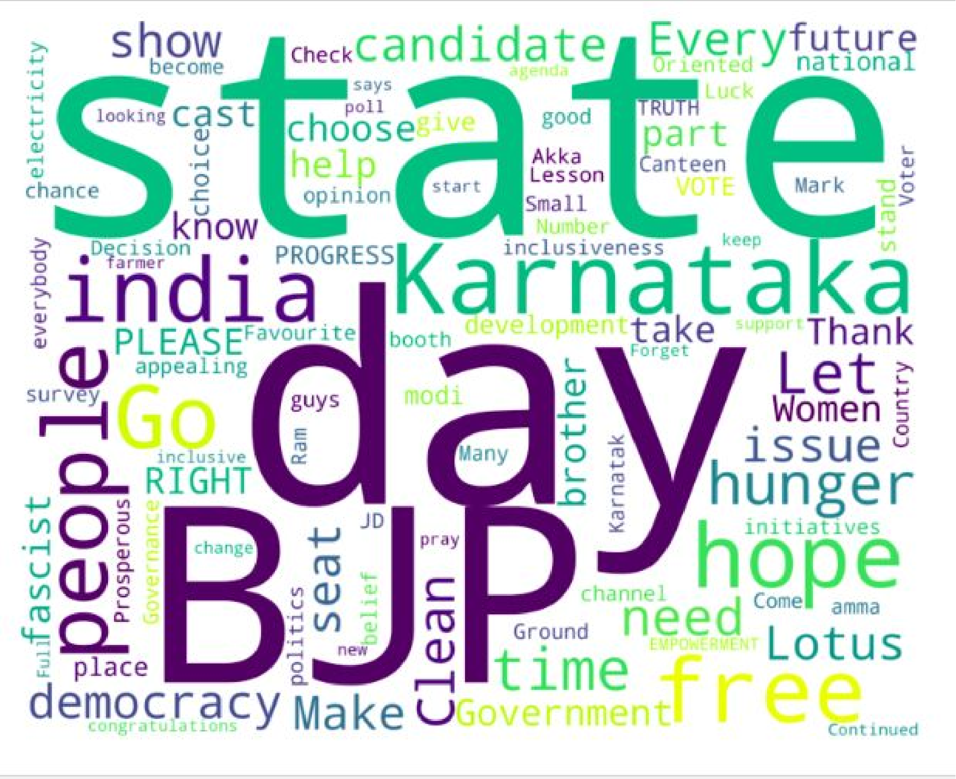
```
np.mean(predicted == targets_SA)
```

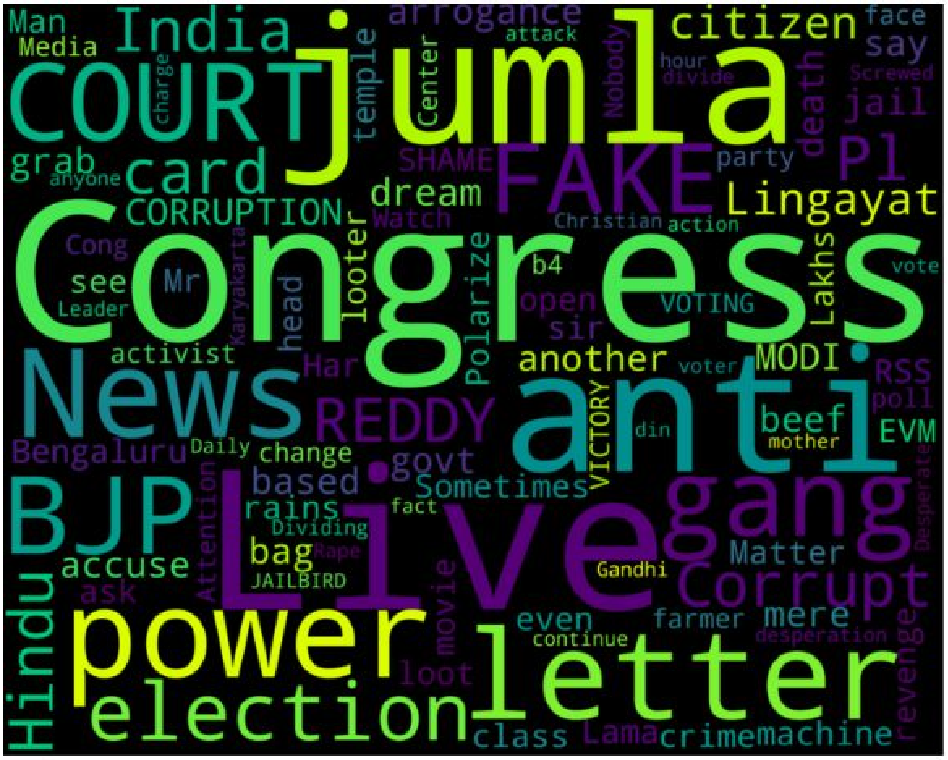
```
0.5227743271221532
```

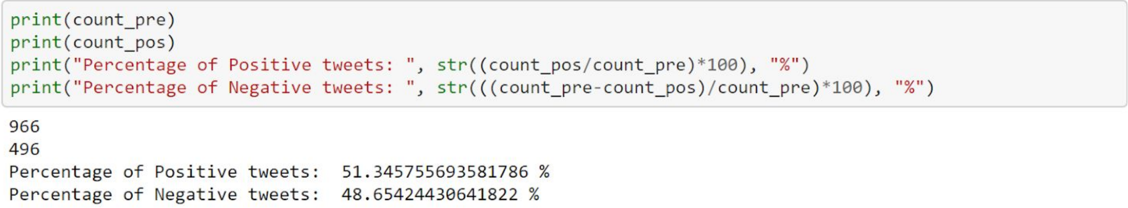
```
np.mean(predicted == targets_sentiment)
```

```
0.8178053830227743
```

In the figures below we represent the positive sentiment and negative sentiment tweets in a visually pleasing format. Most frequent words, where white background represents the positive tweets and black background represents the negative.


state
day
BJP
Karnataka
india
people
Go
show
become
Check
candidate
Every
future
Oriented
national
agenda
says
poll
looking
cast
choose
give
good
TRUTH
Luck
part
electricity
choice
help
Akka
Lesson
Canteen
Voter
chance
opinion
start
Small
VOTE
Mark
stand
know
Number
Decision
farmer
PROGRESS
inclusiveness
keep
everybody
PLEASE
Favourite
booth
development
take
support
Thank
appealing
Forget
Let
survey
guys
modi
Women
Country
issue
Ram
Many
inclusive
RIGHT
brother
hunger
Karnatak
initiatives
fascist
Prosperous
Governance
JD
pray
hope
change
seat
belief
politics
Clean
channel
Come
amma
need
Ground
Full
new
EMPOWERMENT
Lotus
place
time
free
democracy
Make
Government
congratulations
Continued


Man
Media
India
jumla
arrogance
attack
citizen
face
say
COURT
charge
temple
Center
Nobody
hour
divide
death
Screwed
jail
grab
anyone
card
FAKE
SHAME
party
Pl
CORRUPTION
dream
Watch
Christian
action
Lingayat
Congress
Cong
Karyakarta
looter
Polarize
b4
Lakhs
vote
see
Mr
head
open
VOTING
Leader
sir
activist
another
MODI
News
Har
anti
voter
RSS
REDDY
VICTORY
poll
din
beef
EVM
Bengaluru
Daily
change
mother
based
govt
Desperate
Sometimes
BJP
Live
gang
rains
fact
Attention
Dividing
bag
Rape
accuse
Matter
Gandhi
ask
JAILBIRD
Corrupt
Hindu
movie
even
farmer
mere
power
continue
desperation
revenge
loot
letter
election
class
Lama
crime
machine



```
print(count_pre)
print(count_pos)
print("Percentage of Positive tweets: ", str((count_pos/count_pre)*100), "%")
print("Percentage of Negative tweets: ", str(((count_pre-count_pos)/count_pre)*100), "%")
```

```
966
496
Percentage of Positive tweets:  51.345755693581786 %
Percentage of Negative tweets:  48.65424430641822 %
```

The model used to predict the accuracy predicted around 51% of tweets as positive and around 48% as negative. As of now, the only way to test the accuracy is to manually check the correctness by reading each tweet and determining it as positive or negative. Post this the sentiment analysis approach between the two which gave most near sentiment on manual view was chosen as the final one.

8. Conclusion

To conclude we have implemented a system using machine learning, that can identify the sentiment of live tweets using SVM algorithm and classify it as a positive or a negative tweet. Although there is a lot more to add on to this, currently for the scope of the coursework and with limited time frame we were able to implement this successfully and also get the most frequent words used in the collected tweets and represent them in a visually pleasing format using word-cloud library tools. We were also able to get the total percentage of the positive and negative tweets in the collection after getting the sentiment of all the tweets.

9. Future work

Although we have worked on many concepts and algorithms in this project. The future scope of this project is to classify it as positive negative and also neutral tweets. These posts also has to be tested, and validated for correctness. Currently we are unable to handle emotions in the language such as sarcasm, puns etc, in the tweets to classify it correctly as positive or negative tweet. We have handled only for english language as we find lot of preprocessing libraries which reduce our work. System which can handle and classify different languages and mix of languages tweets, which people usually tweet in can be a big upgrade.

We were unable to implement a live animation of live stream data shown on graph as in involved ffmpeg libraries and their installations and a lot of features involving heavy CPU and RAM which we could not afford to handle on our local computers.

Customizing the system to take in tweets based on user specific inputs such as language, location, inclusion and exclusion of previous data etc would be more effective. Mapping each word of tweet with a weight and determining its overall positivity and negativity, and color coding it in the word cloud respectively which we couldn't achieve due to lack of time.

Moving this code on the cloud and running it at fast pace adding the above features would be a great deal to identify the current mindset of a specified group of people.

10. References:

Following sites and sources were used for implementing this project. We have used some references for our understanding only and few of them to implement small features in the project.

[1]https://apps.twitter.com/

[2]https://stackoverflow.com/questions/35075672/filter-tweets-in-tweepy-streamlistener-on-data-method

[3]https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[4]https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/

[5]https://github.com/shwetkm/Real-time-Twitter-sentiment-analysis-on-Baahubali-and-Trump/

[6]https://github.com/yogeshg/Twitter-Sentiment

[7]https://www.youtube.com/watch?v=fSyH8PALzEA

[8]https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-coding-edd8f1cf8f2d

[9]https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/

[10]http://ipullrank.com/step-step-twitter-sentiment-analysis-visualizing-united-airlines-pr-crisis/

[11]https://dev.to/rodolfoferro/sentiment-analysis-on-trumpss-tweets-using-python-

[12]https://marcobonzanini.com/2015/01/19/sentiment-analysis-with-python-and-scikit-learn/

[13]https://marcobonzanini.com/2015/05/17/mining-twitter-data-with-python-part-6-sentiment-analysis-basics/

[14]https://www.kaggle.com/ngyptr/python-nltk-sentiment-analysis/code

[15]https://www.figure-eight.com/data-for-everyone/

[16]https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/

[17]http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

[18]http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/

[19]https://www.nltk.org/book/ch06.html

[20]https://arxiv.org/abs/cs/0212032

[21]http://www.aclweb.org/anthology/P02-1053.pdf