Naman Dhameja and Kanika Arora

Neophytes

IBM HACKATHON 2020

TABLE OF CONTENTS

[Table of figures 2](#_Toc45663544)

[INTRODUCTION 2](#_Toc45663545)

[OVERVIEW 2](#_Toc45663546)

[PURPOSE 3](#_Toc45663547)

[LITERATURE SURVEY 3](#_Toc45663548)

[EXISTING PROBLEM 3](#_Toc45663549)

[PROPOSED SOLUTION 4](#_Toc45663550)

[THEORETICAL ANALYSIS 5](#_Toc45663551)

[BLOCK DIAGRAM 5](#_Toc45663552)

[SOFTWARE DESIGNING 5](#_Toc45663553)

[EXPERIMENTAL investigation 6](#_Toc45663554)

[FLOWCHART 9](#_Toc45663555)

[RESULT 12](#_Toc45663556)

[ADVANTAGES 12](#_Toc45663557)

[DISADVANTAGES 12](#_Toc45663558)

[APPLICATIONS 13](#_Toc45663559)

[CONCLUSION 13](#_Toc45663560)

[FUTURE SCOPE 13](#_Toc45663561)

[BIBLIOGRAPHY 14](#_Toc45663562)

[APPENDIX 15](#_Toc45663563)

[SOURCE CODE: 15](#_Toc45663564)

[1. Creating a module: 15](#_Toc45663565)

[2. Sentiment\_mod.py 18](#_Toc45663566)

[3. Twitter Data Gather: 20](#_Toc45663567)

[4. Graph Twitter Data: 21](#_Toc45663568)

[5. Node Red Flow 21](#_Toc45663569)

Table of figures

[Figure 1: Block Diagram 5](#_Toc44599283)

[Figure 2: Graph for positive topics 6](#_Toc44599284)

[Figure 3: Graph for negative topics 7](#_Toc44599285)

[Figure 4: Only positive tweets 7](#_Toc44599286)

[Figure 5: Only Negative Tweets 8](#_Toc44599287)

[Figure 6: Corona Virus Tweet Graph 8](#_Toc44599288)

[Figure 7: Module Creation Flowchart 9](#_Toc44599289)

[Figure 8: Twitter Data Gathering Flowchart 10](#_Toc44599290)

[Figure 9: Graphing Twitter Data Flowchart 11](#_Toc44599291)

INTRODUCTION

# OVERVIEW

With the widespread increase in the COVID19 pandemic, medical emergency has been declared. Due to this pandemic, the lifestyle of people has changed worldwide. Everyone is practicing social distancing, working from home, and are self-quarantined in their houses to control the spread of the virus. In this project, we are using live Twitter data to perform sentiment analysis on the tweets on the COVID19 virus. With the help of this project, we will get to know what impact this virus is having on the sentiments of people so that we can take all the necessary measures to help them stay motivated and positive. By deploying the Natural Language Tool Kit (NLTK) on it, we were able to extract and quantify the public sentiments over time. The output from this project can be utilized by the government to take action for the current COVID19 situation in the country keeping in mind the mental health of the people. The government will also be aware of the expectations of the countrymen related to the COVID crisis. This will help the government take necessary actions to keep in mind the sentiments of the people and what they expect the government to do for them.

# PURPOSE

The sentiment analysis of twitter data for the COVID19 tweets and some other terms associated with it will help the government to know the mindset of the people. This analysis will give an idea of what impact this COVID19 situation is having on the people, how they are tackling this situation, what problems they are facing in their day to day life. This analysis will also help them know what necessary measures people want the government to take to stop the spread of COVID19 Virus. As per the present situation, we all are stuck in our homes and not allowed to go anywhere. Due to which, there are people whose mental health is affected so this will also help the government to know the mental health of the people and what impact COVID19 virus is having on the people and also take necessary measures to solve these problems.

LITERATURE SURVEY

# EXISTING PROBLEM

The problems which the whole world is facing are new for everyone nobody knows how to deal with this situation and stop the spread of this deadly virus. All the countries are taking the best steps possible to handle this situation but still, they lack in some way or the other. We cannot blame government for this because this is new for them and they are doing their best they can for their countrymen. Still, they lack in fulfilling the needs of their countrymen because they are not aware of what the people are going through in their life due to the situation. The best solution for them to do is to know what people expect the government and then take necessary actions. It is not feasible for the government to talk to everyone in one to one interaction so they should come up with something where they can get a reference about the impact which this situation is having on the people.

# PROPOSED SOLUTION

With the help of Sentiment analysis of Twitter data, we will get real-time graphical analysis for the positive and negative reviews. Based on this analysis the government will be able to find out what impact this particular situation is having on the people and what changes do they expect the government should make in the guidelines to stop the spread of Corona Virus.

People use social media platforms to share their problems and concerns over the steps taken by the government. They share the daily issues they are facing due to lockdown and measures taken by the government. The biggest social media which provides us with data is twitter and as it is always public people prefer sharing their concerns on it much more than other platforms.

THEORETICAL ANALYSIS

# BLOCK DIAGRAM

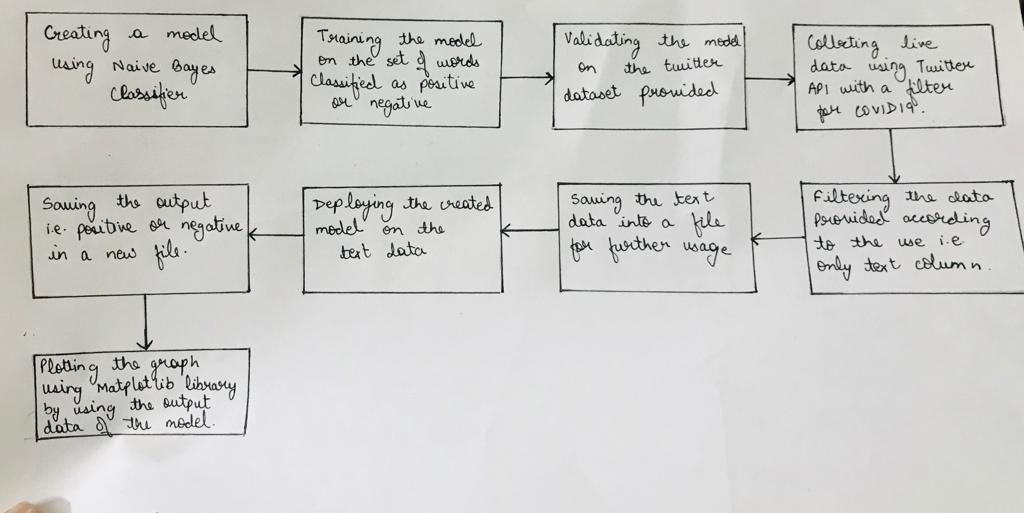


Figure : Block Diagram

# SOFTWARE DESIGNING

**Word Tokenization:** This process is used to split long textual data into words.

**Removing Stop Words**: The words which act as fillers and do not have meaning of their own are called Stop Words. They do not take part in the process of sentiment analysis. These words just use the space in our database. Therefore they need to be removed.

**Classification of Tweets**: After we have removed stop words the next thing which we did was to make two list and classify the data and store it either in positive list or negative list. Then we converted all the word to features.

**Algorithms Used in this project:**

* Naive Bayes Classifier
* Multinomial Naive Bayes Classifier
* Bernoulli Naive Bayes Classifier
* Linear SVC
* SGD Classifier

After processing of words was done every algorithm was stored in a pickle file.

We have another file named “sentiment\_mod.py”, where all the algorithms and their processed models are saved for quick use. This file is imported as a module while the data is gathered by the file “twitter\_data\_gather.py”. The data gathered by the “twitter\_data\_gather.py” file is classified using the sentiment function provided in the “sentiment\_mod.py” module. The function returns 2 parameters: sentiment\_value and confidence. The sentiment\_value is either pos for positive reviews and neg for negative reviews. Confidence level is the amount of confidence does the model have that the data is positive or negative.

If the confidence level of the data is more than 70% then sentiment\_value will be saved in the file named “twitter-out.txt”.

The file named “twitter\_graph.py” is used to create graph using the data present in the “twitter\_out.txt” file. The data is segregated into list and numeric values. Whenever the data is positive i.e. “pos” it adds +1to the value and when the data is negative i.e. “neg” it will subtract 1.

We have also used Matplotlib animation and created a graph which is refreshed at an interval of 1000 milli seconds.

EXPERIMENTAL investigation

As mentioned, we have collected live twitter data and analyzed it. There were few experiments we ran threw before collecting the actual data.

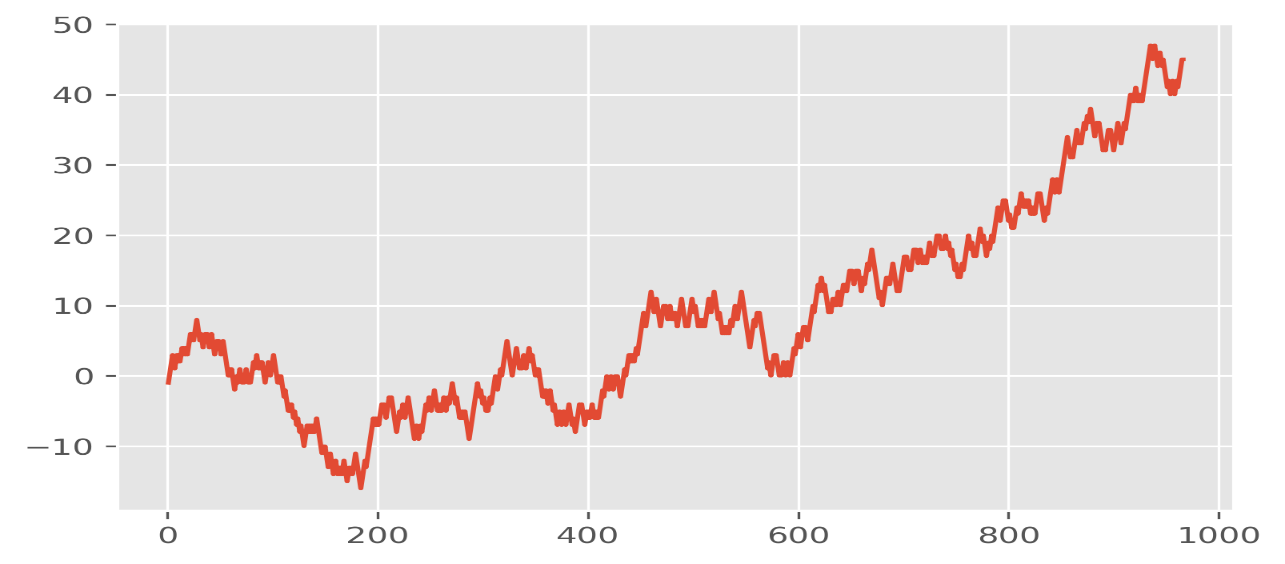
1. Searching tweets for topics (Happy, Joy, Festivals): 

Figure : Graph for positive topics

1. Searching tweets for topics (Gun, Drugs, weapons):

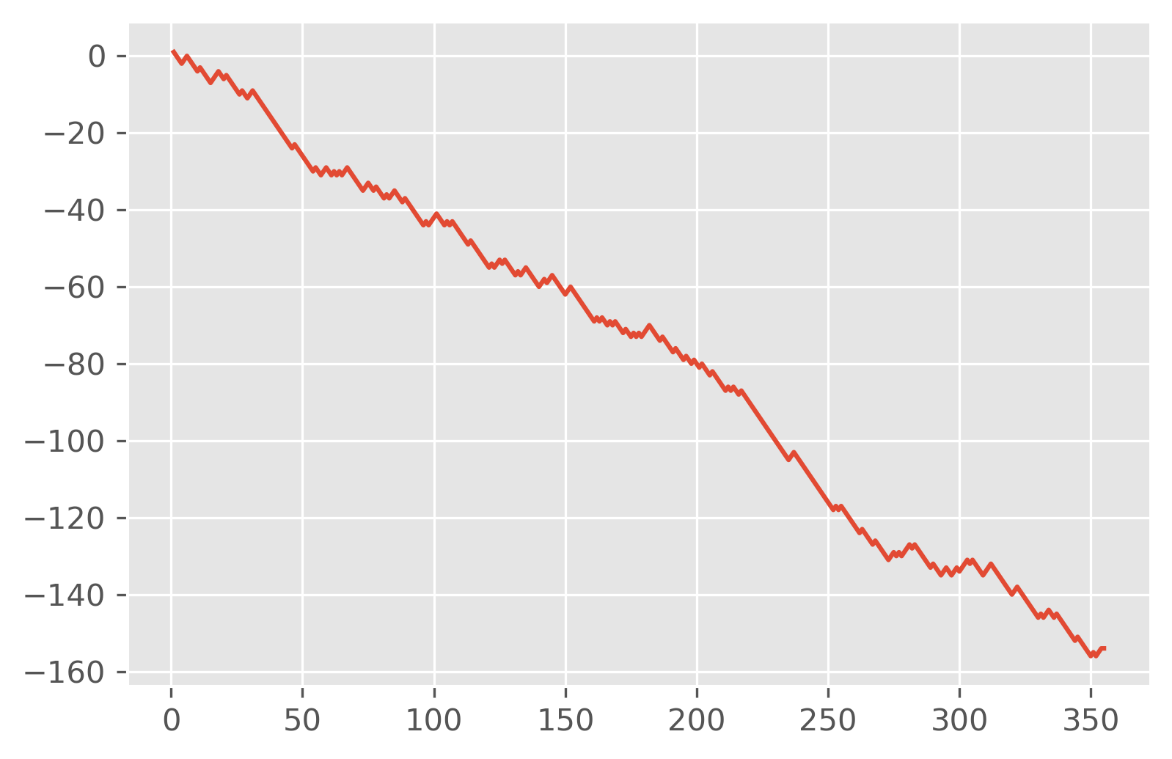


Figure : Graph for negative topics

1. Filtered the graph for only positive tweets:

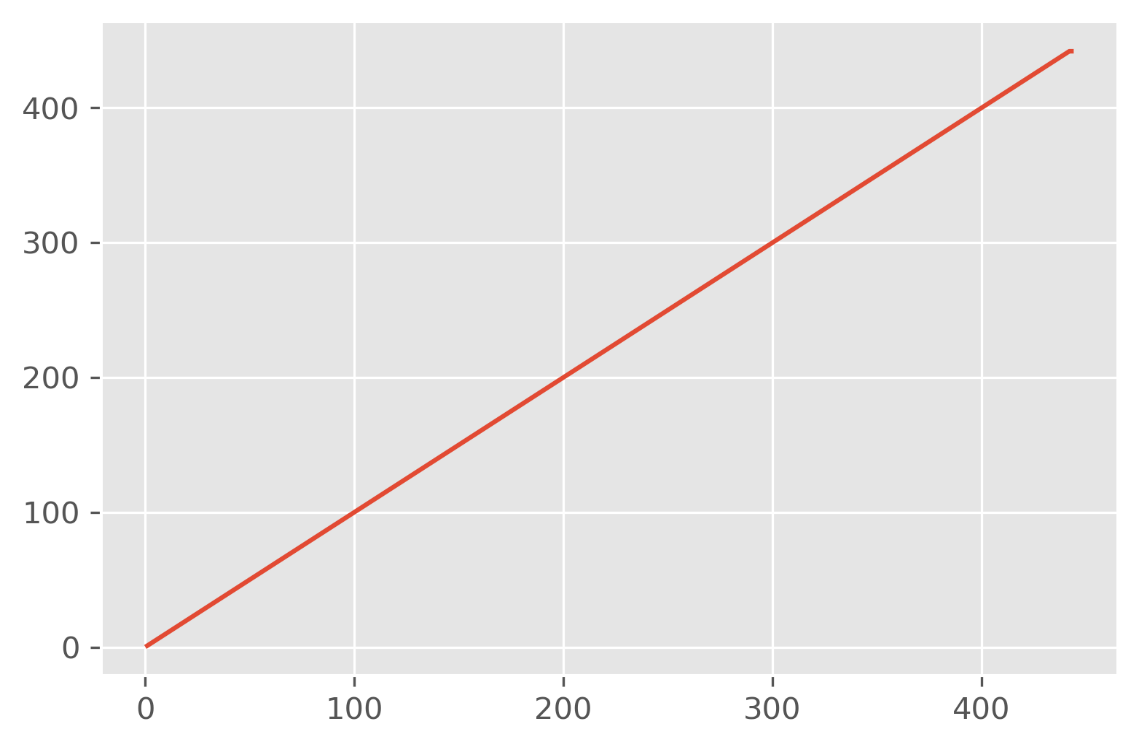


Figure : Only positive tweets

1. Filtered the graph for only negative tweets:

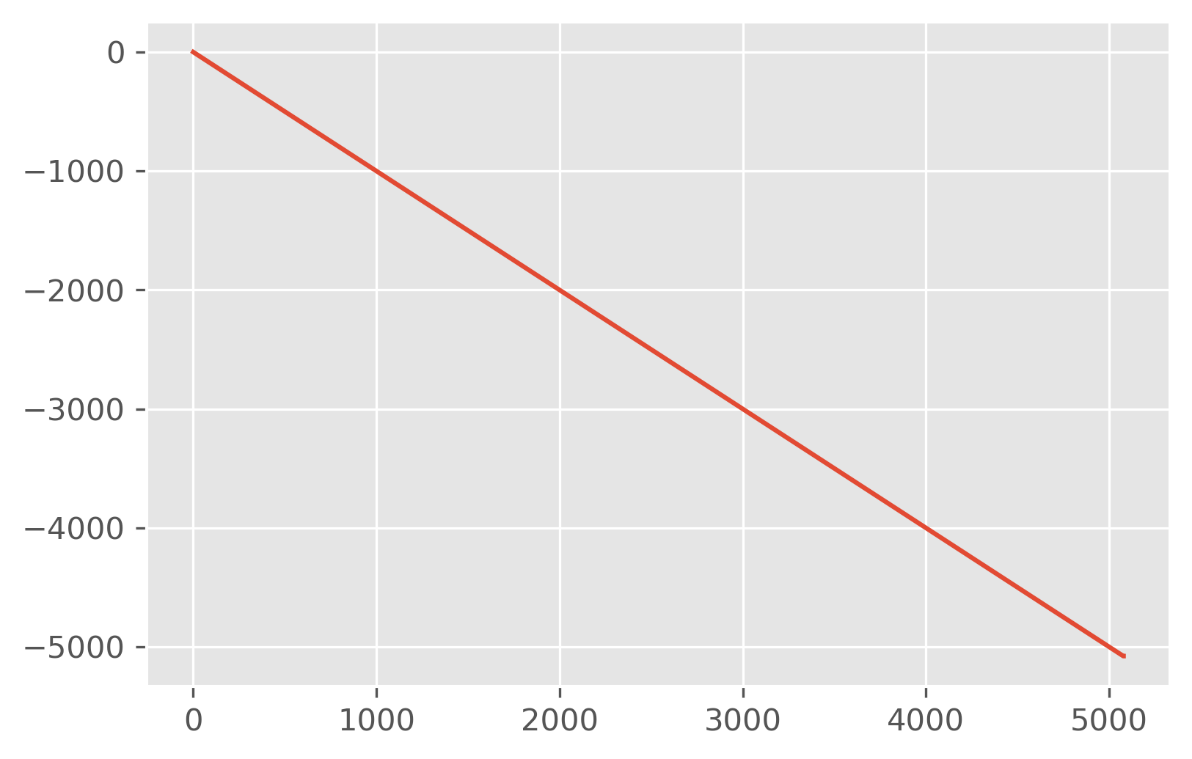


Figure : Only Negative Tweets

1. Corona Virus sentiment analysis:

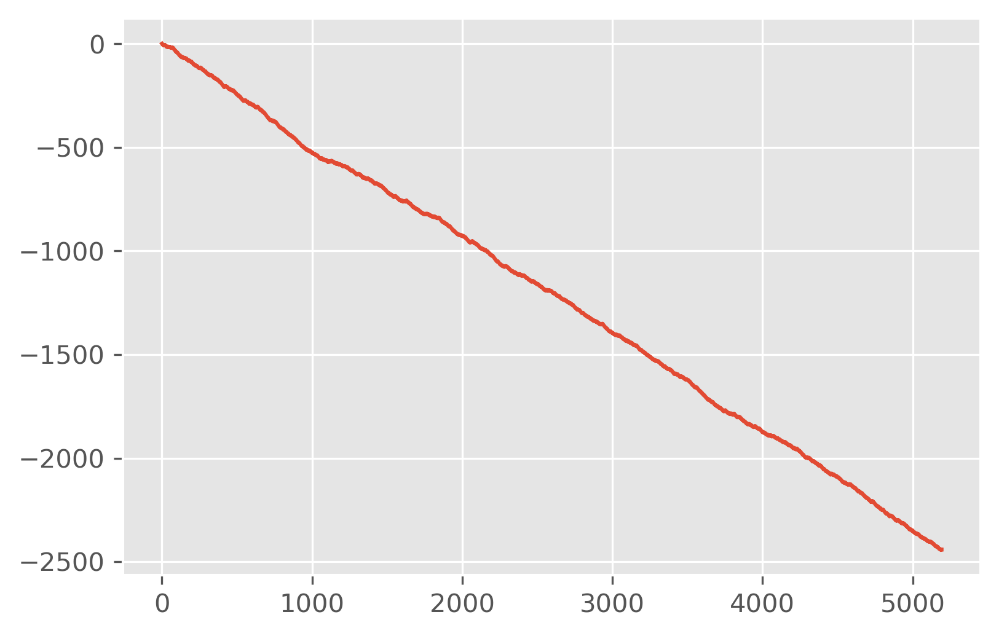


Figure : Corona Virus Tweet Graph

Total tweets: 5191

Positive tweets: 1375

Negative tweets: 3816

FLOWCHART

Creating a module:

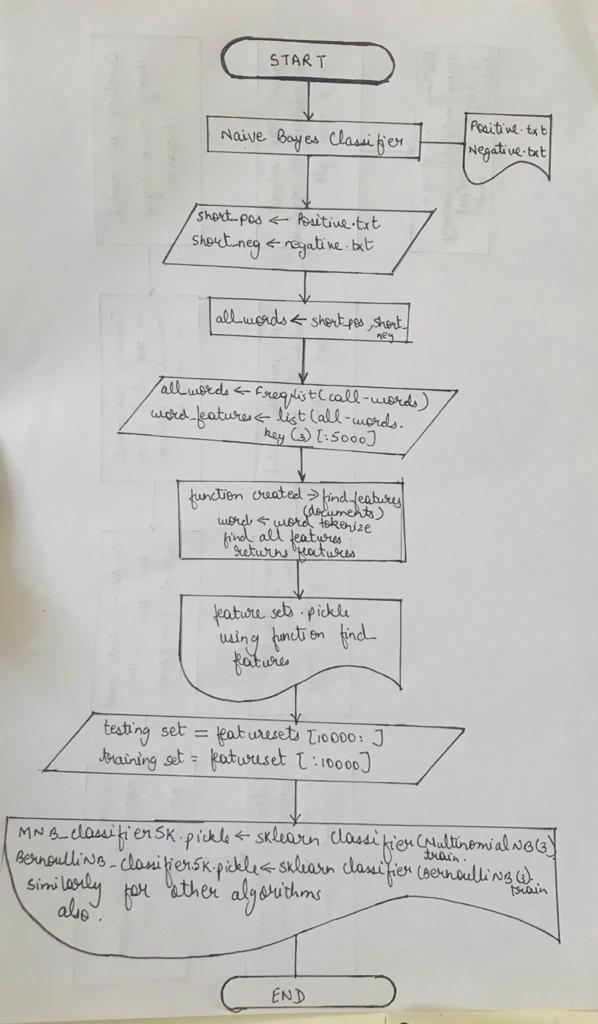


Figure : Module Creation Flowchart

Twitter Data Gathering:

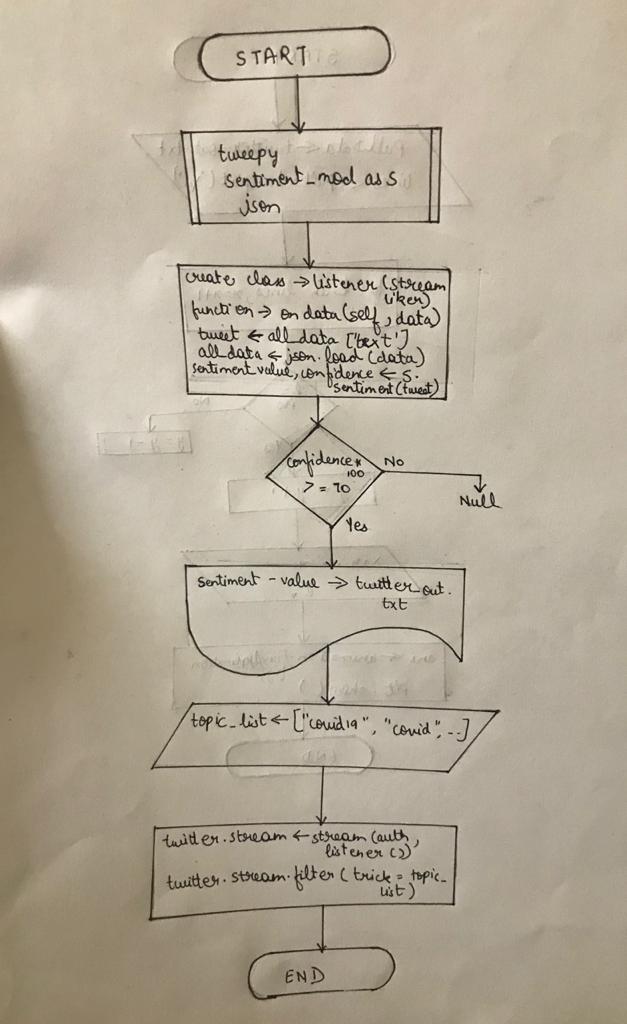


Figure : Twitter Data Gathering Flowchart

Graph Twitter data:

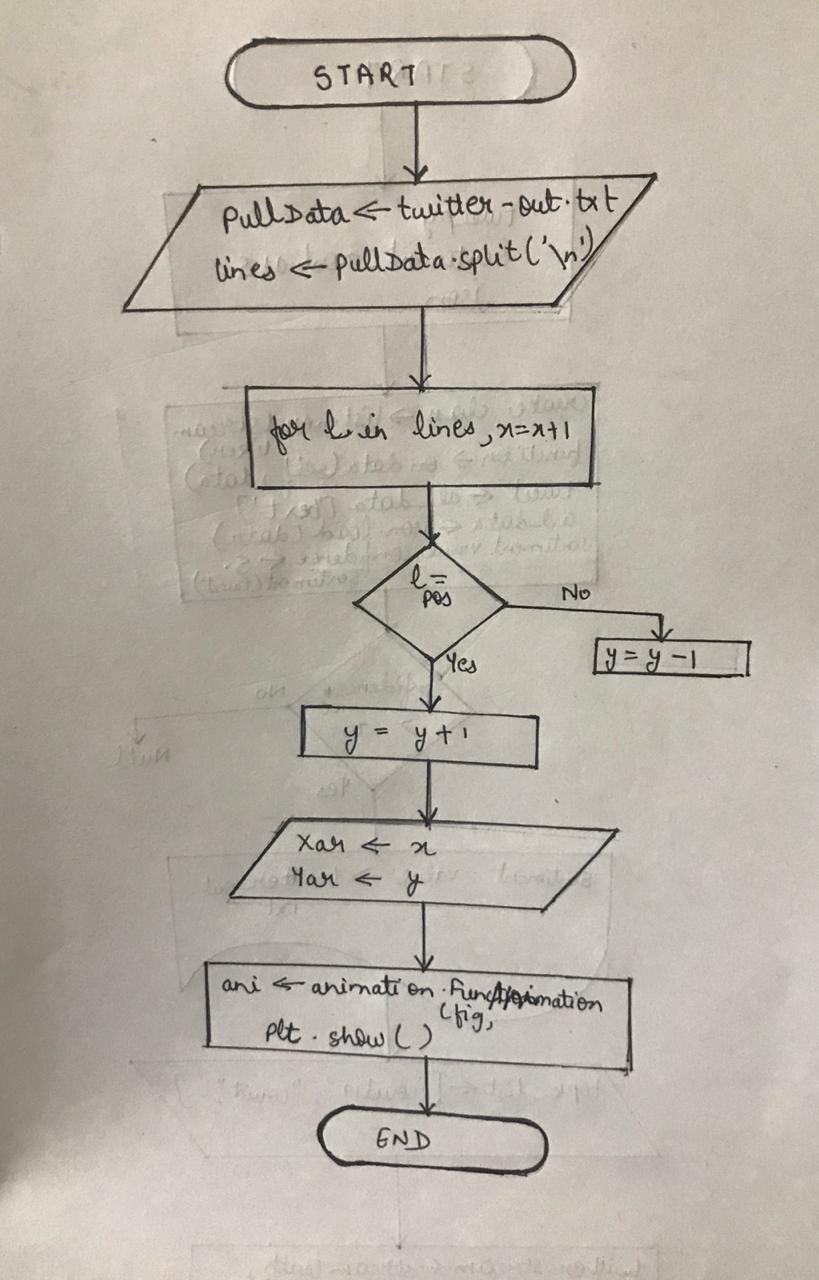


Figure : Graphing Twitter Data Flowchart

RESULT

With help of twitter data, we get to know the impact of COVID19 situation on the people. The later graph shows us the interest of people is going towards a positive side or the negative side. The positive side shows that the people are happy and liking the decisions by the government whereas the negative shows that the people are unhappy and not liking the decisions by the government. If this analysis is used by the government to take the action for the welfare of the people, they should also keep in mind the impact based on this analysis and then only take the necessary steps.

ADVANTAGES

* The result from sentiment analysis help government understand the conversations and discussions taking place about the COVID19 situation and helps them react and take action accordingly.
* The government can quickly identify any negative sentiments being expressed and tries to take necessary actions to solve those problems.
* They can publish guidelines according to the sentiments being expressed by their countrymen.

DISADVANTAGES

* Sentiment analysis do a really great job of analyzing text for opinion and attitude, but they're not perfect.
* Different countries and regions us different expressions and slang, even within the same language.
* It’s currently impossible to accurately assess posts that include ironic or sarcastic comments.

APPLICATIONS

* The sentiment analysis will help the government get general review for the impact of COVID19 on their countrymen.
* This will also help them to form new strategies based on the reviews they get from this sentiment analysis.

CONCLUSION

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far to detect the sentiments of corpus of texts very accurately because of the complexity in the English language and even more if we consider other languages. In this project we tried to show the basic way of classifying tweets into positive or negative category using Naive Bayes as baseline and how language models are related to the Naive Bayes and can produce better results. We could further improve our classifier ​by trying to extract more features from the tweets, trying different kinds of features, tuning the parameters of the naïve Bayes classifier, or trying another classifier all together.

FUTURE SCOPE

In this project we tried to show the basic way of classifying tweets into positive or negative category using machine learning libraries like Textblob,Twitter API and Tweepy. We could further improve our classifier by trying to extract more features from the tweets, trying different kinds of features, tuning the parameters of the naïve Bayes classifier, or trying another classifier all together. Sentiment analysis, has proven to be powerful tools in helping to understand customers’ perceptions related to products and services.

One of the greatest difficulties encountered was in determining the best approach for detecting sentiments in Twitter data because comparing various approaches is a highly challenging task.

BIBLIOGRAPHY

1. <https://developer.twitter.com>
2. [www.stackoverflow.com](http://www.stackoverflow.com)
3. <https://cognitiveclass.ai/courses>
4. <https://cloud.ibm.com/>
5. <https://nodered.org/docs/>
6. <https://cloud.ibm.com/docs>
7. [www.youtube.com](http://www.youtube.com)

APPENDIX

# SOURCE CODE:

## Creating a module:

import nltk

import random

#from nltk.corpus import movie\_reviews

from nltk.classify.scikitlearn import SklearnClassifier

import pickle

from sklearn.naive\_bayes import MultinomialNB, BernoulliNB

from sklearn.linear\_model import LogisticRegression, SGDClassifier

from sklearn.svm import SVC, LinearSVC, NuSVC

from nltk.classify import ClassifierI

from statistics import mode

from nltk.tokenize import word\_tokenize

class VoteClassifier(ClassifierI):

def \_\_init\_\_(self, \*classifiers):

self.\_classifiers = classifiers

def classify(self, features):

votes = []

for c in self.\_classifiers:

v = c.classify(features)

votes.append(v)

return mode(votes)

def confidence(self, features):

votes = []

for c in self.\_classifiers:

v = c.classify(features)

votes.append(v)

choice\_votes = votes.count(mode(votes))

conf = choice\_votes / len(votes)

return conf

short\_pos = open("short\_reviews/positive.txt","r").read()

short\_neg = open("short\_reviews/negative.txt","r").read()

# move this up here

all\_words = []

documents = []

# j is adject, r is adverb, and v is verb

#allowed\_word\_types = ["J","R","V"]

allowed\_word\_types = ["J"]

for p in short\_pos.split('\n'):

documents.append( (p, "pos") )

words = word\_tokenize(p)

pos = nltk.pos\_tag(words)

for w in pos:

if w[1][0] in allowed\_word\_types:

all\_words.append(w[0].lower())

for p in short\_neg.split('\n'):

documents.append( (p, "neg") )

words = word\_tokenize(p)

pos = nltk.pos\_tag(words)

for w in pos:

if w[1][0] in allowed\_word\_types:

all\_words.append(w[0].lower())

save\_documents = open("pickled\_algos/documents.pickle","wb")

pickle.dump(documents, save\_documents)

save\_documents.close()

all\_words = nltk.FreqDist(all\_words)

word\_features = list(all\_words.keys())[:5000]

save\_word\_features = open("pickled\_algos/word\_features5k.pickle","wb")

pickle.dump(word\_features, save\_word\_features)

save\_word\_features.close()

def find\_features(document):

words = word\_tokenize(document)

features = {}

for w in word\_features:

features[w] = (w in words)

return features

featuresets = [(find\_features(rev), category) for (rev, category) in documents]

random.shuffle(featuresets)

features = open("pickled\_algos/featuresets.pickle","wb")

pickle.dump(featuresets, features)

features.close()

print(len(featuresets))

testing\_set = featuresets[10000:]

training\_set = featuresets[:10000]

classifier = nltk.NaiveBayesClassifier.train(training\_set)

print("Original Naive Bayes Algo accuracy percent:", (nltk.classify.accuracy(classifier, testing\_set))\*100)

classifier.show\_most\_informative\_features(15)

###############

save\_classifier = open("pickled\_algos/originalnaivebayes5k.pickle","wb")

pickle.dump(classifier, save\_classifier)

save\_classifier.close()

MNB\_classifier = SklearnClassifier(MultinomialNB())

MNB\_classifier.train(training\_set)

print("MNB\_classifier accuracy percent:", (nltk.classify.accuracy(MNB\_classifier, testing\_set))\*100)

save\_classifier = open("pickled\_algos/MNB\_classifier5k.pickle","wb")

pickle.dump(MNB\_classifier, save\_classifier)

save\_classifier.close()

BernoulliNB\_classifier = SklearnClassifier(BernoulliNB())

BernoulliNB\_classifier.train(training\_set)

print("BernoulliNB\_classifier accuracy percent:", (nltk.classify.accuracy(BernoulliNB\_classifier, testing\_set))\*100)

save\_classifier = open("pickled\_algos/BernoulliNB\_classifier5k.pickle","wb")

pickle.dump(BernoulliNB\_classifier, save\_classifier)

save\_classifier.close()

LogisticRegression\_classifier = SklearnClassifier(LogisticRegression())

LogisticRegression\_classifier.train(training\_set)

print("LogisticRegression\_classifier accuracy percent:", (nltk.classify.accuracy(LogisticRegression\_classifier, testing\_set))\*100)

save\_classifier = open("pickled\_algos/LogisticRegression\_classifier5k.pickle","wb")

pickle.dump(LogisticRegression\_classifier, save\_classifier)

save\_classifier.close()

LinearSVC\_classifier = SklearnClassifier(LinearSVC())

LinearSVC\_classifier.train(training\_set)

print("LinearSVC\_classifier accuracy percent:", (nltk.classify.accuracy(LinearSVC\_classifier, testing\_set))\*100)

save\_classifier = open("pickled\_algos/LinearSVC\_classifier5k.pickle","wb")

pickle.dump(LinearSVC\_classifier, save\_classifier)

save\_classifier.close()

SGDC\_classifier = SklearnClassifier(SGDClassifier())

SGDC\_classifier.train(training\_set)

print("SGDClassifier accuracy percent:",nltk.classify.accuracy(SGDC\_classifier, testing\_set)\*100)

save\_classifier = open("pickled\_algos/SGDC\_classifier5k.pickle","wb")

pickle.dump(SGDC\_classifier, save\_classifier)

save\_classifier.close()

## Sentiment\_mod.py

import nltk

import random

#from nltk.corpus import movie\_reviews

from nltk.classify.scikitlearn import SklearnClassifier

import pickle

from sklearn.naive\_bayes import MultinomialNB, BernoulliNB

from sklearn.linear\_model import LogisticRegression, SGDClassifier

from sklearn.svm import SVC, LinearSVC, NuSVC

from nltk.classify import ClassifierI

from statistics import mode

from nltk.tokenize import word\_tokenize

class VoteClassifier(ClassifierI):

def \_\_init\_\_(self, \*classifiers):

self.\_classifiers = classifiers

def classify(self, features):

votes = []

for c in self.\_classifiers:

v = c.classify(features)

votes.append(v)

return mode(votes)

def confidence(self, features):

votes = []

for c in self.\_classifiers:

v = c.classify(features)

votes.append(v)

choice\_votes = votes.count(mode(votes))

conf = choice\_votes / len(votes)

return conf

documents\_f = open("pickled\_algos/documents.pickle", "rb")

documents = pickle.load(documents\_f)

documents\_f.close()

word\_features5k\_f = open("pickled\_algos/word\_features5k.pickle", "rb")

word\_features = pickle.load(word\_features5k\_f)

word\_features5k\_f.close()

def find\_features(document):

words = word\_tokenize(document)

features = {}

for w in word\_features:

features[w] = (w in words)

return features

featuresets\_f = open("pickled\_algos/featuresets.pickle", "rb")

featuresets = pickle.load(featuresets\_f)

featuresets\_f.close()

random.shuffle(featuresets)

print(len(featuresets))

testing\_set = featuresets[10000:]

training\_set = featuresets[:10000]

open\_file = open("pickled\_algos/originalnaivebayes5k.pickle", "rb")

classifier = pickle.load(open\_file)

open\_file.close()

open\_file = open("pickled\_algos/MNB\_classifier5k.pickle", "rb")

MNB\_classifier = pickle.load(open\_file)

open\_file.close()

open\_file = open("pickled\_algos/BernoulliNB\_classifier5k.pickle", "rb")

BernoulliNB\_classifier = pickle.load(open\_file)

open\_file.close()

open\_file = open("pickled\_algos/LogisticRegression\_classifier5k.pickle", "rb")

LogisticRegression\_classifier = pickle.load(open\_file)

open\_file.close()

open\_file = open("pickled\_algos/LinearSVC\_classifier5k.pickle", "rb")

LinearSVC\_classifier = pickle.load(open\_file)

open\_file.close()

open\_file = open("pickled\_algos/SGDC\_classifier5k.pickle", "rb")

SGDC\_classifier = pickle.load(open\_file)

open\_file.close()

voted\_classifier = VoteClassifier(

classifier,

LinearSVC\_classifier,

MNB\_classifier,

BernoulliNB\_classifier,

LogisticRegression\_classifier)

def sentiment(text):

feats = find\_features(text)

return voted\_classifier.classify(feats),voted\_classifier.confidence(feats)

## Twitter Data Gather:

from tweepy import Stream

from tweepy import OAuthHandler

from tweepy.streaming import StreamListener

import json

import sentiment\_mod as s

from twitterapi import \*

class listener(StreamListener):

def on\_data(self, data):

try:

all\_data = json.loads(data)

tweet = all\_data['text']

sentiment\_value, confidence = s.sentiment(tweet)

print(tweet, sentiment\_value, confidence)

if confidence\*100 >= 70:

output = open("twitter-out.txt","a")

output.write(sentiment\_value)

output.write('\n')

output.close()

return True

except:

return True

def on\_error(self, status):

print(status)

auth = OAuthHandler(ckey, csecret)

auth.set\_access\_token(atoken, asecret)

topic\_list=["Covid19","Coronavirus","StaySafeStayHome","INoLongerRemember","Lockdown","COVIDIOTS","WorkingFromHome","FlattenTheCurve","SARS-CoV-2","corona","stayhome","pandemia","quarantinelife","MyPandemicSurvival","government","politics","India","Modi","PMO"]

twitterStream = Stream(auth, listener())

twitterStream.filter(track=topic\_list)

## Graph Twitter Data:

import matplotlib.pyplot as plt

import matplotlib.animation as animation

from matplotlib import style

import time

style.use("ggplot")

fig = plt.figure()

ax1 = fig.add\_subplot(1,1,1)

def animate(i):

pullData = open("twitter-out.txt","r").read()

lines = pullData.split('\n')

xar = []

yar = []

x = 0

y = 0

for l in lines:

x += 1

if "pos" in l:

y += 1

elif "neg" in l:

y -= 1

xar.append(x)

yar.append(y)

ax1.clear()

ax1.plot(xar,yar)

ani = animation.FuncAnimation(fig, animate, interval=1000)

plt.savefig('plot.png', dpi=300, bbox\_inches='tight')

plt.show()

## Node Red Flow

Json code:-

[{"id":"61825093.7aef28","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"d1cad75.e957ca8","type":"twitter in","z":"61825093.7aef28","twitter":"","tags":"","user":"false","name":"","inputs":1,"x":70,"y":120,"wires":[["564a4c62.3ac56c"]]},{"id":"a20bdd45.edebf8","type":"switch","z":"61825093.7aef28","name":"Splitting the tweets","property":"sentiment.score","propertyType":"msg","rules":[{"t":"gt","v":"0","vt":"str"},{"t":"eq","v":"0","vt":"str"},{"t":"lt","v":"0","vt":"str"}],"checkall":"false","repair":false,"outputs":3,"x":690,"y":200,"wires":[["828d9262.eb1cc8"],["473f1411.1bfb94"],["fef7ccb5.464f78"]]},{"id":"828d9262.eb1cc8","type":"ui\_chart","z":"61825093.7aef28","name":"","group":"8ba8f93b.e91bb8","order":6,"width":"6","height":"4","label":"Positive Tweets","chartType":"line","legend":"true","xformat":"auto","interpolate":"linear","nodata":"Waiting for Data","dot":false,"ymin":"0","ymax":"5","removeOlder":"4","removeOlderPoints":"","removeOlderUnit":"604800","cutout":0,"useOneColor":true,"useUTC":false,"colors":["#1f77b4","#ff7f0e","#d62728","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":860,"y":120,"wires":[[]]},{"id":"473f1411.1bfb94","type":"ui\_chart","z":"61825093.7aef28","name":"","group":"8ba8f93b.e91bb8","order":6,"width":"6","height":"4","label":"Neutral Tweets","chartType":"line","legend":"true","xformat":"auto","interpolate":"linear","nodata":"Waiting for Data","dot":true,"ymin":"-5","ymax":"5","removeOlder":"4","removeOlderPoints":"","removeOlderUnit":"604800","cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#ff7f0e","#d62728","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":920,"y":200,"wires":[[]]},{"id":"fef7ccb5.464f78","type":"ui\_chart","z":"61825093.7aef28","name":"","group":"8ba8f93b.e91bb8","order":6,"width":"6","height":"4","label":"Negative Tweets","chartType":"line","legend":"true","xformat":"auto","interpolate":"linear","nodata":"Waiting for Data","dot":true,"ymin":"-5","ymax":"5","removeOlder":"4","removeOlderPoints":"","removeOlderUnit":"604800","cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#ff7f0e","#d62728","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":880,"y":280,"wires":[[]]},{"id":"11cb3411.25efb4","type":"debug","z":"61825093.7aef28","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":580,"y":80,"wires":[]},{"id":"564a4c62.3ac56c","type":"function","z":"61825093.7aef28","name":"Remove URLS and replace #","func":"var tweet = msg.payload;\nvar newtweet = tweet.replace(/#/g, \" Hash tag \");\n\nmsg.payload = newtweet.replace( /(([a-z]+:\\/\\/)?(([a-z0-9\\-]+\\.)+([a-z]{2}|biz|com|co|edu|gov|info|net|org|ly))(:[0-9]{1,5})?(\\/[a-z0-9\_\\-\\.~]+)\*(\\/([a-z0-9\_\\-\\.]\*)(\\?[a-z0-9+\_\\-\\.%=&amp;]\*)?)?(#[a-zA-Z0-9!$&'()\*+.=-\_~:@/?]\*)?)(\\s+|$)/gi, \"see short URL \" );\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":380,"y":180,"wires":[["11cb3411.25efb4","6dbd6b6e.875ba4"]]},{"id":"6dbd6b6e.875ba4","type":"sentiment","z":"61825093.7aef28","name":"","property":"payload","x":530,"y":300,"wires":[["a20bdd45.edebf8"]]},{"id":"8ba8f93b.e91bb8","type":"ui\_group","z":"","name":"Default","tab":"8fba4b4f.32018","order":1,"disp":true,"width":"6","collapse":false},{"id":"8fba4b4f.32018","type":"ui\_tab","z":"","name":"Corona\_MIne","icon":"dashboard","disabled":false,"hidden":false}]