COVID19 Sentiment Analysis Dashboard

https://c19dashboard.eu-gb.mybluemix.net/

Project Report

Team: Bruteforce

INTRODUCTION

Overview:

- A dynamic auto-updating dashboard, monitoring tweets made with hashtags such as #indialockdown, #Covid19, #IndiaFightsCOVID19.

- The web application has the facility to provide Real-time visualizations for sentiments of the general public towards multiple topics such as 'Lockdown', 'Covid-19', 'UnlockIndia' and will not be limited to above topics.

- The web application scrapes tweets every minute and updates the visualizations.

- Sentiments such as Positive, Neutral and Negative are provided. The tweets are further classified into following categories:
    - Joy
    - Fear
    - Sadness
    - Analytical
    - Anger
    - Confident

- Real-time Trends of Sentiments over time, Most used Hashtags and distribution of Sentiments across India are displayed using intuitive visualizations.

- Real-time Covid-19 cases count with line charts and twitter feed by relevant handles are also displayed.

- On-demand sentiment analysis and twitter timelines of twitter handles are also displayed.

- The web application is developed using the Django Web Framework.

Purpose:

We developed a Free and Open Source dashboard because currently there isn't any easy-to-use product which provides intuitive visualizations for public sentiments towards the Covid-19 pandemic, the Lockdown and Unlock decisions taken by the government.

This will help in bringing insights to sentiment distribution across India over the period of last four months. This will also bring insights to which decisions by the authorities or news lead to spikes in either positive or negative sentiment across the country over time.

This would help the authorities to gauge public opinion more easily leading to better management and would also help in changing the way decisions are taken or announcements are made.

LITERATURE SURVEY

Existing problem

Emerging pandemics show that humans are not infallible and communities need to be prepared. Coronavirus outbreak was first reported towards the end of 2019 and has now been declared a pandemic by the World Health Organization. In absence of concrete medical treatment people have to resort to techniques like social distancing and self quarantine.

These techniques work best when followed by the community as a whole. Thus it is crucial to monitor the general sentiment and morale of the community regarding the crisis.

Proposed solution

Thus to monitor the general sentiment and morale of the community regarding the crisis , we have come up with an effective solution in the form of a product. Our dashboard consists of relevant metrics related to sentiments expressed by people in tweets.

The Dashboard is a Django app hosted on IBM Cloud. Sentiment Analysis is performed with the help of modules and Services such as VADER Sentiment Analysis and IBM Watson Tone Analyser. We have used Chart.js and HighCharts to create intuitive graphs and visualization at the client-side.

Our dashboard provides a live feed of the current sentiment score of tweets related to the Covid-19 pandemic, the lockdown and decisions taken by the government. The sentiment score is calculated by the VADER Sentiment model which ranges between '-1' to '+1'. A tweet with a sentiment score less than '0' is considered as a negative tweet and vice-versa.

The dashboard also provides a State-Wise sentiment distribution visualization in the form of a Map Chart. It has the capability of showing the percentage of positive, negative and neutral tweets in any state.

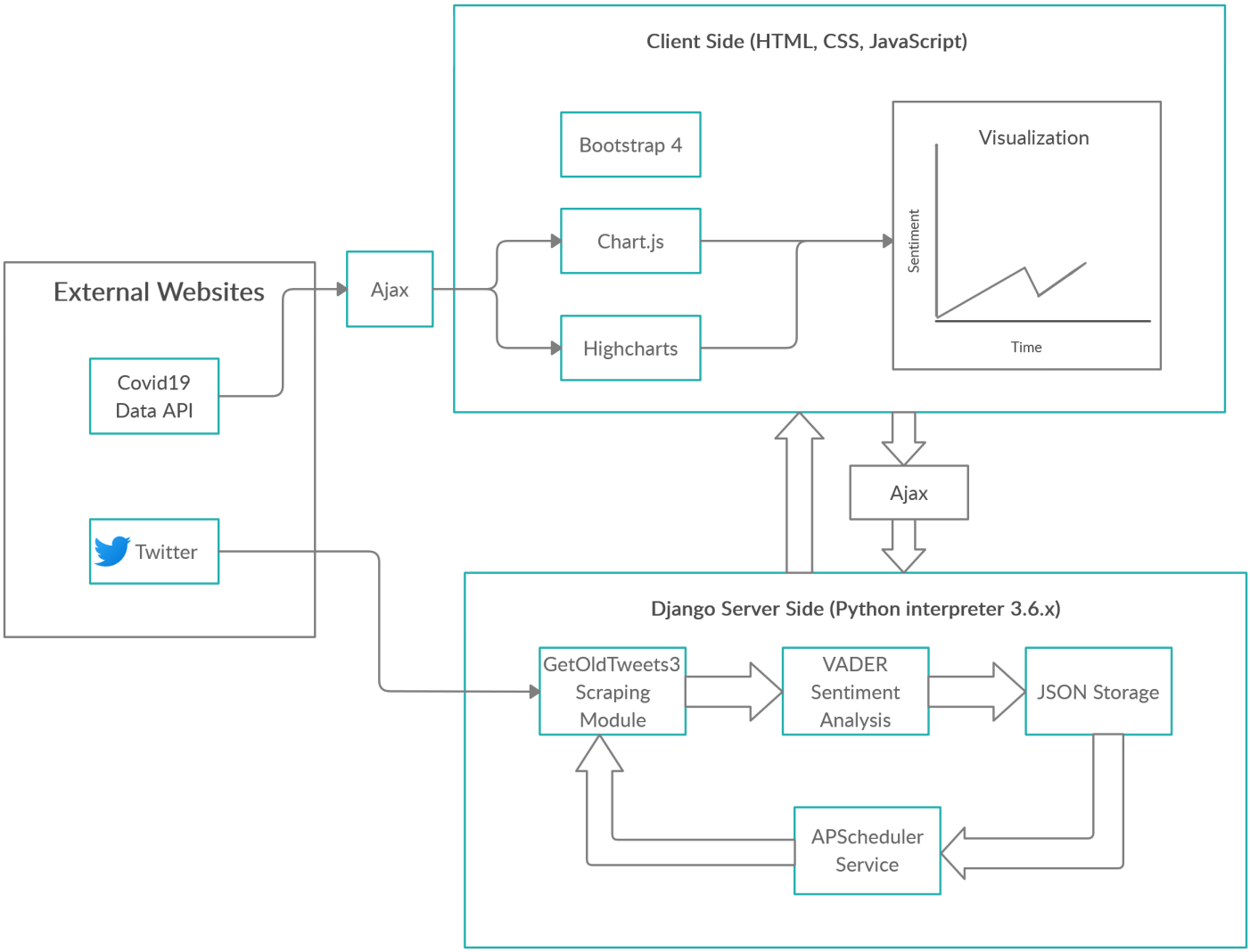
The dashboard also displays the historic trend of sentiment scores, and distribution of

sentiments in tweets which enables the user to identify which events led to the sudden spike in a sentiment across India. The user can also submit any twitter handle to analyse the distribution of the sentiments in it's twitter timeline.

It also displays the Real-time Covid-19 case count with State-Wise distribution in the form of a Map Chart.

THEORITICAL ANALYSIS

Block Diagram



Hardware / Software designing

The web application is made using Django as the backend due to the robustness of the web framework. We use APScheduler library to automate tasks like Scraping tweets and updating

JSON files required to render the graphs and charts.
Since sentiment analysis requires data from twitter the first step involves collecting data. For this purpose we use GetOldTweets3: a Python library for scraping tweets. We use numpy and pandas libraries to process data received from GetOldTweet3.
The next step consists of cleaning the data and make the data suitable for sentiment analysis. We use regex and emoji libraries to remove unwanted characters and convert emoji's and emoticons to their CLDR short names.
This leads us to Sentiment Analysis. To classify tweets into positive, negative and neutral we use the VADER-Sentiment-Analysis library. To classify tweets into Joy, Fear, Sadness, Analytical, Anger, Confident and Tentative we use IBM Watson Tone Analyser.
For a responsive PC and Mobile user interface, we used Bootstrap: an open-source CSS framework.
The next step is to display graphs and charts to the end user. This consists of making JSON objects to be sent to the respective javascript chart libraries.
We make JSON objects consisting of desired data with the json python package. We then use these JSON files to render intuitive charts with chart.js and HighCharts.
The following steps were followed :-

1) Data Collection

We made use of GetOldTweets3 library to scrape tweets from twitter. The result was received in form of .csv file containing the tweet data.

Installation:

```
pip install GetOldTweets3
import GetOldTweets3 as got
```

Usage:

```
def scrape():
    """
    Fetches 500 tweets with predefined search query between yesterday and today.
    Returns Pandas DataFrame.
    """
    tweetCriteria = got.manager.TweetCriteria().setQuerySearch(searchQuery) \
            .setSince((datetime.datetime.now(tz=ist) - datetime.timedelta(days=1)).strftime("%Y-%m-%d")) \
```

```
.setUntil((datetime.datetime.now(tz=ist)).strftime("%Y-%m-%d")) \
        .setLang('en') \
        .setEmoji('unicode') \
        .setMaxTweets(500)

    tweet = got.manager.TweetManager.getTweets(tweetCriteria)
    text_tweets = [[tw.username,
                    tw.text,
                    tw.date,
                    tw.retweets,
                    tw.favorites,
                    tw.mentions,
                    tw.hashtags] for tw in tweet]

    tweet_df = pd.DataFrame(text_tweets,
                            columns=['user', 'text', 'date',
'favorites', 'retweets', 'mentions', 'hashtags'])
    return tweet_df
```

2) Data Preprocessing

After that we performed data preprocessing using regex and emoji python library on the CSV files and added column: "clean_text" thus completing all Data Preprocessing tasks.

Installation:

```
pip install emoji
import emoji as emoji
import re
```

Usage:

```
def data_pre_processing(data_frame_arg):
  data_frame_arg["clean_text"] = data_frame_arg["text"].to_numpy()
  def removeURLs(str):
    return re.sub(r'https?://\S+', ' ', str)

  data_frame_arg["clean_text"] = data_frame_arg["clean_text"].apply(lambda tweet:
removeURLs(tweet))
```

```
7
8     def proc(text):
9         text = emoji.demojize(text, use_aliases=True, delimiters=(", ")).replace('_', ' ').replace('&',
    'and') \
10            .replace('.', '. ').replace(',', ', ').replace('#', '# ').replace('?', '? ').replace('!', '! ').replace('&',
    'and')
11        return re.sub(r'\s+', ' ', text)
12
13    data_frame_arg["clean_text"] = data_frame_arg["clean_text"].apply(lambda tweet:
    proc(tweet))
```

3) Sentiment and Tone Analysis

For sentiment analysis we made use of VADER-Sentiment Analysis python library. The model takes in cleaned data the data preprocessing step. The model makes use lexical analysis to assign a sentiment score to each tweet. This sentiment can be interpreted as follows:-

Sentiment Score	Sentiment
x > 0	Positive
x = 0	Neutral
x < 0	Negative

Tone Analysis was done using IBM Watson Tone Analyser API. The tone analyser classifies tweet in following classes :

- Joy 😄
- Fear 😨
- Sadness 😔
- Analytical
- Anger 😠
- Confident 😀
- Tentative

Installation for VADER:

```
1 pip install vaderSentiment
2 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

Usage for VADER:

```
def get_score_df(data_frame_arg):
    """
    Performs VADER Sentiment Analysis.
    adds 'pos', 'neg', 'neu', 'compound', 'result' colmns to
data_frame_arg passed as argument.
    """
    sia = SentimentIntensityAnalyzer()
    scores_ = data_frame_arg["text"].apply(lambda tweet:
sia.polarity_scores(tweet))
    scores_df_ret = pd.DataFrame(list(scores_))
    scores_df_ret['result'] = scores_df_ret['compound'].apply(
        lambda res: 'neutral' if res == 0 else ('positive' if res > 0
else 'negative'))
    for col in scores_df_ret:
        data_frame_arg[col] = scores_df_ret[col].to_numpy()
```

Installation for Watson Tone Analyser:

```
pip install "ibm-watson>=4.5.0"
from ibm_watson import ToneAnalyzerV3, ApiException
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

Usage for Watson Tone Analyser:

```
def get_tones(data_frame_arg):
  row_list = []
  def get_single_tone(text):
    tone_analysis = tone_analyzer.tone({'text': text},
content_type='application/json', sentences=False).get_result()
    dict1 = {'analytical': 0, 'anger': 0,'confident': 0, 'fear': 0,
'joy': 0, 'sadness': 0, 'tentative': 0}
    for tone in tone_analysis['document_tone']['tones']:
      dict1.update({tone['tone_id'] : tone['score']})
    row_list.append(dict1)
  data_frame_arg['clean_text'].apply(lambda tweet:
get_single_tone(tweet))
  df = pd.DataFrame(row_list)
  for col in df:
    data_frame_arg[col] = df[col].to_numpy()
```

4) Building Web Application

The web application was built using Django backend and HTML, Javascript frontend. We made use of the Chart.js javascript framework to display various charts and illustrations.
The data obtained in previous step was processed and the data for individual charts was generated. This data was stored in JSON files for easy retrieval.
The JSON files can be accessed by javascript through ajax:

```
$.ajax({
  method: "GET",
  url: 'https://api.rootnet.in/covid19-in/stats/history',
  success: function(data) {
    let i;
    for(i = 0; i<data.data.length; i++){
      labels1.push(data.data[i].day);
      values1.push(data.data[i].summary.total);
      if (i >= data.data.length -30) {
        labels.push(data.data[i-1].day);
        values.push(data.data[i].summary.total
  data.data[i-1].summary.total);
      }
    }
    LineCases(labels, values);
    LineCases1(labels1, values1);
  },
  error: function(error_data) {
    console.log(error_data);
  }
})
```

5) Graphs and Visualizations

The web app is divided in four subpages. Each of the sub-page contains different graphs. There are about 22 charts spread across these four pages. The pages are

a) General:

The general section consist of :

1. **Top 10 Hastags:** A Bar graph containing information about top 10 used hastags related to covid 19.
2. **Sentiment:** A Pie chart containing information about positive, negative and neutral sentiments.
3. **Sentiment vs Time:** A line plot containing information about the count of tweets with postive, negative and neutral sentiments every day form 23rd march.
4. **Mean Sentiment Vs Time**: A line plot containing information about the the average of sentiment scores of 500 tweets. The Sentiment score ranges between [-1 to +1], where the sign of the sentiment score tells whether the tweet has a positive or a negative sentiment and the magnitude of the sentiment score tells the intenstity of the sentiment.
5. **Positive Sentiment Vs Time**: A line plot containing information about the the average of postive sentiment scores of 500 tweets.
6. **Negative Sentiment Vs Time**: A line plot containing information about the the average of negative sentiment scores of 500 tweets.
7. **Sentiment Vs Time: Live Feed:** A line plot containing live feed of average sentiment of tweets which is updated every 1 minute.

Graphs 1 to 6 are updated once every day at 1:00 am IST. Graph 7 is updates every minute.

b) Sentiment Graphs:

This section consist of classification of tweets based on the tone of tweet into following categories:

a) Analytical
b) Tentative
c) Confident
d) Joy
e) Sadness
f) Anger
g) Fear

The classifications were made by IBM Watson Tone analyser API. The data classified by IBM Watson Tone Analyser was cleaned and fed to BERT model as training data.

c) Covid 19 Updates

This page consist of live auto-updating information about COVID19 in INDIA. This was done by making use of API found at: https://github.com/amodm/api-covid19-in
The information includes:
1) Count of total cases
2) Count of patients recovered

3) Count of Active cases
4) Count of total deaths
5) Graph of rise in number of cases every day
6) Graph of total cumulative cases
7) India Map with state-wise covid-19 cases.

The graphs are updated once every day.

d) Tweet Analysis (By twitter username)

This page consists of dynamically updating graph of positive, negative and neutral sentiments of twitter user whose username is specified by the user.

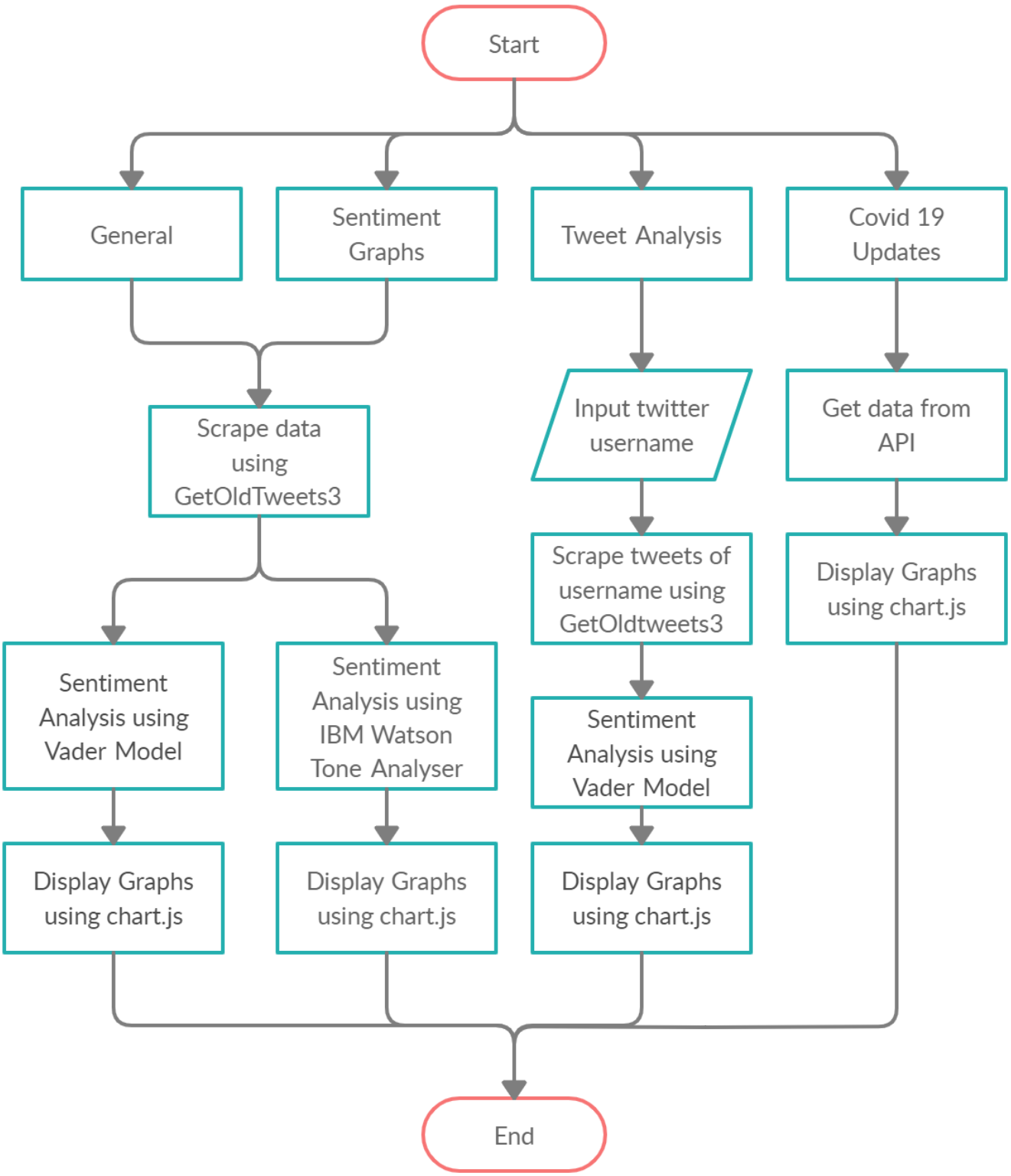
Input : Twitter Username (eg. WHO)
output : Graph containing frequency of positive, negative and nuetral sentiment of the user

The graph updates every time user submits a new username

EXPERIMENTAL INVESTIGATIONS

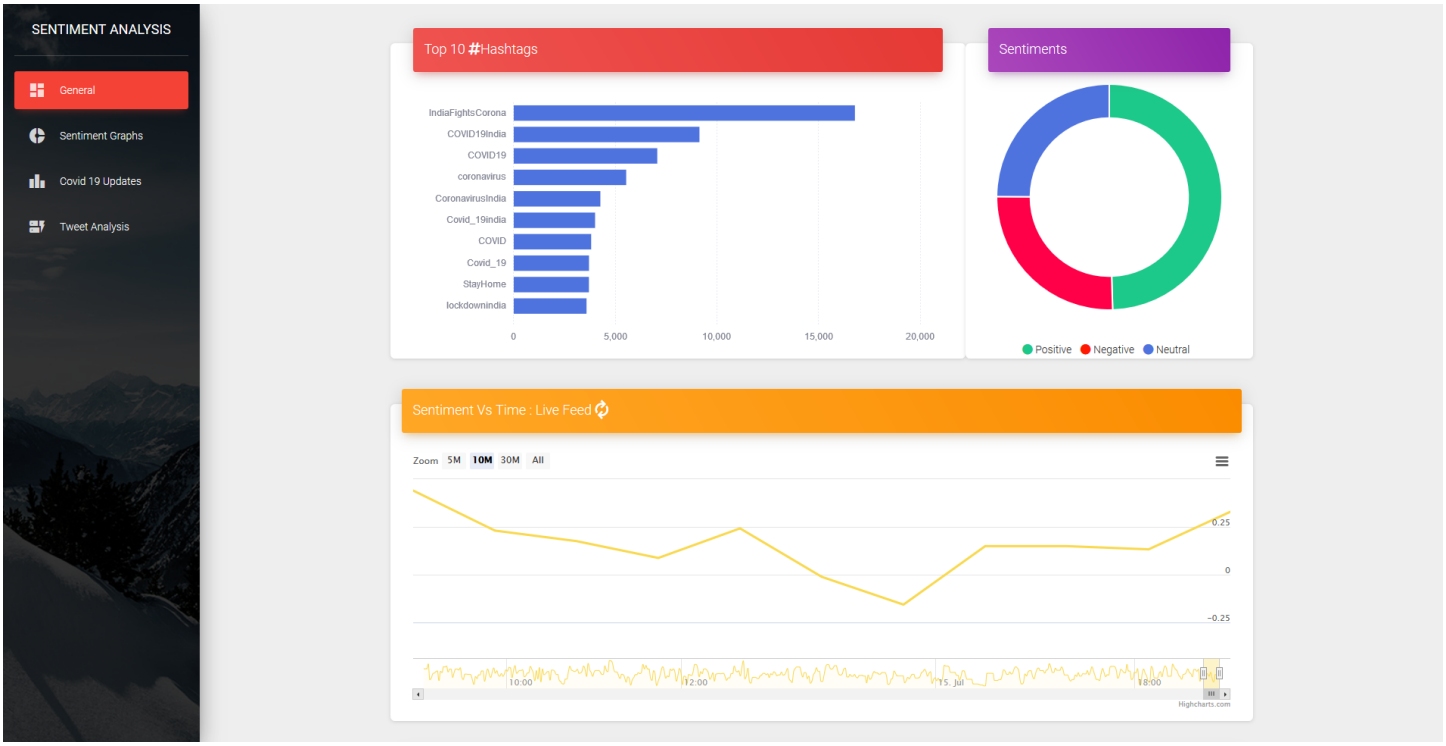
We obseverved that the number of positive tweets decreased on the days Lockdown was extended / announced.
We also obseverved that number of positive tweets increased after Unlock decisions were announced.

FLOWCHART


Start
General
Sentiment Graphs
Tweet Analysis
Covid 19 Updates
Scrape data using GetOldTweets3
Input twitter username
Get data from API
Sentiment Analysis using Vader Model
Sentiment Analysis using IBM Watson Tone Analyser
Scrape tweets of username using GetOldtweets3
Display Graphs using chart.js
Sentiment Analysis using Vader Model
Display Graphs using chart.js
Display Graphs using chart.js
Display Graphs using chart.js
End

RESULT

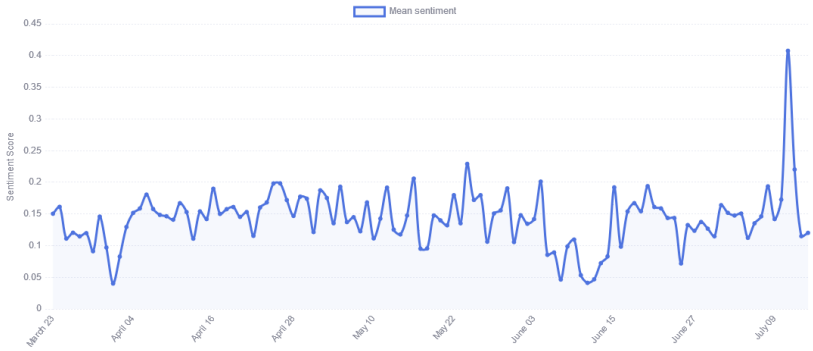
User Interface:



SENTIMENT ANALYSIS

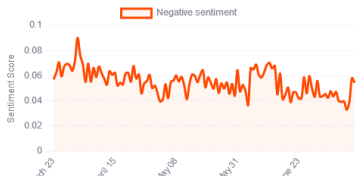
- General
- Sentiment Graphs
- Covid 19 Updates
- Tweet Analysis

Mean Sentiment Vs Time : Mean Sentiment Score of 500 Tweets Vs Time



Positive Sentiment Vs Time : Mean Postive Sentiment Score of 500 Tweets Vs Time

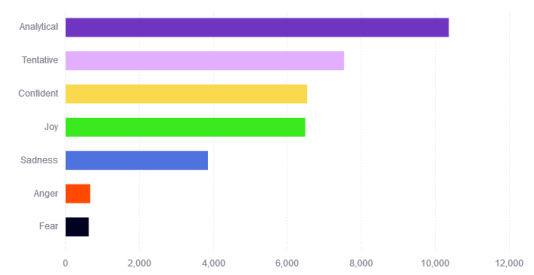
Negative Sentiment Vs Time : Mean Negative Sentiment Score of 500 Tweets Vs Time



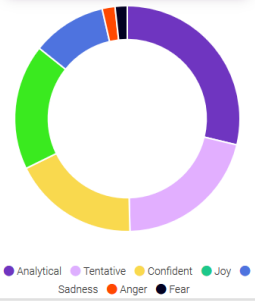
SENTIMENT ANALYSIS

- General
- Sentiment Graphs
- Covid 19 Updates
- Tweet Analysis

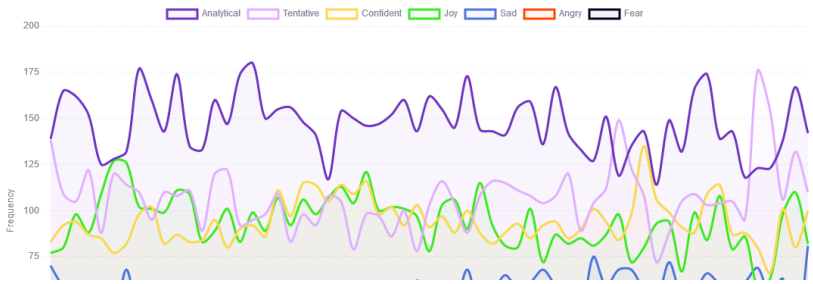
Number of Tweets by Sentiment

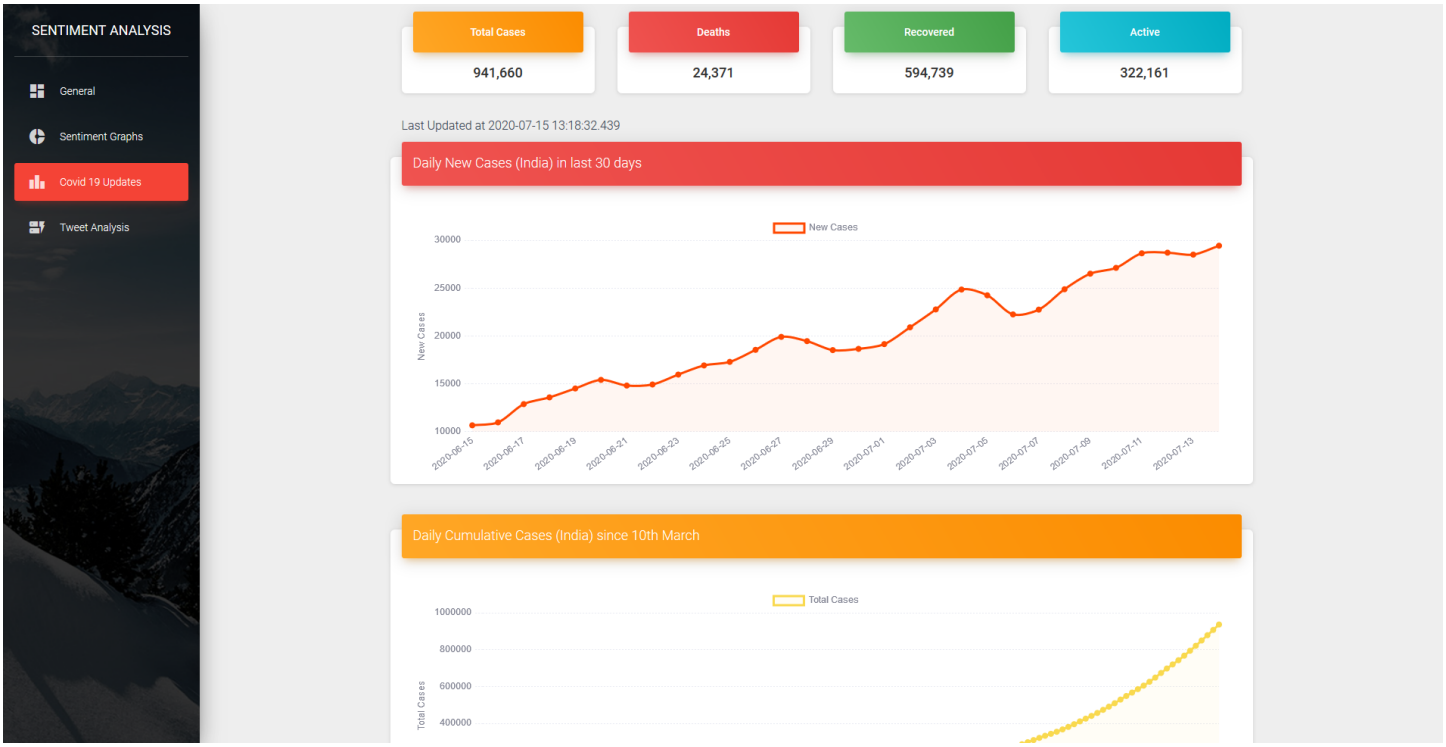


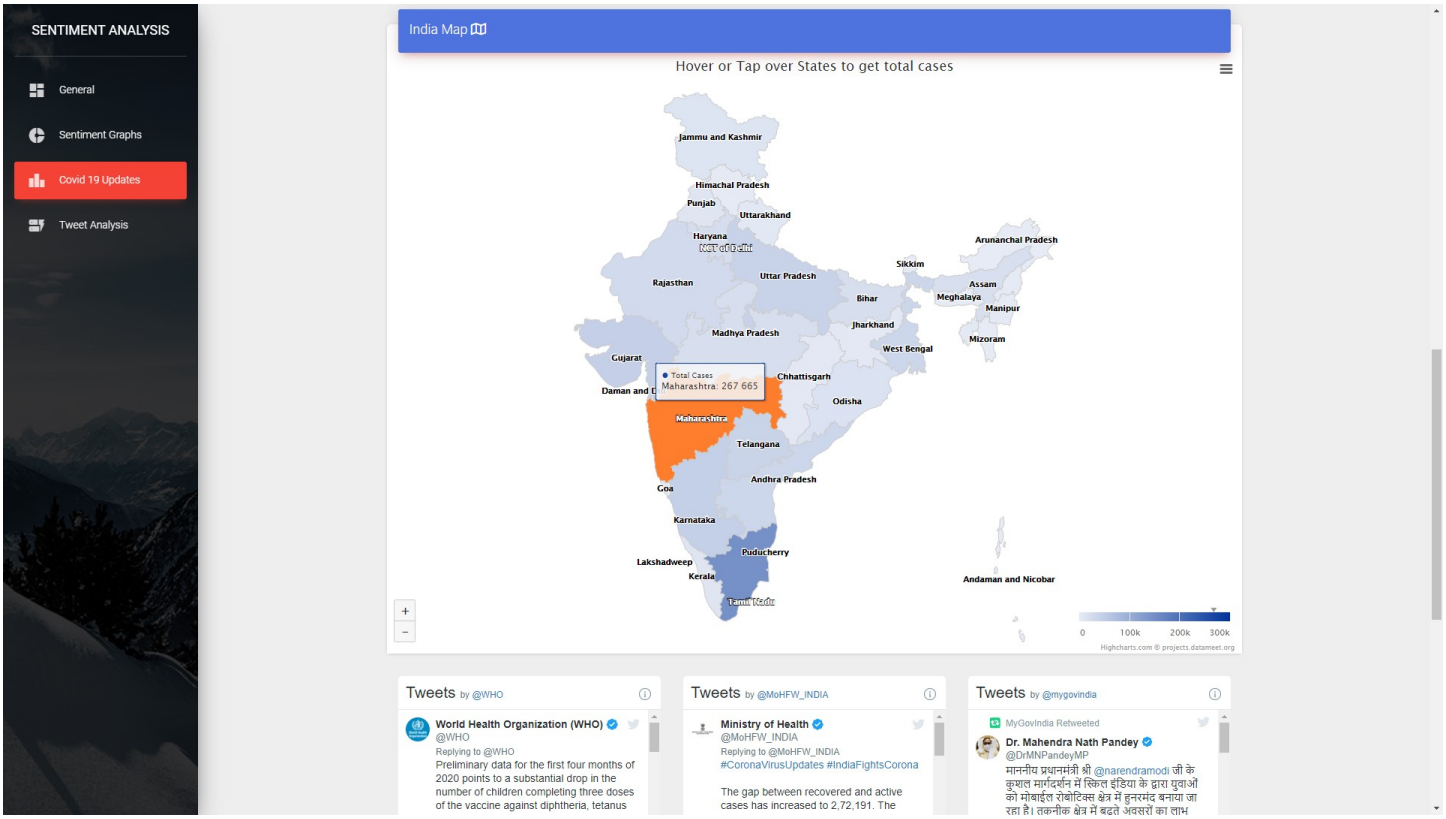
Sentiments

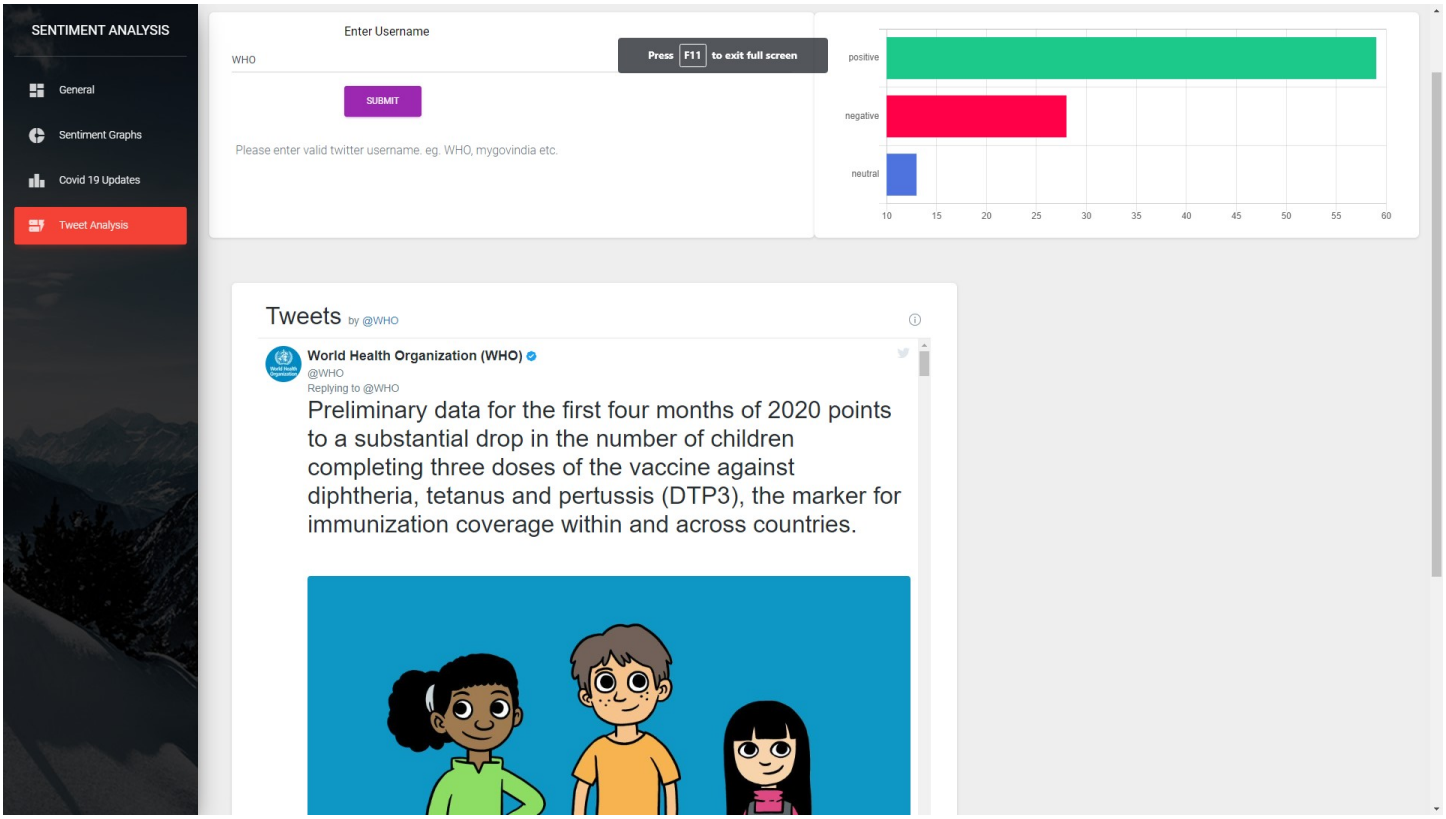


Frequency of Tweets with Specific Emotion per 500 tweets Vs Time




SENTIMENT ANALYSIS
General
Sentiment Graphs
Covid 19 Updates
Tweet Analysis
Total Cases
941,660
Deaths
24,371
Recovered
594,739
Active
322,161
Last Updated at 2020-07-15 13:18:32.439
Daily New Cases (India) in last 30 days
New Cases
30000
25000
20000
15000
10000
New Cases
2020-06-15
2020-06-17
2020-06-19
2020-06-21
2020-06-23
2020-06-25
2020-06-27
2020-06-29
2020-07-01
2020-07-03
2020-07-05
2020-07-07
2020-07-09
2020-07-11
2020-07-13
Daily Cumulative Cases (India) since 10th March
Total Cases
1000000
800000
600000
400000
Total Cases


SENTIMENT ANALYSIS
General
Sentiment Graphs
Covid 19 Updates
Tweet Analysis
India Map
Hover or Tap over States to get total cases
Jammu and Kashmir
Himachal Pradesh
Punjab
Uttarakhand
Haryana
NCT of Delhi
Rajasthan
Uttar Pradesh
Bihar
Sikkim
Arunanchal Pradesh
Assam
Meghalaya
Manipur
Mizoram
Jharkhand
West Bengal
Madhya Pradesh
Gujarat
Daman and D
Total Cases
Maharashtra: 267 665
Chhattisgarh
Odisha
Maharashtra
Telangana
Andhra Pradesh
Goa
Karnataka
Puducherry
Lakshadweep
Kerala
Tamil Nadu
Andaman and Nicobar
0
100k
200k
300k
Highcharts.com © projects.datameet.org
Tweets by @WHO
World Health Organization (WHO)
@WHO
Replying to @WHO
Preliminary data for the first four months of 2020 points to a substantial drop in the number of children completing three doses of the vaccine against diphtheria, tetanus
Tweets by @MoHFW_INDIA
Ministry of Health
@MoHFW_INDIA
Replying to @MoHFW_INDIA
#CoronaVirusUpdates #IndiaFightsCorona
The gap between recovered and active cases has increased to 2,72,191. The
Tweets by @mygovindia
MyGovIndia Retweeted
Dr. Mahendra Nath Pandey
@DrMNPandeyMP
माननीय प्रधानमंत्री श्री @narendramodi जी के कुशल मार्गदर्शन में स्किल इंडिया के द्वारा युवाओं को मोबाईल रोबोटिक्स क्षेत्र में हुनरमंद बनाया जा रहा है। तकनीक क्षेत्र में बढ़ते अवसरों का लाभ



ADVANTAGES

This dashboard has enabled many users to easily understand how people think about the pandemic and the decisions taken by the government.
As the dashboard is a web application made using bootstrap, it is visible on any device ranging from PC's, tablets to mobiles. Being a web application, it also has a huge advantage of being cross platform, thus reducing time for development and pushing updates.

DISADVANTAGES

As it is a web application it is dependant on an internet connection. In the event IBM Cloud Services go down or offline we currently don't have any alternatives / backup in place.

APPLICATIONS

The Covid-19 Sentiment Analysis Dashboard will be used to monitor the Real-time general sentiment and morale of the community regarding the crisis. It will help in finding the sentiments of people towards trending topics related to the covid-19 pandemic.
Government offices would be able to use this dashboard to help in changing the way decisions are taken or announcements are made.
Users would be able to perform sentiment analysis for tweets from any twitter handle within seconds.
State officials will easily be able to gauge the public sentiment towards the decisions taken.

CONCLUSION

Thus we have successfully developed a robust visualization dashboard depicting the sentiment of people towards the covid-19 pandemic and decisions taken by the government using the Django web framework.
In the process we learnt how to deploy web applications to the cloud and learnt about the many capabilities of IBM Cloud Services.

FUTURE SCOPE

This dashboard currently includes sentiment distribution across states in India, it could be further expanded to include sentiment distribution across cities across any state in any country.
This dashboard is currently a web application which can also be developed as an Android or iOS

application using the same back-end.
This dashboard can also be used in other projects to increase functionality.
Personalization features such as recommended topics of interest and twitter handles can be added.
Currently the dashboard includes hashtags related to the Covid-19 pandemic which can be easily expanded to more topics and sectors such as the finance sector, sentiments related to a particular stock / share or any stock market index such as the NIFTY 50.

BIBILIOGRAPHY

Python Software Foundation. Python Language Reference, version 3.6. Available at http://www.python.org
The NumPy project. Available at https://numpy.org
The Pandas project. Available at https://pandas.pydata.org/
GetOldTweets3. Available at https://github.com/Mottl/GetOldTweets3
VADER-Sentiment-Analysis. Available at https://github.com/cjhutto/vaderSentiment
Watson Tone Analyser. Available at https://www.ibm.com/watson/services/tone-analyzer/
Django, the web framework for perfectionists with deadlines. Available at https://www.djangoproject.com
Bootstrap, the world's most popular front-end open source toolkit. Available at https://getbootstrap.com/

APPENDIX

Source code

Github repository link:
https://github.com/SmartPracticeschool/SBSPS-Challenge-1385-COVID-19-Sentiment-Analysis-Dashboard

Live Website link: https://c19dashboard.eu-gb.mybluemix.net/

Youtube Video Presentation link:
https://www.youtube.com/watch?v=YXsMKCpl7xA