IBM Hack Challenge 2020

Project Report

Sentiment Analysis of Covid-19 Tweets

Visualisation Dashboard

Application ID: SPS_CH_APL_20200005048

Project ID: SPS_PRO_746

Team Name: Code-inators

Team Members

Ritika Bhagchand Sethiya (Team Leader)

Reema Gopal Israni

Rahul Prakash Sawra

Table of Contents

1 INTRODUCTION

1.1 Overview

The proposed solution uses tweets to analyse sentiments of Indians during the Corona Pandemic. The solution would take into account sentiments and corona case data together to provide analysis. The proposed solution would provide the analysis of social sentiment regarding Covid-19 pandemic and government decision on lockdown extension. The solution would also provide the social sentiment analysis regarding various critical issues that people face on topics like hospitals, workers, PM Relief Fund, economy and lockdown. Live sentiment analysis is also provided. Finally, all the results and insights are presented on a visualisation dashboard.

1.2 Purpose

The best way to capture the impact of the Corona Pandemic on human lives is social media. People express their problems and support on platforms like Twitter and by analysing the responses one can gain more insights into the situation. These insights are essential to understand and solve critical problems to help all classes of society and create a better world for everybody to live in. The proposed solution provides complete analysis and visualisations of Covid-19 tweets and will help in quick understanding of social sentiment and provide useful insights into the situation.

2 LITERATURE SURVEY

2.1 Existing Problem

Social media platforms like Twitter have millions of people tweeting every second about their opinions on different situations. Summarizing tweets and opinions is not feasible for a human. In this Covid-19 Pandemic situation, there might be different opinions on different situations and decisions. Mining such opinions can be valuable but require huge time and effort. The proposed solution would provide the analysis of tweets and visualisations for quicker analysis of the situation.

2.1 Proposed Solution

The proposed solution uses tweets to analyse sentiments of Indians during the Corona Pandemic.
The analysis is divided into the following modules:

1. Sentiment Analysis
   The tweets are categorised into the following 5 sentiments: Fear, Anger, Joy, Sadness, Neutral. Another categorisation provided is Positive, Negative and Neutral. Also, a positivity score is calculated that displays the rise and fall in the positivity of people during the pandemic.
   Proper visualisations of the sentiments are provided in order to understand the social sentiment for an interval of 3 months (1st February 2020- 30th April 2020)
2. Topics and trending hashtags
   Topic Modelling is used to extract discussed topics for every 5 day interval Period. Visualisations are provided in the form of a timeline of topics discussed and top 5 trending hashtags.
3. Social Sentiment on critical topics
   Separate analysis and visualisation are provided for obtaining and understanding the sentiment of people on some critical topics like hospitals, PM Relief fund, problems faced by workers and job issues, lockdown situation and economic crisis.
4. Predictive Analytics on the lockdown situation
   The predictive analytics on lockdown indicates the rise and fall of the positivity score (indicating positivity) if the government decides to extend the lockdown. The score provided is dynamic and will change according to the current cases of Covid-19.
5. Analysis of Live tweets
   Visualisations of sentiment on the live tweets are provided to emphasize on the current discussions. The analysis includes the sentiments of live tweets and the word cloud. The analysis is provided for the topics and keywords mentioned by the user.

3 THEORETICAL ANALYSIS

3.1 Block diagram

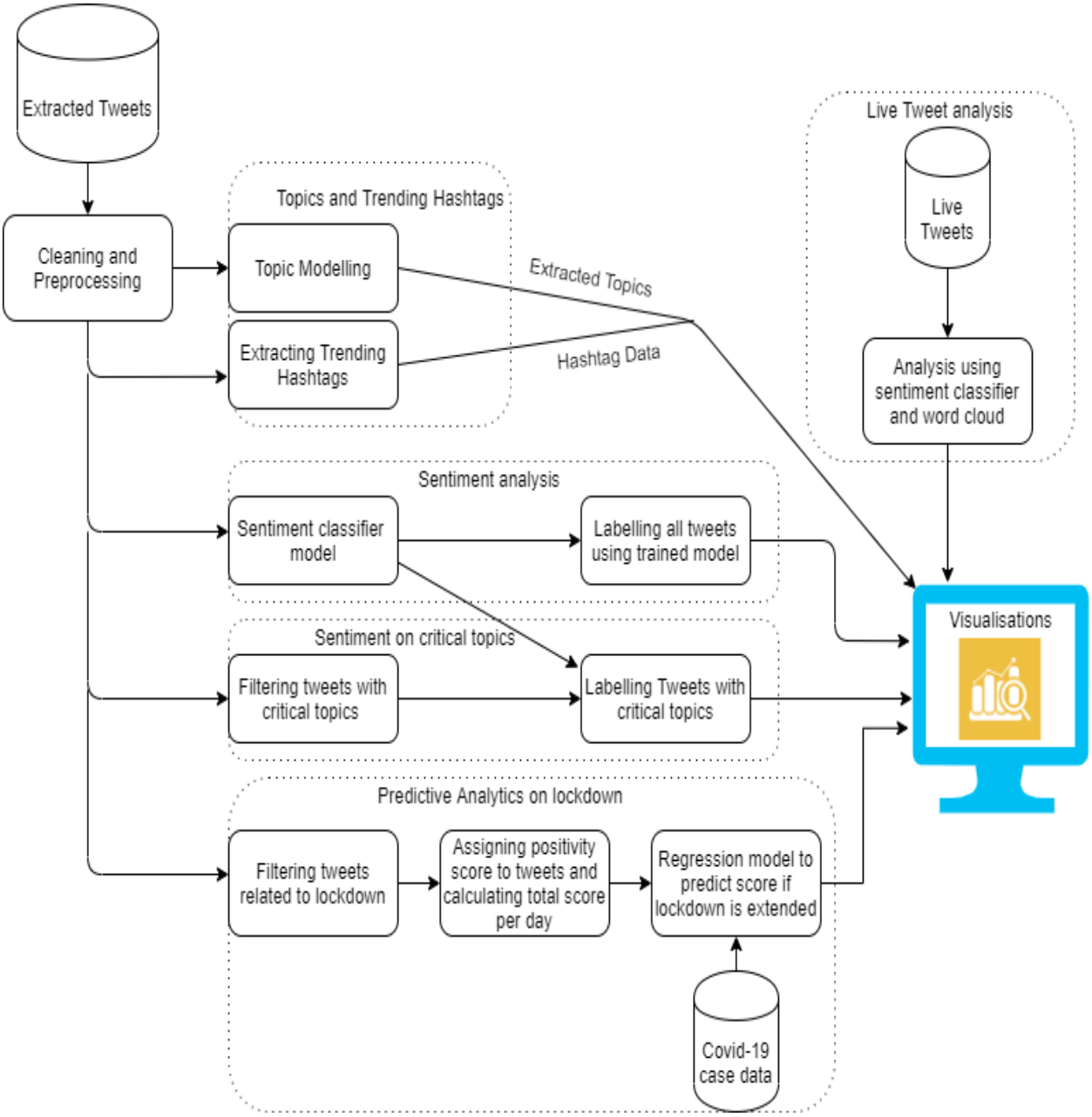

Figure 3.1.1 Block Diagram

3.2 Implementation Details

1. **Data Extraction, cleaning and preprocessing**

   Data Extraction: Geo Data and Covid-19 tweets were collected from GeoCoV19 [6]. The data provided consists of tweet IDs and Geodata of the tweets.

   Keywords included in the data can be found at [7].

   Only the tweet IDs with geolocation as India in the English Language were filtered and hydrated to obtain full tweet texts and hashtags. Hydration is performed using the tool Twarc [9].

   Cleaning: Removing URLs, mentions, other non-language characters, stop words, tokenization and lemmatization of text.

   Preprocessing: The tweet text is converted to numerical data.

   Preprocessing is different for different modules built.

   1. Sentiment Analysis using (Multiclass Classification BERT - Bidirectional Encoder Representations from Transformers) - BERT preprocessing for obtaining vectors of length 350 and considering 35000 features.
   2. Topic Modelling using (LDA - Latent Dirichlet allocation) - Count Vectorization for bigrams and trigrams.

2. **Sentiment Analysis**

   Labelling sentiments to tweets is done in 3 ways:

   1. **Multiclass emotion classification (anger, fear, joy, sadness, neutral) using BERT and self-training -**

      Semi-supervised Self Training is applied in 2 iterations. Steps are as follows:

      1. Generation of an initial labelled dataset using datasets EmoInt [12], DailyDialog, Emotion Stimulus and Isear [13] and 1500 manually labelled neutral tweets from the collected tweet set. (Train: Test ratio 75:25)
      2. Fine-tuning BERT in 3 iterations to obtain first self-training iteration accuracy of 83.92% on the test set.

         (The array represents Confusion Matrix)

```
              precision    recall  f1-score   support

         joy       0.87      0.89      0.88       762
     sadness       0.80      0.83      0.82       729
        fear       0.86      0.81      0.84       787
       anger       0.81      0.83      0.82       738
     neutral       0.85      0.83      0.84       728

    accuracy                           0.84      3744
   macro avg       0.84      0.84      0.84      3744
weighted avg       0.84      0.84      0.84      3744

array([[676,  17,  13,   9,  47],
       [ 20, 603,  32,  60,  14],
       [ 27,  53, 641,  47,  19],
       [  7,  48,  40, 615,  28],
       [ 48,  29,  17,  27, 607]])
```

Figure 3.2.1 Evaluation Metrics(Self Training Iteration 1)

3. Using the first iteration trained BERT model to label tweets of Covid-19.
4. Filtering tweets having label confidence > 95%
5. Generation of second iteration dataset

   Train set: 3000 labelled tweets for each class, 1000 labelled texts from the dataset of iteration 1 for each class

   Total train set size: 20,000

   Test set: 1000 labelled tweets for each class, 500 labelled texts from the dataset of iteration 1 for each class

   Total test set size: 7500
6. Fine-tuning BERT in 3 iterations to obtain second self-training iteration accuracy of 93.38% on the test set.

   (The array represents Confusion Matrix)

```
              precision    recall  f1-score   support

         joy       0.96      0.94      0.95      1450
     sadness       0.92      0.93      0.92      1463
        fear       0.93      0.93      0.93      1469
       anger       0.94      0.92      0.93      1488
     neutral       0.92      0.95      0.94      1352

    accuracy                           0.93      7222
   macro avg       0.93      0.93      0.93      7222
weighted avg       0.93      0.93      0.93      7222

array([[1365,   21,    9,    9,   46],
       [  13, 1361,   33,   34,   22],
       [  17,   38, 1363,   36,   15],
       [  10,   39,   48, 1368,   23],
       [  13,   25,   20,    7, 1287]])
```

Figure 3.2.2 Evaluation Metrics (Self Training Iteration 2)

7. Using trained BERT to label all tweets.

2. **Positive, Negative and Neutral classification -**

   This classification was performed using Vader [22]. The compound score was obtained for individual tweets and the compound scores were converted to labels 0 (Neutral), 1 (Positive) and -1 (Negative) using the threshold values [22]:
   Positive sentiment: compound score >= 0.05
   Neutral sentiment: (compound score > -0.05) and (compound score < 0.05)
   Negative sentiment: compound score <= -0.05

3. **Positivity Score -**

   Positivity score is the compound score assigned to individual tweets using Vader [22]. The score is calculated for 1 day time period from 1st Feb 2020 to 30th April 2020. The positivity score for 1 day is obtained using the formula - Positivity score = Total score of individual tweets / Count of tweets. The score indicates the positivity of people on that day. A higher score indicates high positivity. The score can also be negative indicating negative sentiment. Results on analysis are then plotted using different visualisations.

3. **Topics and Trending Hashtags**

   The time period interval is 5 days each for Topics extraction and Trending Hashtags.
   Topic Modelling:
   Algorithm Used: LDA (Latent Dirichlet Allocation)
   Count Vectorized bigrams and trigrams are given as input to LDA and the output is set to 10 top words each for 10 topics.
   Trending Hashtags:
   Hashtags are extracted from tweets and the count of top 5 hashtags is plotted for 5 days time interval.

4. **Sentiment analysis on critical topics**

   The tweets related to the critical topics like hospitals, workers, economy, PM Relief Fund and lockdown are filtered using the following keywords:

   Hospitals - ['hospital','bed','ventilator']
   Care Fund - ['pm care', 'pm care fund','care fund','pmcarefund','pmcare','carefund','pm cares fund','pmcaresfund','pm cares','cares fund','caresfund','pmcares','relief

fund','relieffund','pmrelieffund','pm relief funds','pmrelieffunds','relieffunds']
Lockdown - ['lockdown','lock down','stayhome','stay home','quarantine','lock down extension','lockdown extension','lockdownextension','stayhomesavelives','stayhome save life','stayhome savelife','self isolation','selfisolation','selfquarantine']
Economy - ['economy','economycollapse','economy collapse','financialcrisis','financial crisis','debt']
Workers - ['worker','migrant','labourer','job']

These filtered tweets are then labelled according to the point 2) of implementation details (sentiment analysis) and the visualisations are plotted for 5 emotions fear, anger, joy, sadness, neutral; positive, negative, neutral and the positivity score graphs.

5. **Predictive analytics on lockdown situation**

Gradient Boosting Regression model was used to predict the positivity score (daily) if the government decides to extend lockdown. (The score depends on the case data from 2 days before the date the score is to be obtained, so a lag of 2 days in introduced before giving data as input to the algorithm).

Input Features: Confirmed Cases, Active Cases, Deaths, Recovered Cases
Output: Positivity score indicating how positive will the people be if the government decides to extend lockdown.

	Train metrics	Test metrics
Mean Absolute Error (MAE)	0.0262	0.0312
Root Mean Square Error (RMSE)	0.0501	0.0496
Mean Squared Error (MSE)	0.0025	0.0025
R2 Score	0.815	0.8214

Table 1: Evaluation metrics of Trained model (Train:Test ratio = 75:25)
(Rounded up to 4 decimal places)

The output provided is dynamic and depends on current Covid-19 Cases in India.
A Python REST API is built that can be called from the Dashboard to plot the predictive

lockdown analytics graph daily.
API performs the following steps:
1) Get Current Date
2) Get Case data for the dates for which it is not available
Case data is updated daily at the URL (https://api.covid19india.org/data.json)
3) Extract confirmed count, active count, deaths count and recovered count.
4) Give the collected data to the Trained model to obtain predictions.
5) Send the prediction values to the UI for plotting.

6. **Live Tweet Analysis**

   Python REST API is built which can be called from the User dashboard. The user can provide keywords or hashtags for which analysis in needed. The API uses Twitter API (Tweepy) to fetch the tweets and the labels the tweets using trained sentiment classifier. The results are then sent back to UI for plotting visualisations along with tweets text and word cloud.

3.3 Software Designing

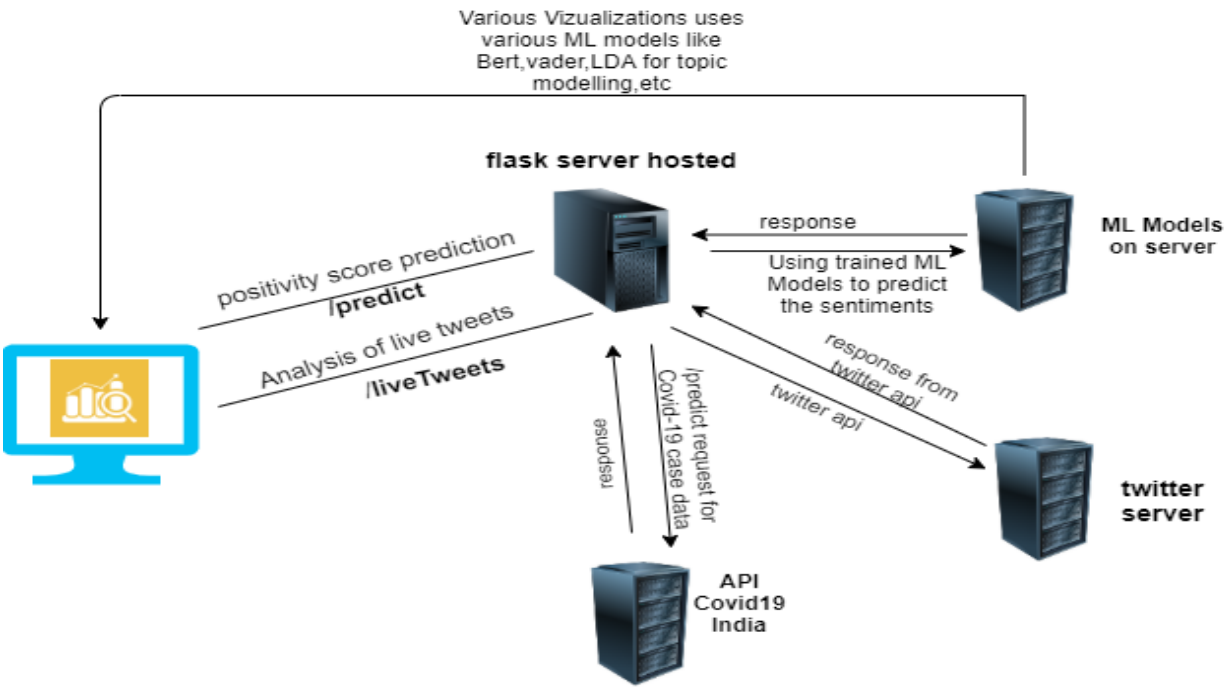

Figure 3.3.1 Software Design

/predict - Python Flask REST API for prediction of positivity score if government decides to extend lockdown. It is hosted on Flask Server. The process takes current case data and uses trained model to predict the score and the output is sent back for plotting the visualisations.

/liveTweets - Python Flask REST API for Live Sentiment Analysis. It is hosted on Flask Server. The process sends request to Twitter for getting current Tweets and uses trained model to predict the sentiment and the output is sent back for plotting the visualisations.

4 EXPERIMENTAL INVESTIGATIONS

Sentiment Analysis:

The tweets extracted were unlabelled.

Initially, we tried Unsupervised Clustering Techniques to obtain clusters of sentiments.

The algorithms tried were Density-Based Spatial Clustering of Application with Noise (DBSCAN), K-means Clustering, Spherical K-means and Agglomerative clustering (average linkage).

The results were not satisfactory.

The next method tried was Self Training (Semi-Supervised learning) for which different datasets were combined for initial iteration and the subsequent iteration included high confidence labelled tweets from the trained model of initial iteration. Unlabeled data, when used in conjunction with a small amount of labelled data, can produce a considerable improvement in learning accuracy.

The method was used in combination with BERT ((Bidirectional Encoder Representations from Transformers) in order to obtain high accuracy for multiclass classification of tweets into anger, fear, joy, sadness and neutral classes.

Predictive analytics on lockdown situation:

Correlation was observed between Covid-19 case data and sentiment of people regarding lockdown. (Fig. 4.1)

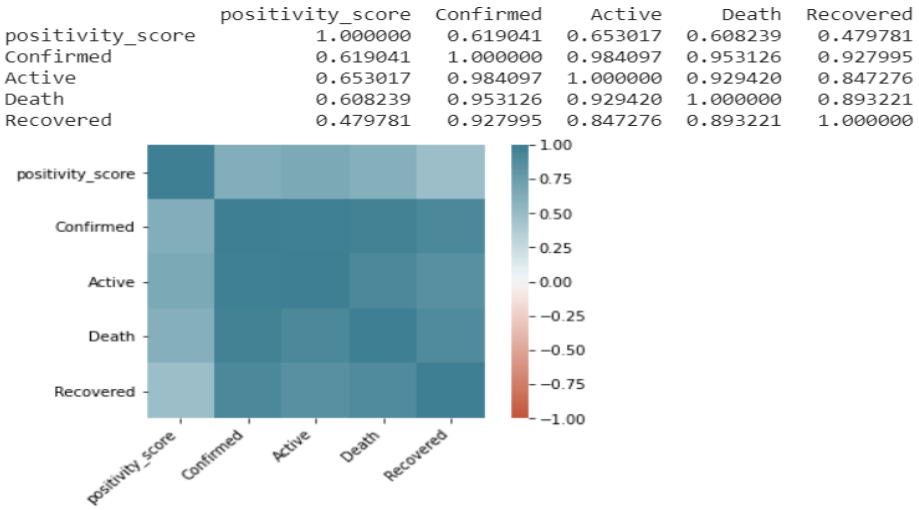

Figure 4.1 Correlation Matrix

Sentiment Score and Case data: Line graph Fig. 4.2 shows that the people had negative sentiment towards lockdown initially but the sentiment towards lockdown becomes more positive as the cases rise. (The positivity score depends on Covid-19 Case dataof 2 days before the date for which the score is observed. Hence the cases graph ie confirmed, active, death, recovered is plotted with a lag of 2 days).

These observations were considered to train the predictive analytics model.

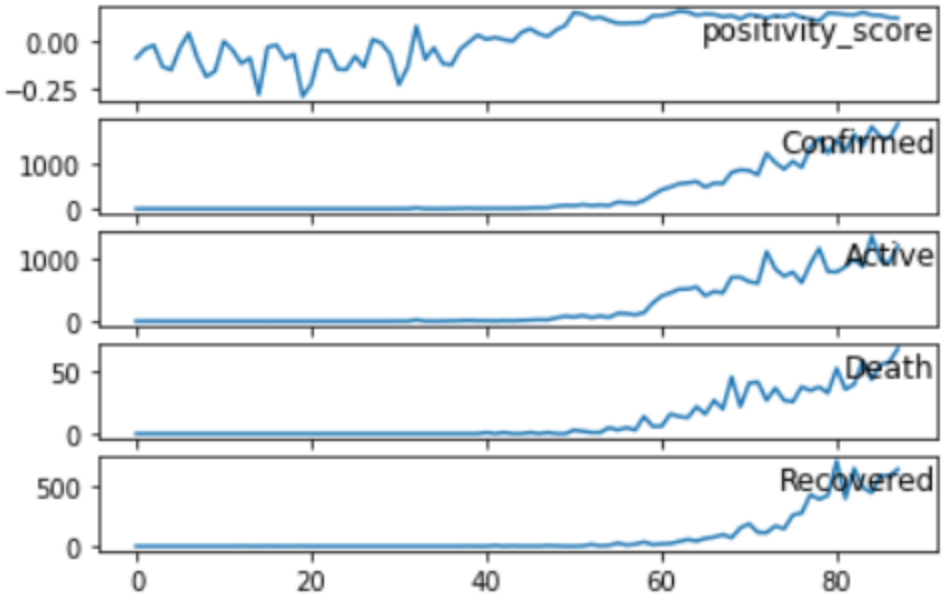

Figure 4.2 Sentiment score and Case Data

Initial analysis was based on Multivariate Time Series algorithms like LSTM and FbProphet forecasting procedures but the graphs indicate less dependency of sentiment on time variable but more on case data. Hence Regression (Gradient Boosting Regression) was chosen for final analysis.

5 FLOWCHART

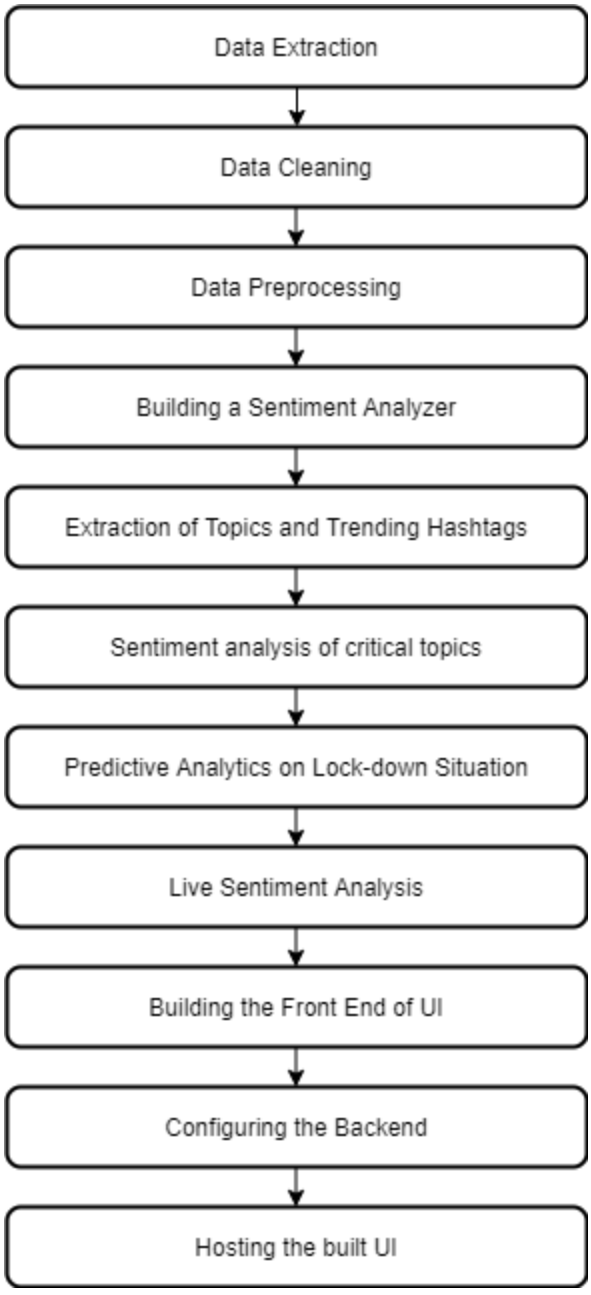

Figure 5.1 Flowchart

6 RESULTS

We have built a Visualization Dashboard to show the sentiments of the users during the spread of Covid-19. The website shows all the analysis based on user sentiments. We have categorised the sentiments as:

1. Fear, Joy, Neutral, Sadness and Anger.
2. Positive, Negative and Neutral.
3. Positivity Score (How much positive the user was through time)

The time interval used for showing analysis was:

1. Daily (From 1st Feb 2020 to 30th April 2020)
2. Monthly (For 3 months: Feb, Mar, Apr)
3. For the interval of 5 days each (For eg. 1st-5th Feb 2020)
4. Overall for 3 months in total.

The dashboard consist of the following graphs:

1. Analysis of User Sentiments like fear, joy, sadness, anger and neutral which is shown in three ways (daily, monthly, for the interval of 5 days) Fig. 6.1, 6.2 & 6.4
2. Analysis of User Sentiments categorized into positive, negative and neutral (daily and overall for the period of 3 months) Fig. 6.1 & 6.3
3. Top 5 hashtags used by users (in the interval of 5 days) Fig. 6.5
4. Top 10 topics discussed by users (in the interval of 5 days) Fig. 6.6
5. The sentiment of users whenever lockdown was extended and also the prediction of sentiments if lockdown is further extended. Fig. 6.7 & 6.8
6. 5 most critical topics discussed by users and their sentiments (daily and overall for the period of three months) Fig. 6.9, 6.10, 6.11 & 6.12
7. Rise in COVID-19 cases all over the world. Fig. 6.13 & 6.14
8. User sentiments analysed for live tweets of keywords provided by the user. Fig. 6.15, 6.16 & 6.17

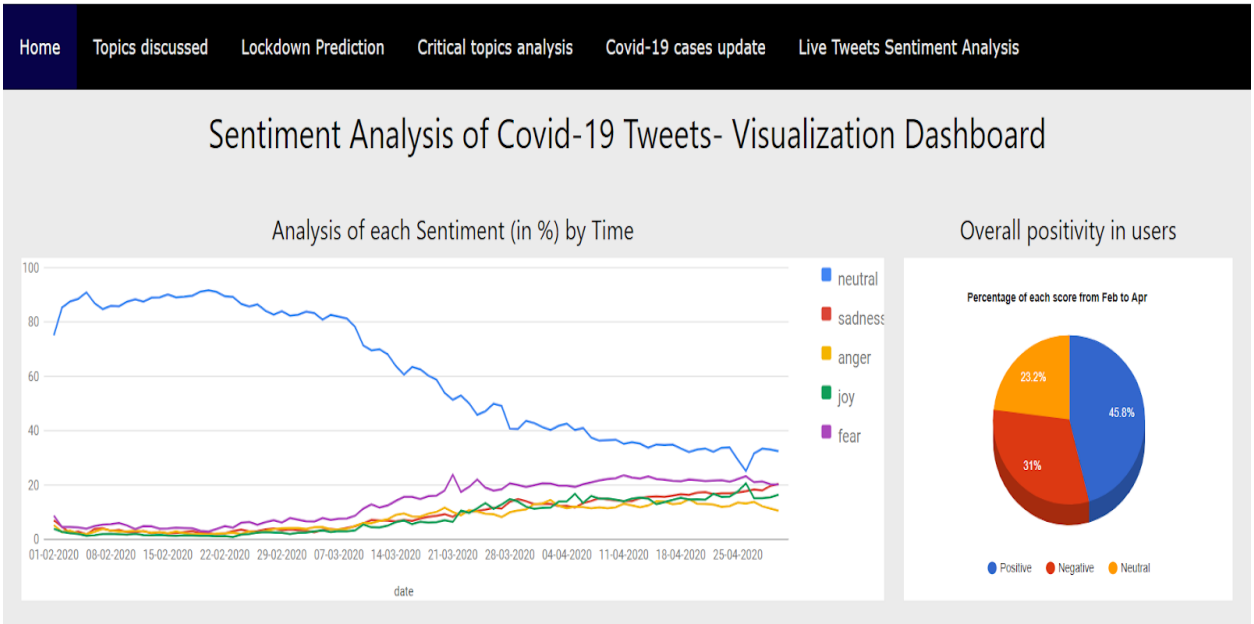

Figure 6.1

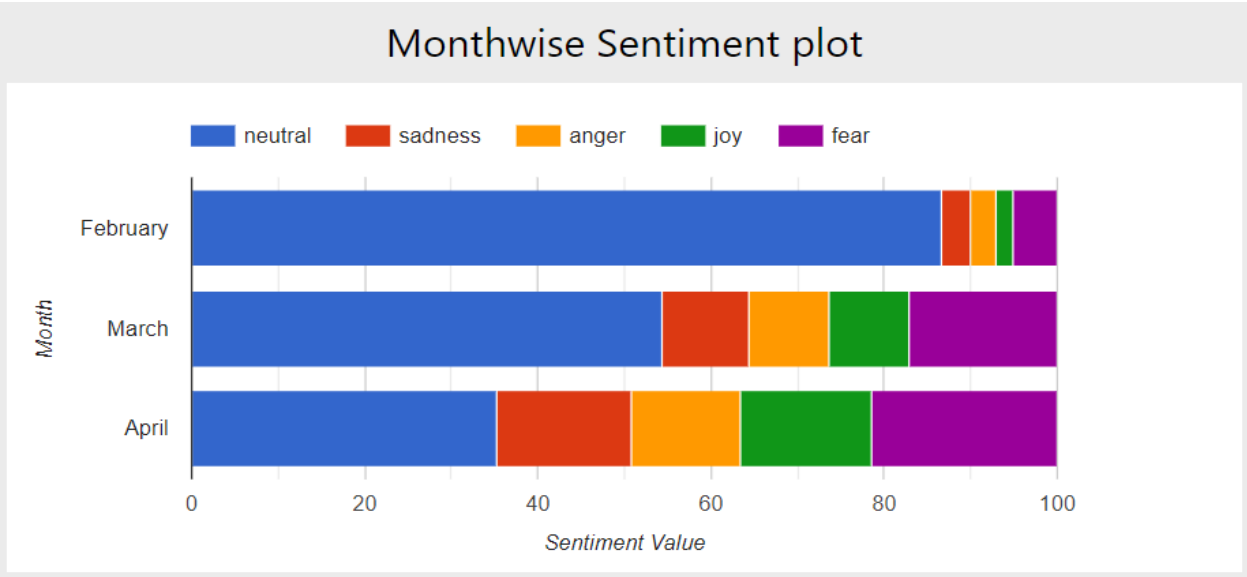

Figure 6.2

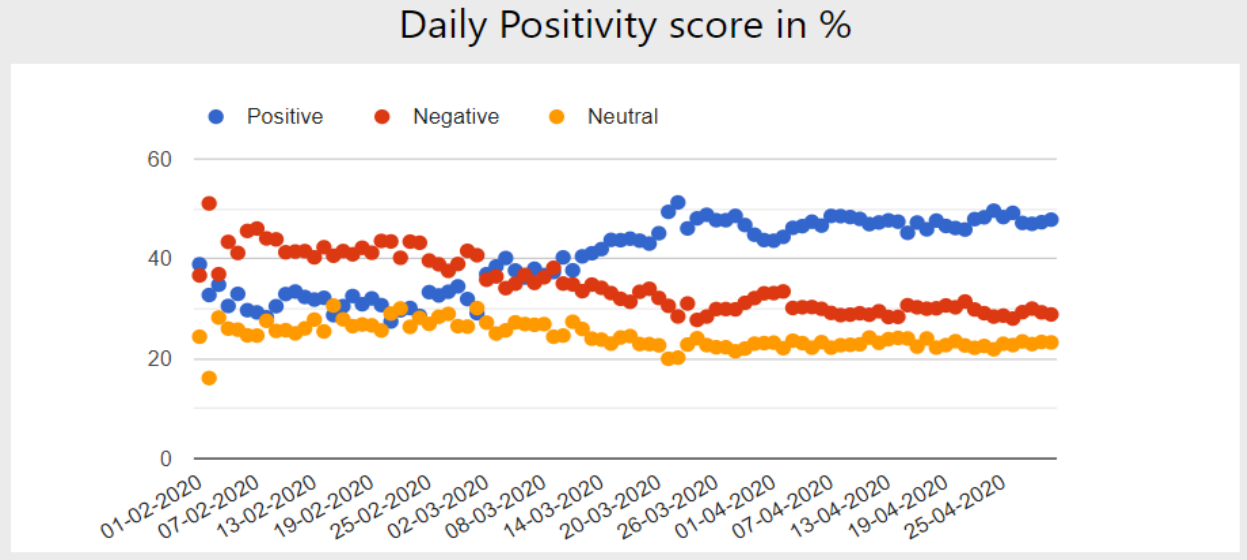

Figure 6.3

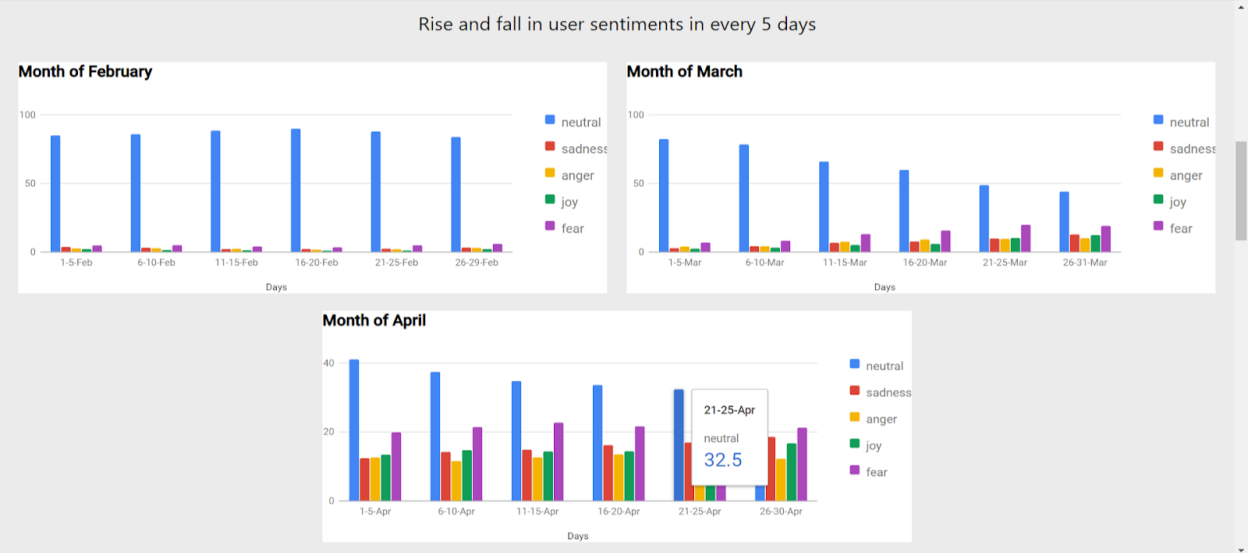

Figure 6.4

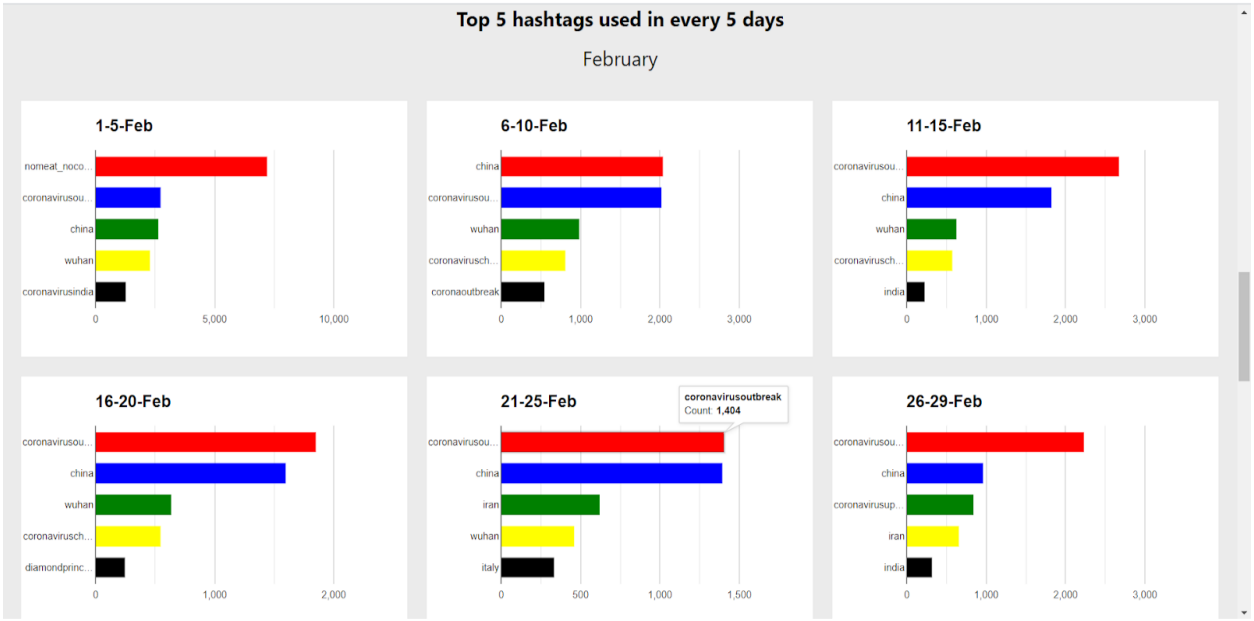

Figure 6.5

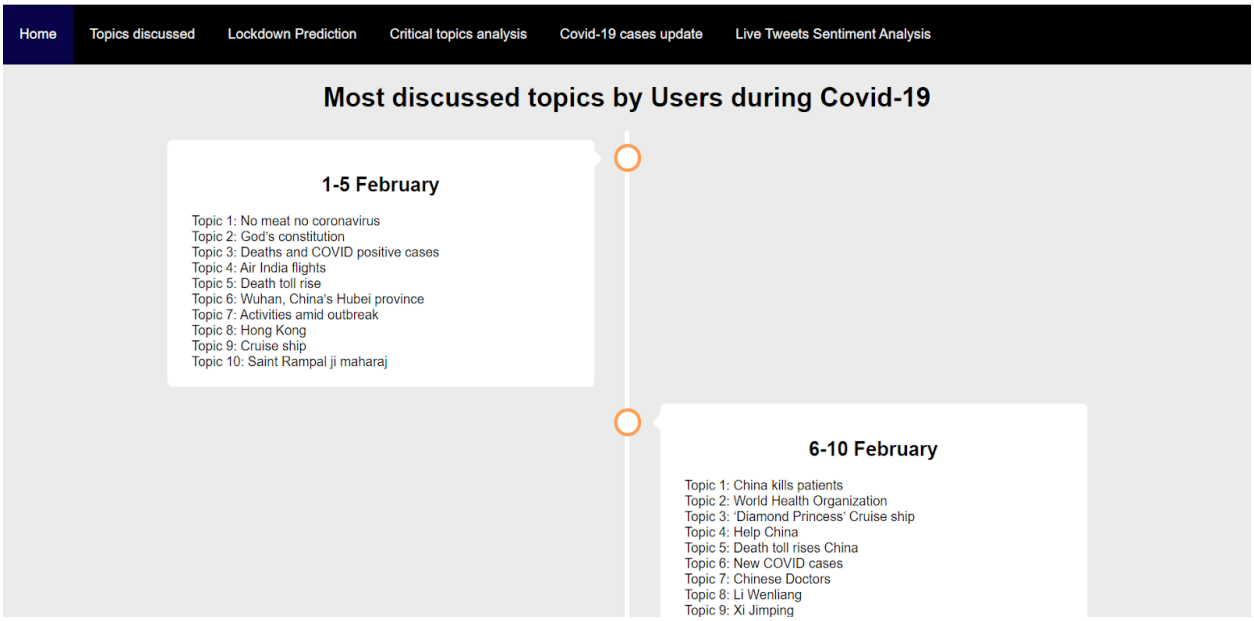

Figure 6.6

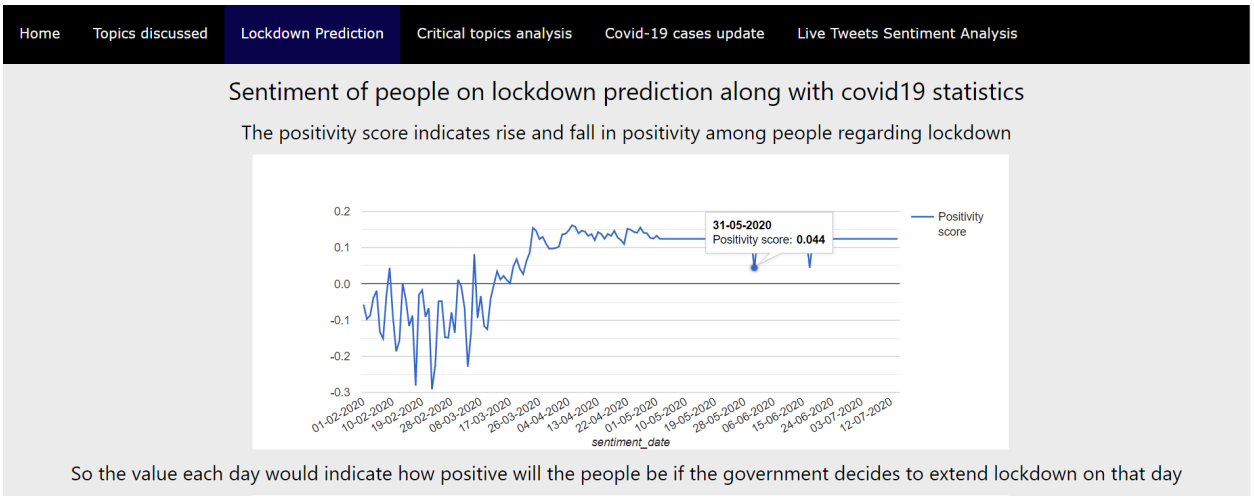

Figure 6.7

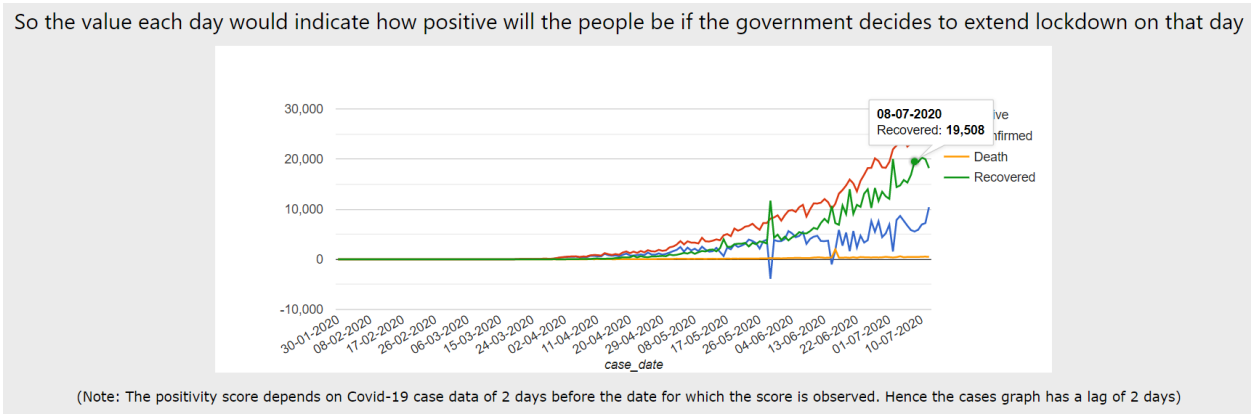

Figure 6.8

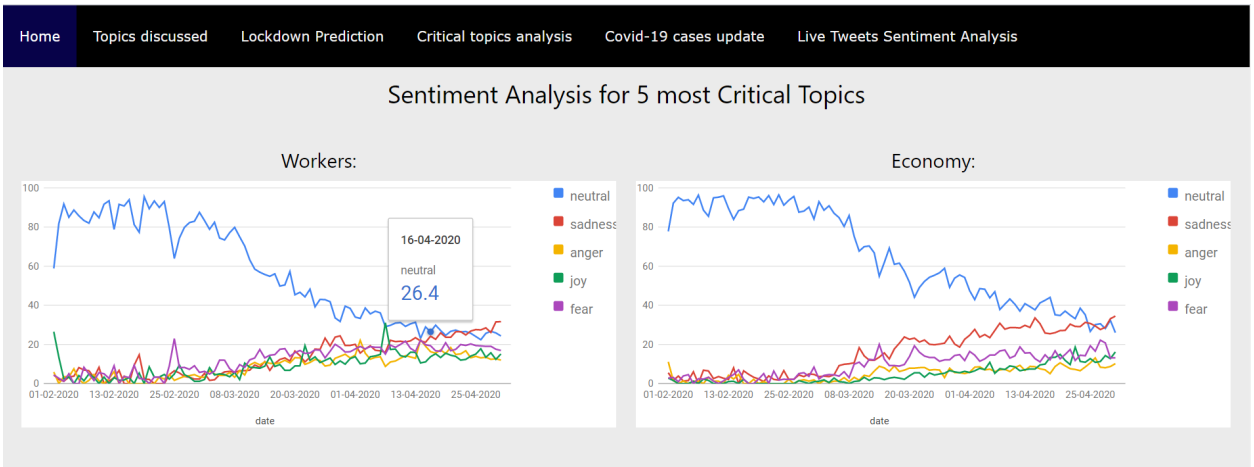

Figure 6.9

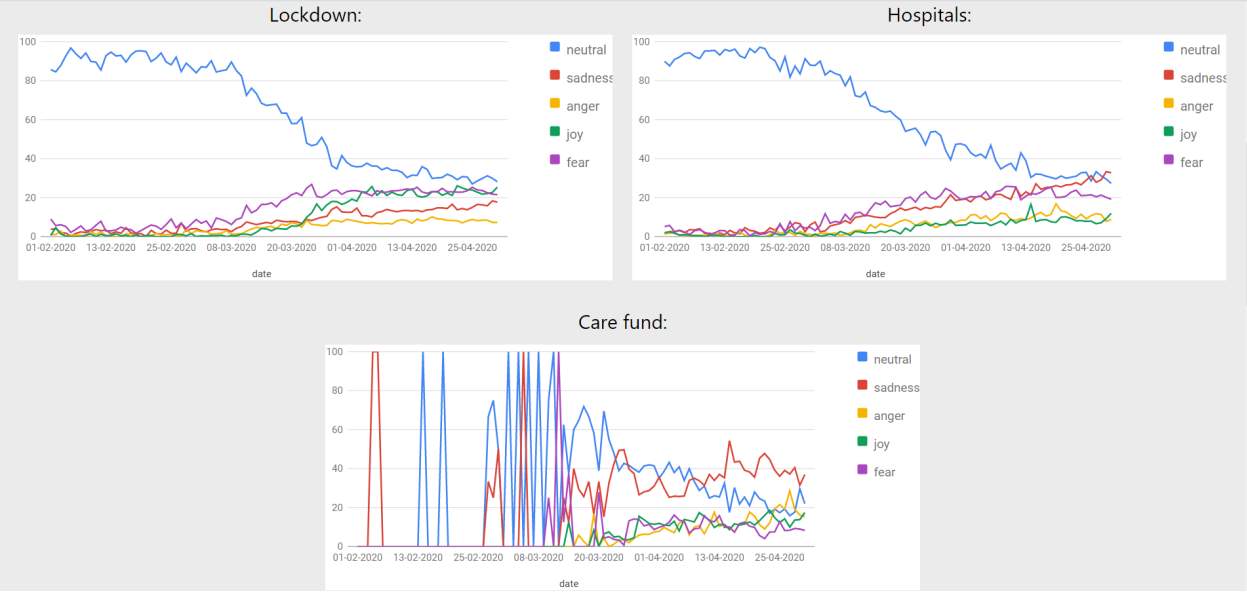

Figure 6.10

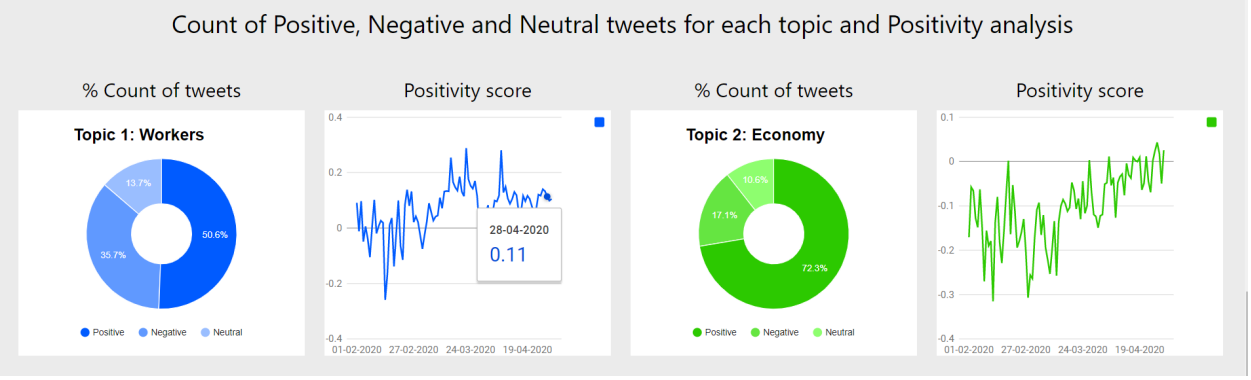

Figure 6.11

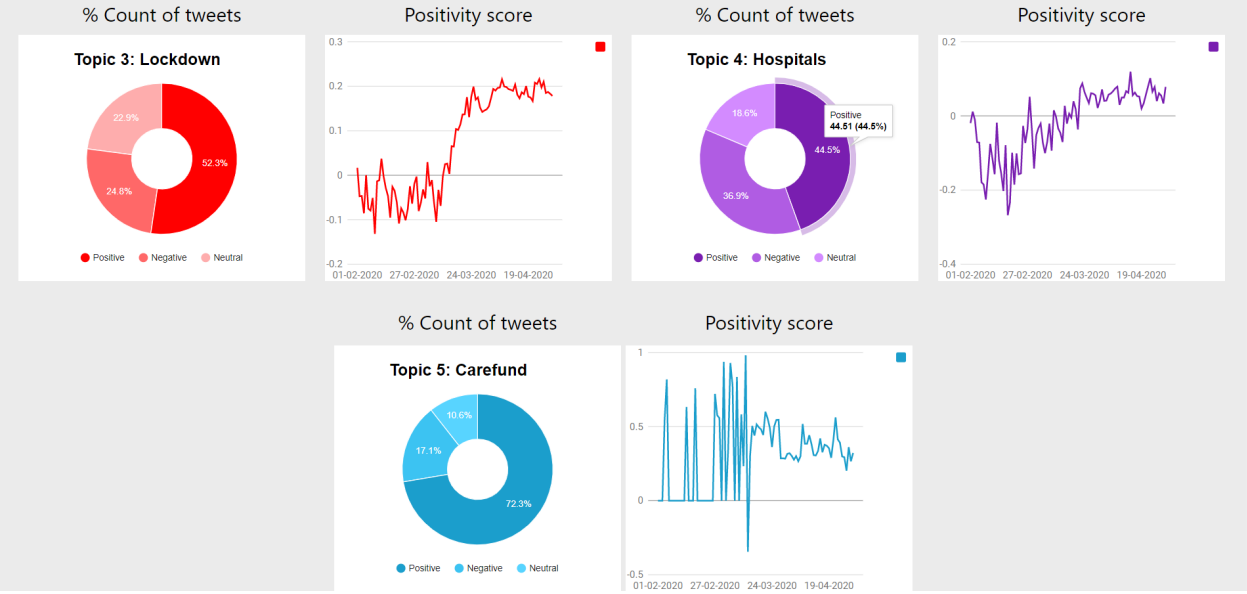

Figure 6.12

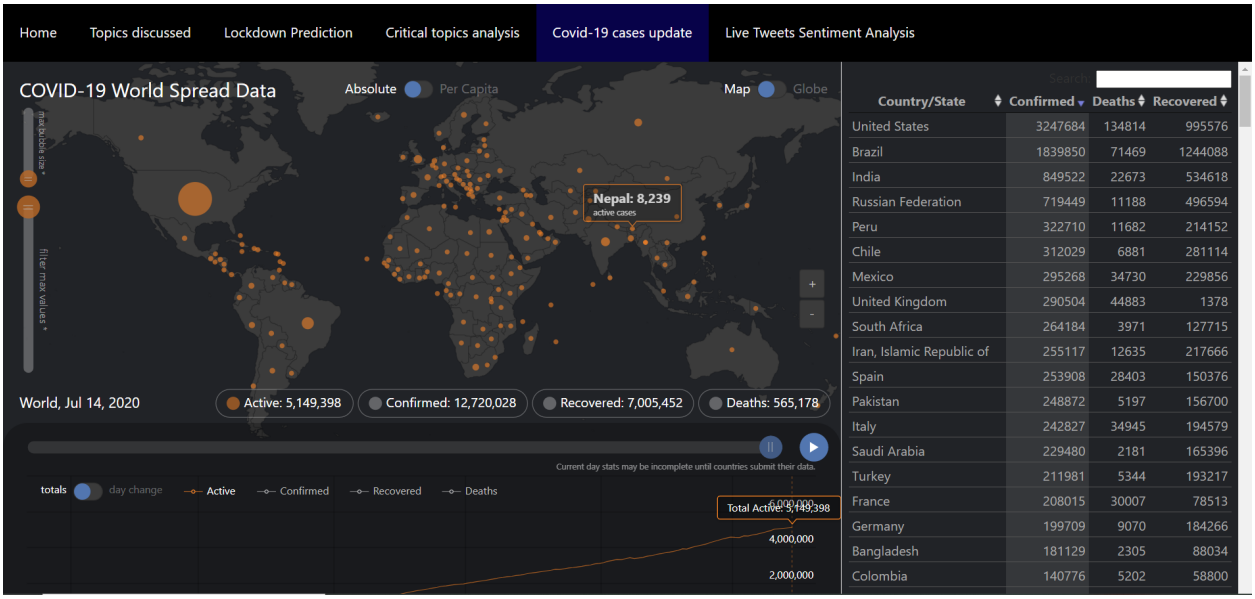

Figure 6.13

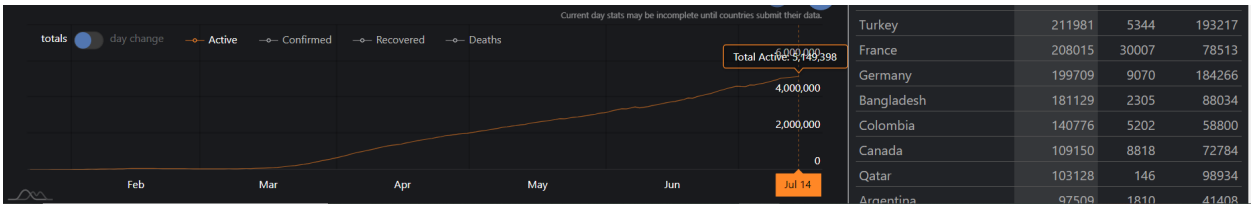

Figure 6.14

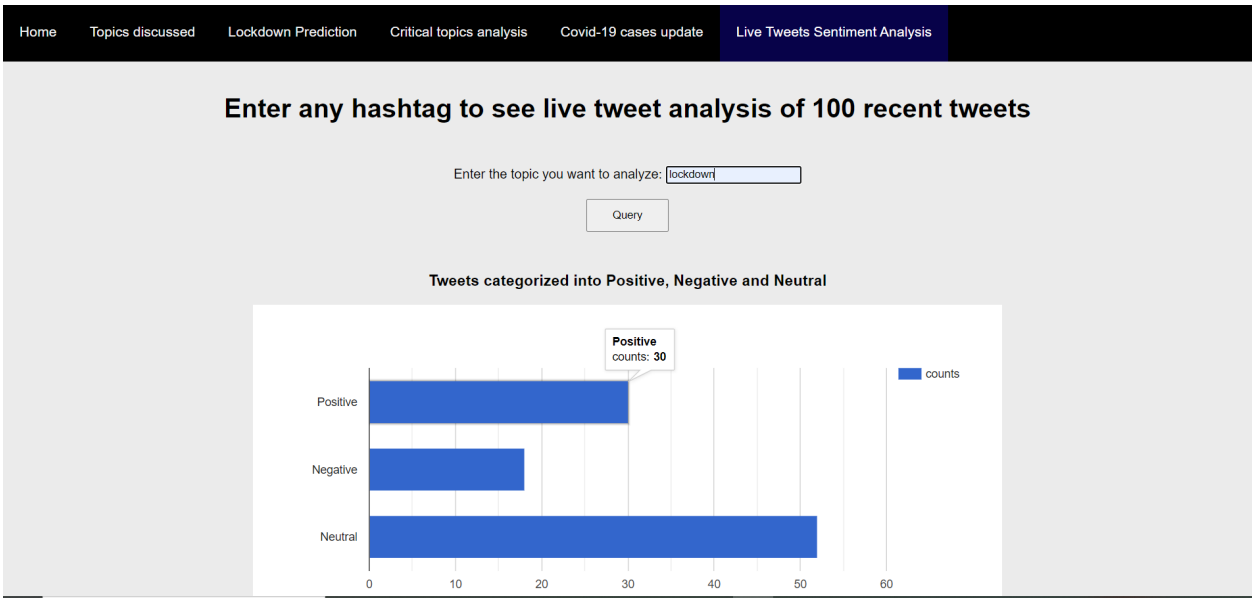

Figure 6.15

Live tweets extracted from Twitter

sentiment	text
Positive	RT With an open mind one can discover within the virtual dimension new opportunities that do not exist in physical engage
Positive	Ambleside thankyou for a fantastic meal First meal out after lockdown and first meal out with our baby
Neutral	RT Importano positivi per mantenere il lockdown ma non contenti dobbiamo pagargli le navi per la quarantena cu
Negative	Have I fucking missed something So we're now being told we have to wear face masks or face a fine of quid
Neutral	RT Coronavirus Lockdown Vaccine Indi
Neutral	Don't you see that the whole aim of Newspeak is to narrow the range of thought In the end we shall make thought c
Positive	RT I just won myself a new T T shirt from CASHMERE recordz Ghana COVID se
Neutral	RT louma lo
Positive	RT Beautiful Oil paint on wood Wooden finish black frame Size x cm R Kindly RT like or DM if interested Artist
Neutral	RT Banksy nei panni di un insospettabile sanificatore colpisce ancora indisturbato Nei vagoni della metro di Londra realiz
Neutral	RT lockdown photography a cloud photo as sunset approaches for when there s not many much else you can see stormhour
Neutral	SA full hours without electricity ka lockdown isoul yomuntu uyakwazi ukuyibulala more than covid
Neutral	RT portoEmpedocle arrivati migranti alcuni dei quali scappano dall hotspot mentre sono in quarantena Cos riescono a
Negative	BEFORE MEETS AFTER We re available for birthday glam photo shoot and weddings Follow empire on IG for bo
Positive	RT LockDown is actually a Joke
Neutral	I dont know if I'm an Indian because I haven't watched TV for Month tuesdayvibes lockdown
Neutral	lockdown juillet FeteNationale armyisoverparty Macron h
Positive	RT HQ ukeff oldham Lovely kind gesture that will make a big difference to many in need Tha
Neutral	RT ArianaGrande ariana moots thankunext sweetner Arianators arianator mutuals ari lockdown Quarantine Quaranti

Figure 6.16

Word Cloud of most used words

Figure 6.17

7 ADVANTAGES & DISADVANTAGES

7.1 Advantages

1. The proposed solution makes analysis of tweets easier and faster and provides useful insights into the social sentiment during the Covid-19 Pandemic.
2. The dashboard provides overall sentiment analysis and social sentiment on different critical topics like hospitals, PM Care Fund, economy etc.

3. Dashboard also provides insights into the topics discussed by the people.
4. The predictive analytics helps understand the social sentiment if the government decides to extend lockdown.
5. Live sentiment analysis is also provided for gaining insights into current discussion.

7.2 Disadvantages

1. The analysis is over a period of 3 months from February - April. More data and computational power is required for further analysis.
2. The Live sentiment analysis takes into account current 100 tweets as increasing the number would lead to slower results.
3. There might be other factors like Occupation of people on which the sentiment of people regarding lockdown depends. These demographics like occupation were not considered while training the predictive analytics model due to unavailability of data.

8 APPLICATIONS

1. Sentiment analysis of tweets
2. Predictive Analytics of Lockdown Situation
3. Topic Extraction from tweets
4. Live Sentiment Analysis

9 CONCLUSION

The report discusses the overall implementation of Sentiment analysis of Covid-19 tweets visualisation dashboard. The results include different visualizations for emotion (fear, anger, neutral, joy, sadness), sentiment (positive, negative, neutral), and positivity score (positivity measure) of people during the Covid-19 pandemic. Visualisations also include sentiment analysis on critical topics like hospitals, economy, PM care fund, workers and lockdown. Topics discussed, trending hashtags, prediction of social sentiment if lockdown is further extended and sentiment analysis of live tweets is also included in the built visualisation dashboard. Thus, analysis and visualisations provided of Covid-19 tweets will help in quick understanding of social sentiment and provide useful insights into the situation.

10 FUTURE SCOPE

1. Analysis can be extended further in the timeline.
2. Number of critical topics currently considered is 5. Number of topics can be increased to understand social sentiment on those topics.
3. Occupation and other demographic data can be considered to increase the accuracy of predictive analytics model (For obtaining sentiment if lockdown is extended).
4. More number of live tweets can be considered for live sentiment analysis.

11 BIBLIOGRAPHY

[1] Umair Qazi, Muhammad Imran, Ferda Ofli. GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information. ACM SIGSPATIAL Special, May 2020. doi: 10.1145/3404111.3404114

[2] M. Munikar, S. Shakya and A. Shrestha, "Fine-grained Sentiment Classification using BERT," 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 2019, pp. 1-5, doi: 10.1109/AITB48515.2019.8947435.

[3] D. Wu, M. Shang, G. Wang and L. Li, "A self-training semi-supervised classification algorithm based on density peaks of data and differential evolution," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6, doi: 10.1109/ICNSC.2018.8361359.

[4] M. Zhou, Y. Li, H. Lu, C. Nengbin and Z. Xuejun, "Semi-Supervised Meta-Learning via Self-Training," 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2020, pp. 1-7, doi: 10.1109/ICoIAS49312.2020.9081851.

[5] A. F. Hidayatullah and M. R. Ma'arif, "Road traffic topic modeling on Twitter using latent dirichlet allocation," 2017 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, 2017, pp. 47-52, doi: 10.1109/SIET.2017.8304107.

[6] GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information. [online] https://crisisnlp.qcri.org/covid19

[7] Keywords and Hashtags. [online] https://crisisnlp.qcri.org/covid_data/COVID19_AIDR_Keywords.zip

[8] Tweepy to extract Live tweets. [online] https://www.tweepy.org/

[9] A command line tool (and Python library) for archiving Twitter JSON - DocNow/twarc. [online] https://github.com/DocNow/twarc

[10] 5 first steps for Natural Language Processing. [online] https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306be

f0f
[11] Sentence Embeddings with BERT & XLNet. [online]
https://github.com/UKPLab/sentence-transformers
[12] EmoInt [online] http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html
[13] Daily Dialog, Emotion Stimulus and Isear dataset. [online]
https://github.com/lukasgarbas/nlp-text-emotion/tree/master/data/datasets
[14] Vader, IBM Watson or TextBlob. [online]
https://medium.com/@Intellica.AI/vader-ibm-watson-or-textblob-which-is-better-for-unsupervised-sentiment-analysis-db4143a39445
[15] Unsupervised Sentiment Analysis. [online]
https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483
[16] Tao, J., Fang, X. Toward multi-label sentiment analysis: a transfer learning based approach. J Big Data 7, 1 (2020). https://doi.org/10.1186/s40537-019-0278-0
[17] Twitter Sentiment Analysis Based on News Topics during COVID-19. [online] https://towardsdatascience.com/twitter-sentiment-analysis-based-on-news-topics-during-covid-19-c3d738005b55
[18] Topic Extraction from Tweets using LDA. [online] https://medium.com/@osas.usen/topic-extraction-from-tweets-using-lda-a997e4eb0985
[19] Topic Modelling (LDA) on Elon Tweets. [online] https://www.kaggle.com/errearanhas/topic-modelling-lda-on-elon-tweets
[20] Multivariate Time Series Forecasting with LSTMs in Keras. [online]
https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/
[21] fbprophet [online] https://pypi.org/project/fbprophet/
[22] Vader Sentiment [online] https://pypi.org/project/vader-sentiment/
[23] Covid-19 Dashboard [online] https://covid-19.splunkforgood.com/hub
[24] COVID19-India API (for getting daily case data) [online] https://api.covid19india.org/

12 APPENDIX

A. Source code

1. Building a Sentiment Analyser
Emotion sentiment classification (fear, anger, neutral, joy, sadness) using semi-supervised self training and fine-tuning on BERT

```
1  #BERT
```

```
encoding = {
    'joy': 0,
    'sadness': 1,
    'fear': 2,
    'anger': 3,
     'neutral': 4
}
model=text.text_classifier('bert',train_data=(x_train,y_train),preproc=preproc)
learner=ktrain.get_learner(model,train_data=(x_train,y_train),val_data=(x_test, y_test),batch_size=6)
learner.fit_onecycle(2e-5, 3)
learner.validate(val_data=(x_test, y_test), class_names=class_names)
predictor = ktrain.get_predictor(learner.model, preproc)
```

Sentiment classification (Positive, Negative, Neutral) using Vader-sentiment

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()
def sentiment_analyzer_scores(sentence):
    score = analyser.polarity_scores(sentence)
    sentiments = score['compound']
    return sentiments
sentiment =[]
for s in df['text']:
  senti = sentiment_analyzer_scores(s)
  sentiment.append(senti)
df['sentiment'] = sentiment
def label(sentiment_value):
      if(sentiment_value>=0.05):
            return 1
      elif(sentiment_value<=-0.05):
            return -1
      else:
            return 0
df['sentiment_label']=df['sentiment'].apply(label)
```

Sentiment Positivity score :

```
#daily score
y = sorted(set(df['date']))
date = []
sentiment_scores = []
for i in range(len(y)):
  x = y[i]
```

```
    count = 0
    sentiment_score=0
    for index,row in df1.iterrows():
      if(x==row['date']):
        sentiment_score+=row['sentiment']
        count+=1
    date.append(x)
    sentiment_scores.append(sentiment_score/count)
```

2.Topic Modelling

```
#LDA
from sklearn.decomposition import LatentDirichletAllocation
lda=LatentDirichletAllocation(n_components=10,max_iter=5,learning_method=
'online', learning_offset=50.,random_state=0).fit(tf)
#display topics
def display_topics(model, feature_names, no_top_words):
  s=''
  for topic_idx, topic in enumerate(model.components_):
    s=s+("Topic %d:" % (topic_idx))
    s=s+(",".join([feature_names[i]
                   for i in topic.argsort()[:-no_top_words - 1:-1]]))
    s=s+'\n'
  return s
no_top_words = 10
display_topics(lda, tf_feature_names, no_top_words)
#perplexity
lda.perplexity(tf)
```

3. Sentiment analysis on Critical Topics

```
#Emotion classification daily(fear, anger, joy, sadness, neutral)
searchfor = list_of_keywords
df1=df[df['cleaned'].str.contains('|'.join(searchfor))]
import datetime
full=[]
start_date = datetime.date(2020, 2, 1)
end_date = datetime.date(2020, 4, 30)
delta = datetime.timedelta(days=1)
while start_date <= end_date:
    temp=df1[df1['date']==str(start_date)]
    n=temp[temp['emotion']=='neutral']
    s=temp[temp['emotion']=='sadness']
    a=temp[temp['emotion']=='anger']
    j=temp[temp['emotion']=='joy']
    f=temp[temp['emotion']=='fear']
```

```
    nc=n.shape[0]
    sc=s.shape[0]
    ac=a.shape[0]
    jc=j.shape[0]
    fc=f.shape[0]
    total=nc+sc+ac+jc+fc
    if(total!=0):
      ncp=(nc/total)*100
      scp=(sc/total)*100
      acp=(ac/total)*100
      jcp=(jc/total)*100
      fcp=(fc/total)*100
    else:
      ncp=0
      scp=0
      acp=0
      jcp=0
      fcp=0
    row=[start_date,nc,sc,ac,jc,fc,total,ncp,scp,acp,jcp,fcp]
    full.append(row)
    start_date += delta
dff=pd.DataFrame(full,columns=['date','neutral_count','sadness_count','ang
er_count','joy_count','fear_count','total_count','neutral_count_pct','sad
ness_count_pct','anger_count_pct','joy_count_pct','fear_count_pct'])
```

4. Predictive Analytics on lockdown situation

```
#gradient boosting regression
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.ensemble import GradientBoostingRegressor
grad=GradientBoostingRegressor(n_estimators=200,random_state=100,learning_r
ate=0.1,max_depth = 5,min_samples_leaf =3,min_samples_split = 12)
grad.fit(x_train,y_train)
y_pred=grad.predict(x_test)
y_tr=grad.predict(x_train)
```

```
#Lockdown REST API
@app.route('/predict',methods=['GET','POST'])
def lockdownPrediction():
        today=datetime.today().strftime('%d-%m-%Y')
        s_date=datetime.strptime('03-07-2020','%d-%m-%Y')
        e_date=datetime.strptime(today,'%d-%m-%Y')
        dates=pd.date_range(s_date,e_date,freq='d')
        case_dates=[]
```

```
        for d in dates:
            d1 = d - timedelta(days=2)
            d2=d1.strftime('%d-%m-%Y')
            case_dates.append(d2)
        x= requests.get('https://api.covid19india.org/data.json')
        data_stats = x.json()
        data = data_stats["cases_time_series"]
        confirmed=[]
        active=[]
        death=[]
        recovered=[]
        date=[]
        print(case_dates)
        for i in data:
            confirmed.append(i["dailyconfirmed"])
            death.append(i["dailydeceased"])
            recovered.append(i["dailyrecovered"])
            date.append(i["date"])

activee=int(i["dailyconfirmed"])-(int(i["dailydeceased"])+int(i["dailyrecover
ed"]))
            active.append(activee)
        cases=pd.DataFrame()
        cases['date']=date
        cases['confirmed']=confirmed
        cases['active']=active
        cases['death']=death
        cases['recovered']=recovered

        cdates=[]
        for d in cases['date']:
            d1=d.strip()+' 2020'
            d2=datetime.strptime(d1,'%d %B %Y').strftime('%d-%m-%Y')
            cdates.append(d2)
        cases['date']=cdates
        cases1=cases[cases['date'].isin(case_dates)]

        gbr=joblib.load('grad.sav')
        y=gbr.predict(cases1.iloc[:,[1,2,3,4]])
        output=pd.DataFrame()
        fd=[]
        for i in range(len(dates)):
             d=dates[i]
             d1=d.strftime('%d-%m-%Y')
```

```
            fd.append(str(d1))

        cases1=cases1.reset_index(drop=True)
        output['case_date']=cases1['date']
        output['confirmed']=cases1['confirmed']
        output['active']=cases1['active']
        output['death']=cases1['death']
        output['recovered']=cases1['recovered']
        output['sentiment_date']=fd
        output['positivity_score']=y
        output=output.reset_index(drop=True)
        print(output.head())
        original = pd.read_csv('predict - full.csv')
        frames = [original,output]
        final_pd = pd.DataFrame()
        final_pd = pd.concat(frames)
        print(final_pd.tail())
        final_dict={}
        final_dict['case_date']=list(final_pd['case_date'])
        final_dict['confirmed']=list(final_pd['confirmed'].astype('int64'))
        final_dict['active']=list(final_pd['active'].astype('int64'))
        final_dict['death']=list(final_pd['death'].astype('int64'))
        final_dict['recovered']=list(final_pd['recovered'].astype('int64'))
        final_dict['sentiment_date']=list(final_pd['sentiment_date'])

final_dict['positivity_score']=list(final_pd['positivity_score'].astype('float'
))

        return jsonify({'prediction': final_dict})
```

5. Live Sentiment Analysis

```
1  @app.route('/liveTweets',methods = ['GET','POST'])
2  def live_tweets():
3       MAX_TWEETS = 100
4       tweets=[]
5       if request.method == 'POST':
6           hashtag = request.args.get('hashtag')
7                auth   =   tweepy.OAuthHandler('d2PWAgPJ46WALqw92JhVRdYue',
   'WkoPasIYAwvJspylhr3bikyvkYCSgFhFbEqbqoF4Lk1cVQtKCe')
   auth.set_access_token("1010912451792195585-Oy9Xtv8kkm8kGPIwR8v6tuhPGkG4C
   Z","RtrZKT9mVhCLmGaSVN3t3f1oaudGdLeLLa98UuOKbQg3T")
```

```
8
9       api = tweepy.API(auth)
10
11              for   tweet   in   tweepy.Cursor(api.search,   q='#'+hashtag,
   rpp=100).items(MAX_TWEETS):
12          tweets.append(tweet.text)
13
14      cleaned_tweets = []
15      para=''
16      stop_words = set(stopwords.words('english'))
17      for tweet in tweets:
18         tweet=bs4.BeautifulSoup(tweet,'lxml').get_text()
19         tweet=re.sub(r'@[A-Za-z0-9]+',' ',tweet)
20         tweet=re.sub(r'https?://[A-Za-z0-9./]+',' ',tweet)
21         tweet=re.sub(r"[^a-zA-Z']",' ',tweet)
22         tweet = re.sub(r'&amp;', '&', tweet)
23         tweet=re.sub(r" +"," ",tweet)
24         tweet=tweet.strip()
25         cleaned_tweets.append(tweet)
26         para=para+tweet
27      stop_words = set(stopwords.words('english'))
28
29      para=para.replace('\n',' ')
30      para=para.replace('\r','')
31      para=bs4.BeautifulSoup(para,'lxml').get_text()
32      para=re.sub(r'@[A-Za-z0-9]+',' ',para)
33      para=re.sub(r'https?://[A-Za-z0-9./]+',' ',para)
34      para=re.sub(r"[^a-zA-Z']",' ',para)
35      para=re.sub(r" +"," ",para)
36      para=para.strip()
37      para=para.lower()
38      word_tokens = word_tokenize(para)
39      filtered_tweet = [w for w in word_tokens if not w in stop_words]
40      para=' '.join(filtered_tweet)
41
42                          from     vaderSentiment.vaderSentiment     import
   SentimentIntensityAnalyzer
43      analyser = SentimentIntensityAnalyzer()
```

```
    def sentiment_analyzer_scores(sentence):
        score = analyser.polarity_scores(sentence)
        sentiments = score['compound']
        return sentiments
    sentiment =[]
    df=pd.DataFrame(tweets,columns=['text'])
    df['cleaned']=cleaned_tweets
    for s in df['cleaned']:
        senti = sentiment_analyzer_scores(s)
        sentiment.append(senti)
    df['sentiment'] = sentiment

    def label(sentiment_value):
         if(sentiment_value>=0.05):
               return 1
         elif(sentiment_value<=-0.05):
               return -1
         else:
               return 0
    df['sentiment_label']=df['sentiment'].apply(label)
    pos=0
    neg=0
    neu=0
    count=[]
    sentimentt=[]
    text=[]
    sent=[]
    for x in df['sentiment_label']:
        if x==1:
            pos=pos+1
            sent.append('Positive')
            text.append(df['text'])
        elif x==-1:
            neg=neg+1
            sent.append('Negative')
            text.append(df['text'])
        else:
            neu=neu+1
```

```
            sent.append('Neutral')
            text.append(df['text'])
    count.append(pos)
    count.append(neg)
    count.append(neu)
    sentimentt.append('Positive')
    sentimentt.append('Negative')
    sentimentt.append('Neutral')
    df['Sentiment']=sent
    df_table=df[['text','Sentiment']]
    tweets_dict={}
    tweets_dict['para']=para
    tweets_dict['positive']=pos
    tweets_dict['negative']=neg
    tweets_dict['neutral']=neu
    tweets_dict['text']=list(df_table['text'])
    tweets_dict['Sentiment']=list(df_table['Sentiment'])
    return jsonify({'prediction': tweets_dict})
```