SENTIMENT ANALYSIS OF COVID-19 TWEETS

PROJECT REPORT

By Team Byte-Biters
V. Maheysh
Amudhini P. K.
Divyashree S.
Shalini A.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Overview:

In this project, we analyzed the sentiments in tweets and thereby predicted the feelings of people during this pandemic using machine learning. Tweets were studied to gauge the feelings of people. And we obtained the data set of twitter and fed the data set in a suitable algorithm that will give accurate output and after doing the analysis we plotted the live graph and sentiment map, thereby the user will get an idea about the current pandemic situation. Additionally, we have also done the search by state and search by keyword, from which the user can get the sentimental analysis of particular state and word respectively.

1.2 Purpose:

The main purpose of the project is to understand the feelings of people all over the world(particularly in India) during this lockdown period. The project also finds in which state the people are struggling more due to this pandemic situation which will help to solve their problems accordingly. This project will also help to understand the mentality of people in different states. From this, the government/business person can take ideal steps in their work.

2. LITERATURE SURVEY

2.1 Existing Problem

Coronavirus has grabbed the headlines and the attention of everyone in the world. It has brought the world to a standstill. People are currently practicing social distancing to avoid the spread of the novel coronavirus as a result of which most of the people are not aware of the overall emotional quotient of our country or are completely relying on the media. Completely relying on the media will not give one a clear picture of the current situation. To facilitate this purpose something user interactive should be done and it is done by a group of enthusiasts who have the same vision as ours.

The article, *Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India*[1], analyses the top sentiments prevailing in our country. It has two features, one is the word cloud of the top words used and the other is the top sentiments prevailing. Their work is user friendly and good, but has concentrated on the country as a whole and not the particular states and have limited their work only to a primeval level of sentiment analysis and hence we have proposed a better working model with additional features facilitating the better understanding of the users about the current situation in both large and small scale.

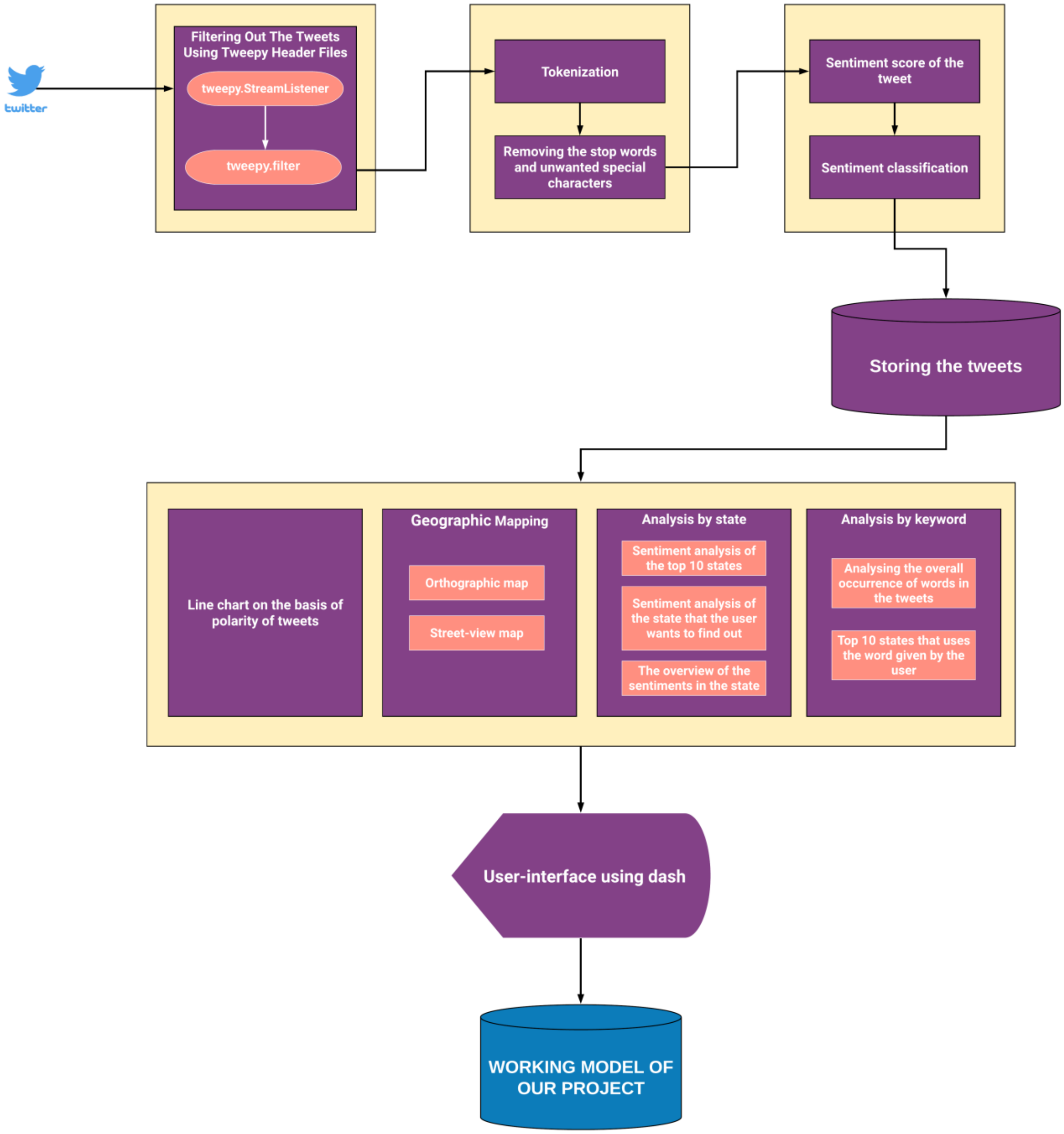
2.2 Proposed Solution

Corona-virus has grabbed the headlines and the attention of everyone in the world. Most of the people are advised to practice social distancing as a result of which people are either completely relying on the media or are not aware of the emotional situation of different sections of people in our country. Our project will help those people in giving insights about the effect of the pandemic on the emotional well-being of a person. Our project focuses more on our country's emotional well being. To make it interesting and user-interactive we have done geographical mapping of the sentiments in our country and the whole world, line chart representing the emotions of people, and analysis of sentiments by keyword and by State. Geographical mapping will help the user get an idea about the sentiments prevailing in our country, analysis by keyword feature allows the user to find the frequency of the keyword used and top states where the particular keyword is used, analysis by State allows the user to find the sentiment of any State according to the user. Our project will help the user to understand the emotional quotient of people during this pandemic.

[1] *Barkur G, Vibha, Kamath GB. Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India. Asian J Psychiatr. 2020 Apr 12;51:102089. doi: 10.1016/j.ajp.2020.102089. Epub ahead of print. PMID: 32305035; PMCID: PMC7152888.*

3. THEORITICAL ANALYSIS

3.1 Block Diagram/Flowchart


twitter
Filtering Out The Tweets Using Tweepy Header Files
tweepy.StreamListener
tweepy.filter
Tokenization
Removing the stop words and unwanted special characters
Sentiment score of the tweet
Sentiment classification
Storing the tweets
Line chart on the basis of polarity of tweets
Geographic Mapping
Orthographic map
Street-view map
Analysis by state
Sentiment analysis of the top 10 states
Sentiment analysis of the state that the user wants to find out
The overview of the sentiments in the state
Analysis by keyword
Analysing the overall occurrence of words in the tweets
Top 10 states that uses the word given by the user
User-interface using dash
WORKING MODEL OF OUR PROJECT

3.2 Hardware / Software Designing

Software requirements

To facilitate the working of our project the following software's were used:

1) TWITTER API

The keys provided by the twitter after the verification of credentials by the twitter developers is needed to collect the tweets pertaining to COVID-19 with the help of tweepy which is a python library used for streaming tweets.

2) ANACONDA CLOUD

To access the import packages required for our project

3) JUPYTER NOTEBOOK

To facilitate the process of cleaning the tweets and for visualizations. The following packages which were inbuilt in anaconda cloud were used in the jupyter interface:

i) Pandas

Pandas were used for reading of the data and formatting the specified data in data frame format.

ii)Numpy

Numpy is used for scientific computing. We have used numpy for performing some mathematical operations on arrays

iii) Nltk

We have used nltk package for tokenization and to remove the stop words

iv) Plotly

Plotly is used to make the interactive graphs in our project.

4) DASH

Dash is Python framework for building web applications. It built on top of Flask, Plotly.js, React and React js. It enables you to build dashboards using pure Python. Dash is open source, and its apps run on the web browser.

4. EXPERIMENTAL INVESTIGATIONS

Our project analyses the tweets posted by the users in their respective twitter handle and analyses the sentiment of people during this pandemic. Features like line chart, geographical mapping of the sentiments, search by keyword and search by state allow the user to learn about the emotions of the people.

Our first task was to stream tweets from twitter in real-time. For this a python library called tweepy was used. From tweepy, tweets are listened by tweepy.StreamListener and streamed with tweepy.stream. The Covid-19 related tweets was filtered with the help of tweepy.filter.

Although the tweets were streamed, not all tweets has the user-location enabled. So the tweets which has the location was taken into account. The latitude and longitude was obtained with the help of geopy.geocoders and the state was interpreted with the help of reverse_geocoder. The above task made the streaming slower, so for the smooth streaming for the live-sentiment line plot, another streaming program was used for that plot alone, which streams all the tweets of Covid-19 irrespective of whether the user-location is enabled or not. The tweet text was cleaned by removing the stop-words and unwanted special characters.

All the visualization graphs were made with plotly.py . One of the advantage in creating in Plotly was that the graphs were interactive. The source code for the respective graphs is given in Appendix - Source code.

1. **Line chart for the sentiment polarity in tweets:** A line chart is plotted in real-time for the sentiment polarity in each tweet. Also the line chart was smoothed to make it more visualizable.
2. **Orthographic map:** Sentiments in the tweets around the world are displayed in shape of the world for better visualization. The world can also be rotated which is done with the help of slider and frames.
3. **Geographic distribution map:** A street view map is made with plotly.mapbox which shows the sentiments. The interactive feature in Plotly allows the user to zoom the map and view any area he wants.
4. **Analysis by state:**

   i. **Bar graph showing the sentiment distribution:** The emotions in the top 10 states in our country is displayed. The states are chosen based on the number of tweets in that state.
   ii. **Bar graph for the top keywords in a State:** Displays the top keywords used in the tweets on any particular state the user wants.
   iii. **Radar chart:** The overview of the sentiments in the state is shown as a radar chart.

5. Analysis by keyword:
   i. **Top keywords used in the tweets:** A polar bar graph is made showing the top keywords occurred in the tweets.
   ii. **Top occurrence of a keyword in the states:** A bar graph is made showing the top 10 states where a particular keyword is used is shown. The user has the liberty to input any keyword he wants.

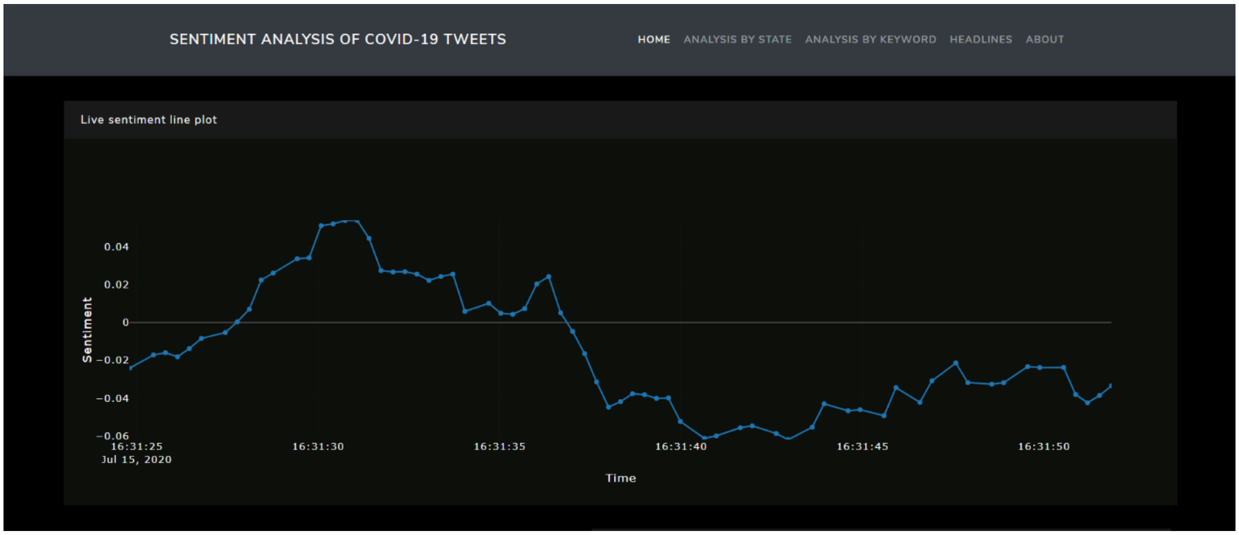
Finally, an interactive user interface is created with the help of python framework Dash created by Plotly.

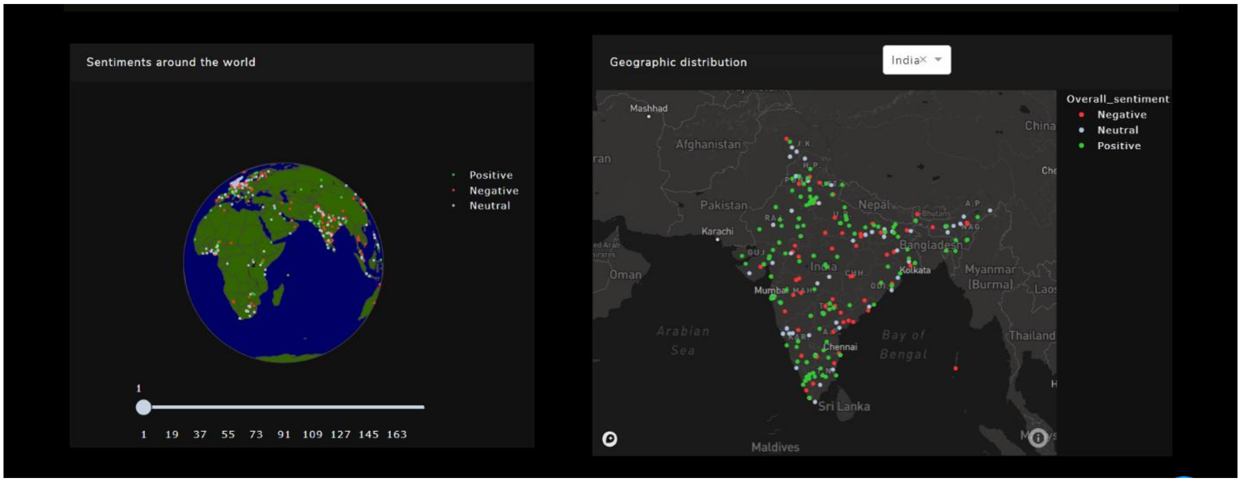
Additional ideas were implemented after this stage. Representative figures like word-clouds were also included to support the respective idea. A Headlines section is added to provide hyperlinks to official news sources and twitter handles. An About section felt necessary to explain our project and its components to our end-user.

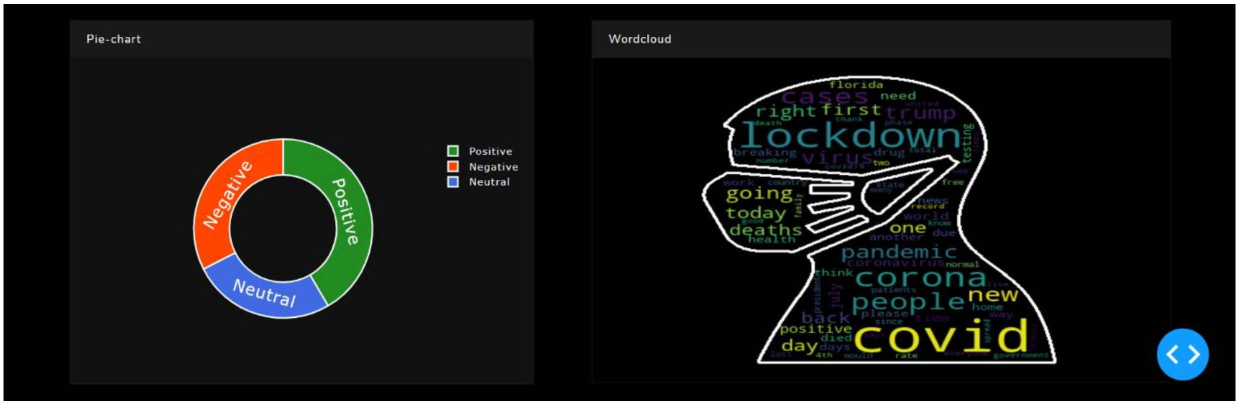
5. RESULT

The result of our project is that user/government can understand the current pandemic condition and sentiment of people in a better manner and thereby they can take proper steps to solve the problem.

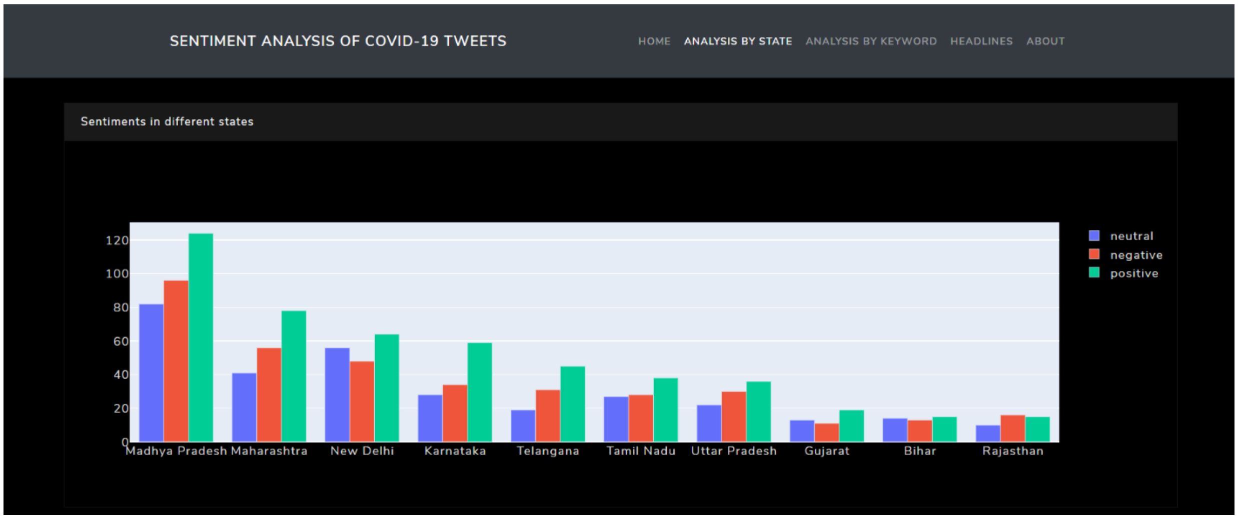
HOME :

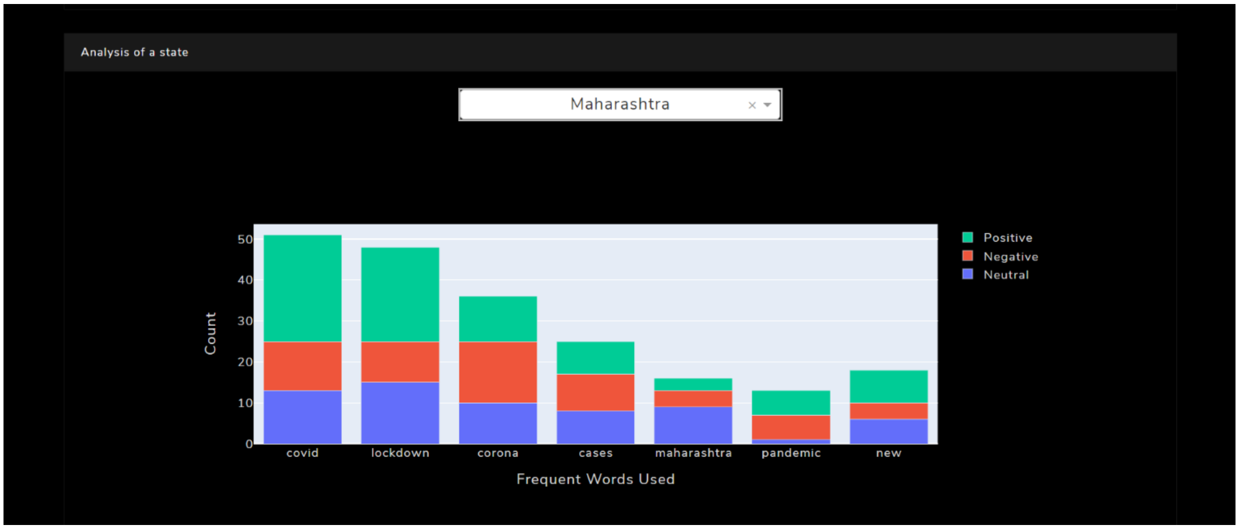

SENTIMENT ANALYSIS OF COVID-19 TWEETS
HOME
ANALYSIS BY STATE
ANALYSIS BY KEYWORD
HEADLINES
ABOUT
Live sentiment line plot
Sentiment
0.04
0.02
0
−0.02
−0.04
−0.06
16:31:25
Jul 15, 2020
16:31:30
16:31:35
16:31:40
16:31:45
16:31:50
Time

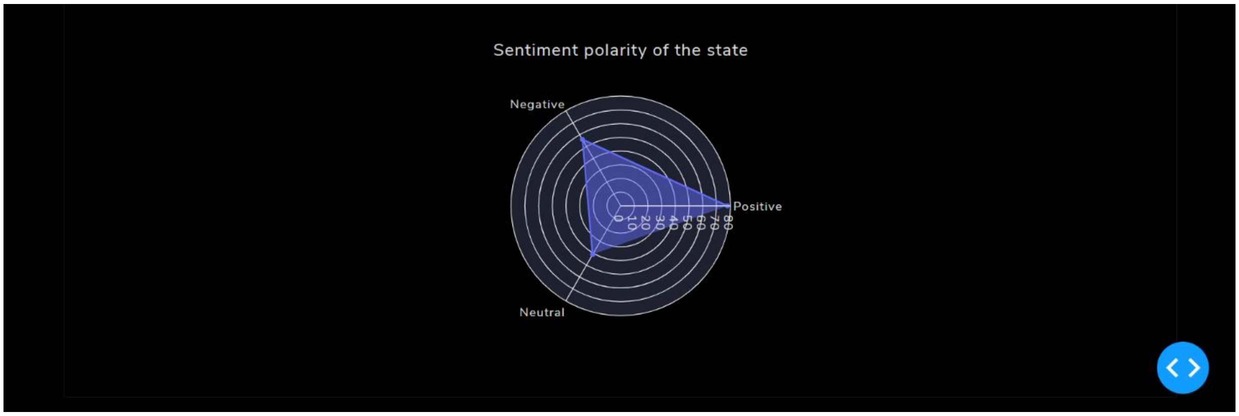

Sentiments around the world
Positive
Negative
Neutral
1
1 19 37 55 73 91 109 127 145 163
Geographic distribution
India×
Overall_sentiment
Negative
Neutral
Positive
Mashhad
Afghanistan
China
Pakistan
Karachi
Nepal
Bhutan
Bangladesh
India
Kolkata
Oman
Mumbai
Myanmar (Burma)
Laos
Arabian Sea
Chennai
Bay of Bengal
Thailand
Sri Lanka
Maldives


Pie-chart
Negative
Positive
Neutral
Positive
Negative
Neutral
Wordcloud
florida
cases
need
right
first
trump
lockdown
testing
breaking
virus
drug
going
today
deaths
one
health
pandemic
corona
new
people
back
positive
day
covid

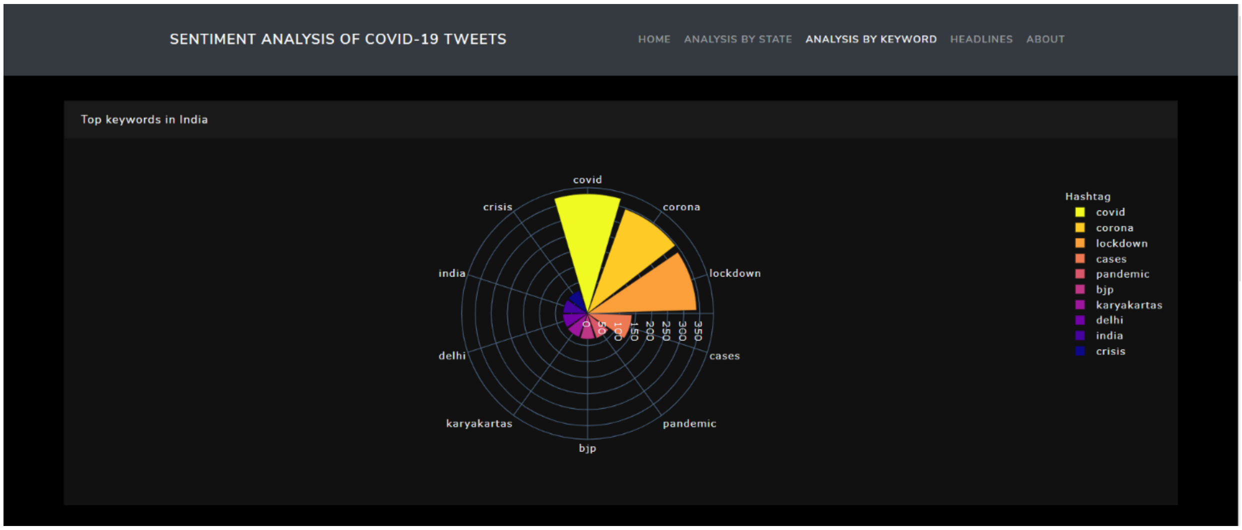
ANALYSIS BY STATE:

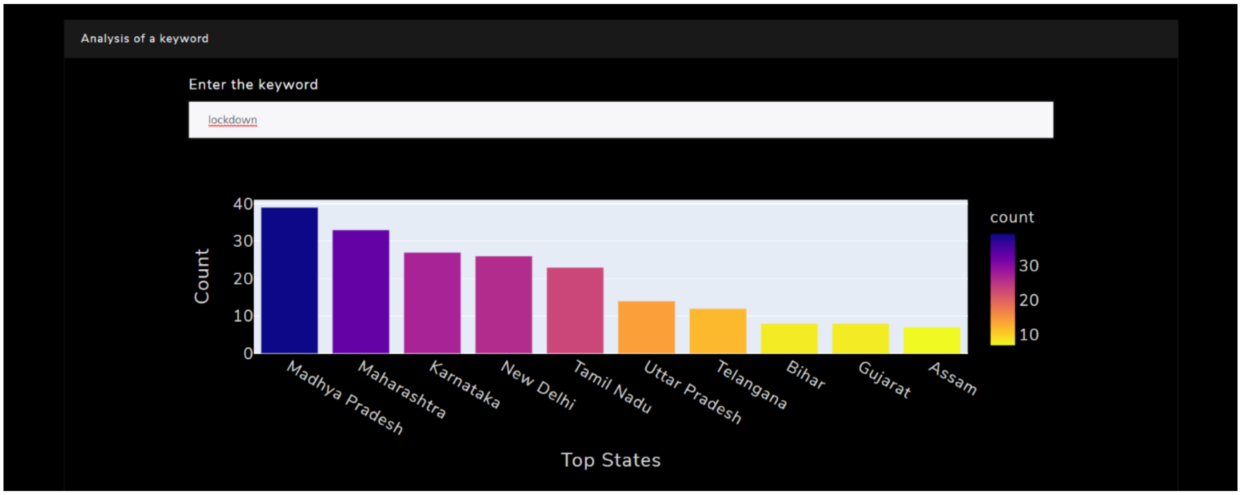






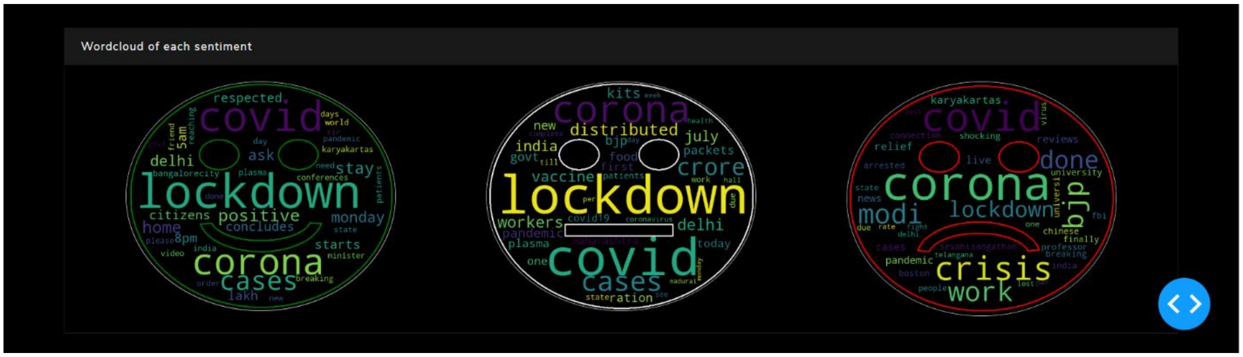
ANALYSIS BY KEYWORD:



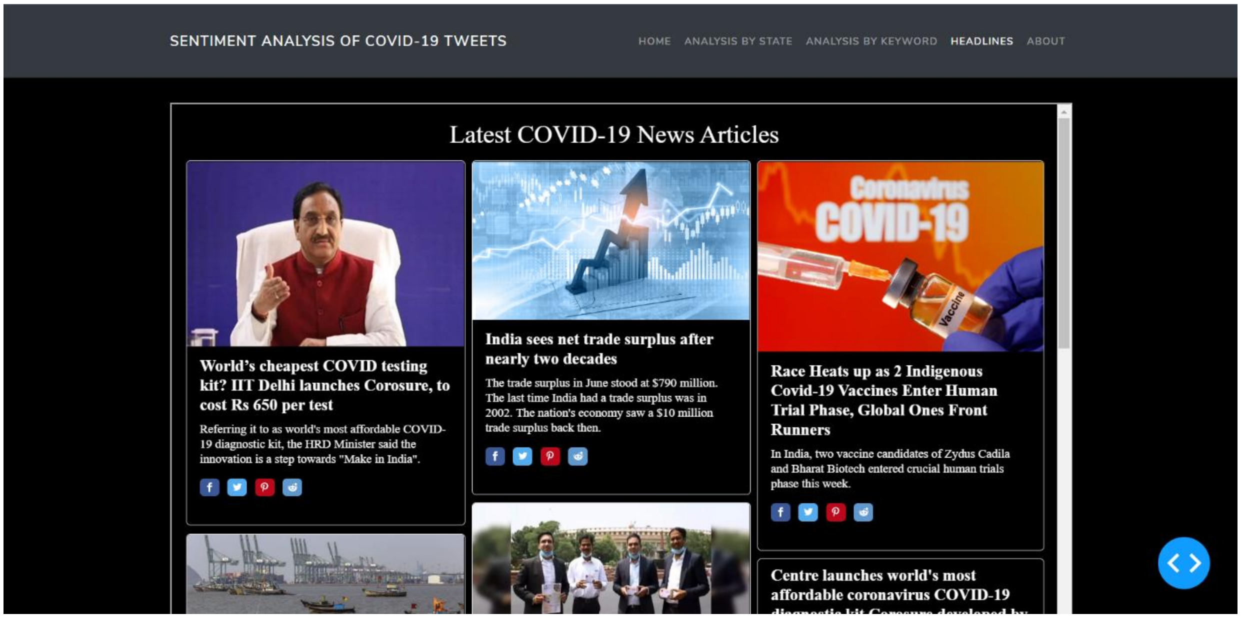


Enter the keyword

lockdown



HEADLINES:



SENTIMENT ANALYSIS OF COVID-19 TWEETS

Latest COVID-19 News Articles

World's cheapest COVID testing kit? IIT Delhi launches Corosure, to cost Rs 650 per test

Referring it to as world's most affordable COVID-19 diagnostic kit, the HRD Minister said the innovation is a step towards "Make in India".

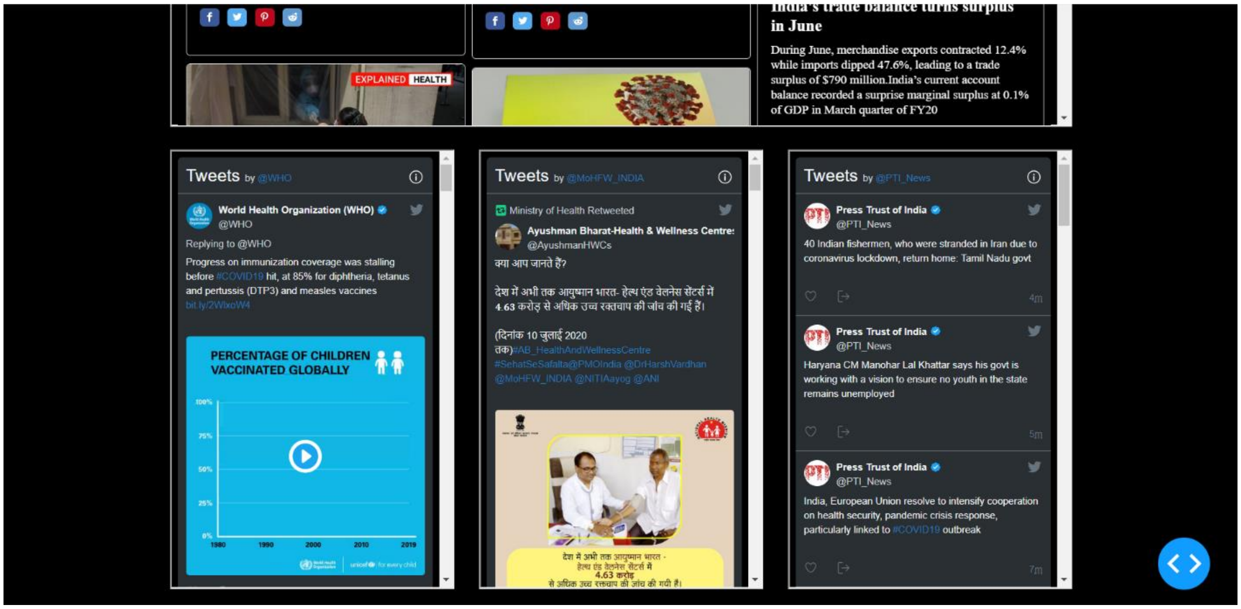
India sees net trade surplus after nearly two decades

The trade surplus in June stood at $790 million. The last time India had a trade surplus was in 2002. The nation's economy saw a $10 million trade surplus back then.

Race Heats up as 2 Indigenous Covid-19 Vaccines Enter Human Trial Phase, Global Ones Front Runners

In India, two vaccine candidates of Zydus Cadila and Bharat Biotech entered crucial human trials phase this week.

Centre launches world's most affordable coronavirus COVID-19 diagnostic kit Corosure developed by



India's trade balance turns surplus in June

During June, merchandise exports contracted 12.4% while imports dipped 47.6%, leading to a trade surplus of $790 million.India's current account balance recorded a surprise marginal surplus at 0.1% of GDP in March quarter of FY20

Tweets by @WHO

World Health Organization (WHO)
@WHO
Replying to @WHO
Progress on immunization coverage was stalling before #COVID19 hit, at 85% for diphtheria, tetanus and pertussis (DTP3) and measles vaccines bit.ly/2WlxoW4

Tweets by @MoHFW_INDIA

Ministry of Health Retweeted
Ayushman Bharat-Health & Wellness Centre:
@AyushmanHWCs
क्या आप जानते हैं?

देश में अभी तक आयुष्मान भारत- हेल्थ एंड वेलनेस सेंटर्स में 4.63 करोड़ से अधिक उच्च रक्तचाप की जांच की गई हैं।

(दिनांक 10 जुलाई 2020 तक)#AB_HealthAndWellnessCentre #SehatSeSafalta@PMOIndia @DrHarshVardhan @MoHFW_INDIA @NITIAayog @ANI

देश में अभी तक आयुष्मान भारत -
हेल्थ एंड वेलनेस सेंटर्स में
4.63 करोड़
से अधिक उच्च रक्तचाप की जांच की गयी है।

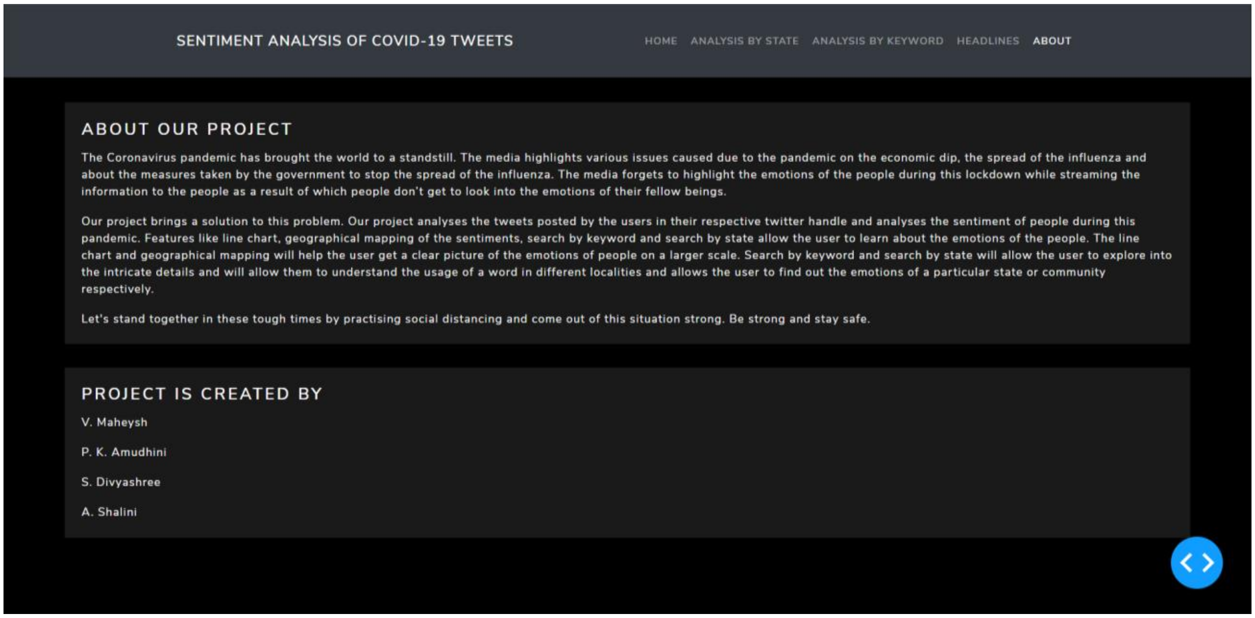
Tweets by @PTI_News

Press Trust of India
@PTI_News
40 Indian fishermen, who were stranded in Iran due to coronavirus lockdown, return home: Tamil Nadu govt
4m

Press Trust of India
@PTI_News
Haryana CM Manohar Lal Khattar says his govt is working with a vision to ensure no youth in the state remains unemployed
5m

Press Trust of India
@PTI_News
India, European Union resolve to intensify cooperation on health security, pandemic crisis response, particularly linked to #COVID19 outbreak
7m

ABOUT:



ABOUT OUR PROJECT

The Coronavirus pandemic has brought the world to a standstill. The media highlights various issues caused due to the pandemic on the economic dip, the spread of the influenza and about the measures taken by the government to stop the spread of the influenza. The media forgets to highlight the emotions of the people during this lockdown while streaming the information to the people as a result of which people don't get to look into the emotions of their fellow beings.

Our project brings a solution to this problem. Our project analyses the tweets posted by the users in their respective twitter handle and analyses the sentiment of people during this pandemic. Features like line chart, geographical mapping of the sentiments, search by keyword and search by state allow the user to learn about the emotions of the people. The line chart and geographical mapping will help the user get a clear picture of the emotions of people on a larger scale. Search by keyword and search by state will allow the user to explore into the intricate details and will allow them to understand the usage of a word in different localities and allows the user to find out the emotions of a particular state or community respectively.

Let's stand together in these tough times by practising social distancing and come out of this situation strong. Be strong and stay safe.

PROJECT IS CREATED BY

V. Maheysh

P. K. Amudhini

S. Divyashree

A. Shalini

6. ADVANTAGES & DISADVANTAGES

6.1 Advantages:

1) Our project will help the user to understand the emotions of the people during this pandemic in a better way by analyzing the line chart and the geographical mapping

2) The user can also understand a particular state of his/her own interest to see the effect of the pandemic on people and also can find out the usage of a particular keyword used by people and can analyze the top states where the usage of the word is more.

3) Our project brings doesn’t narrow down to a particular stream of opinions, but rather it focuses on the opinions of different sections and the above process is facilitated by the usage of twitter API to gather the tweets .

6.2 Disadvantages:

1) As our project focuses on the sentimental analysis of our country, the tweets are narrowed down to the tweets of our country as a result of which the number of tweets pertaining to a particular state is less.

7. APPLICATIONS

Our project helps the user/government to find in which state the sentiment is negative, positive and neutral and thereby the higher authorities can understand this situation more clearly. Businesses meanwhile can use this tool to understand the sentiment of people and interpret in what way the sentiment of the people would affect their business. This project is a great tool to solve the problem that is happening during this lock down period.

8. CONCLUSION

From running the application, terms related to corona such as 'corona','covid-19' fetched a net negative sentiment in general as observed the sentiment line plot. But the term 'lockdown' specifically had a more positive sentiment attached to it as observed from the 'analysis of a state' plot. So as a conclusion to the problem statement provided, it can be inferred that the lockdown had a positive impact on the people, because responsible citizens they knew that it was the only way to 'flatten the curve' and to decrease the spread and eventually stop it.

9. FUTURE SCOPE

Our project is easily scalable. For the API to extract tweets, a standard API is being used. This can be scaled up by using the Premium or Enterprise API's when funded for more volume of tweets and various other functionalities which would make the data analytic process easier.

Our search term for keywords by which we filter corona related tweets can be easily modified to get generic tweets or from a specific topic also. So when the pandemic finally ends and coronavirus related real time tweets are non-existent, our application can be converted into one of a general tweet sentiment analysis. A similar change of keyword helps in making the newsfeed to display the specific genre of content required.

When deployment to cloud is done, the storage of the database can be increased based on our needs so that a longer history of tweets can be extracted anytime and displayed for analysis.

10. BIBILOGRAPHY

https://towardsdatascience.com/real-time-twitter-sentiment-analysis-for-brand-improvement-and-topic-tracking-chapter-1-3-e02f7652d8ff

https://plotly.com/python/

https://dash.plotly.com/introduction

https://www.kaggle.com/smid80/coronavirus-covid19-tweets

https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7152888/

11. APPENDIX

A. Source code

Line chart for the sentiment polarity in tweets

```
conn = sqlite3.connect('twitter.db')
c = conn.cursor()

df = pd.read_sql("SELECT * FROM sentiment WHERE tweet LIKE '%covid%'
ORDER BY unix DESC LIMIT 1000", conn)

df.sort_values('unix', inplace=True)

df['sentiment_smoothed'] =
df['sentiment'].rolling(int(len(df)/5)).mean()

df['date'] = pd.to_datetime(df['unix'],unit='ms')
df.set_index('date', inplace=True)
df = df_resample_sizes(df)
df.dropna(inplace=True)

X = df.index
Y = df.sentiment_smoothed

data = go.Scatter(x=X, y=Y, name='Scatter', mode= 'lines+markers',)

```

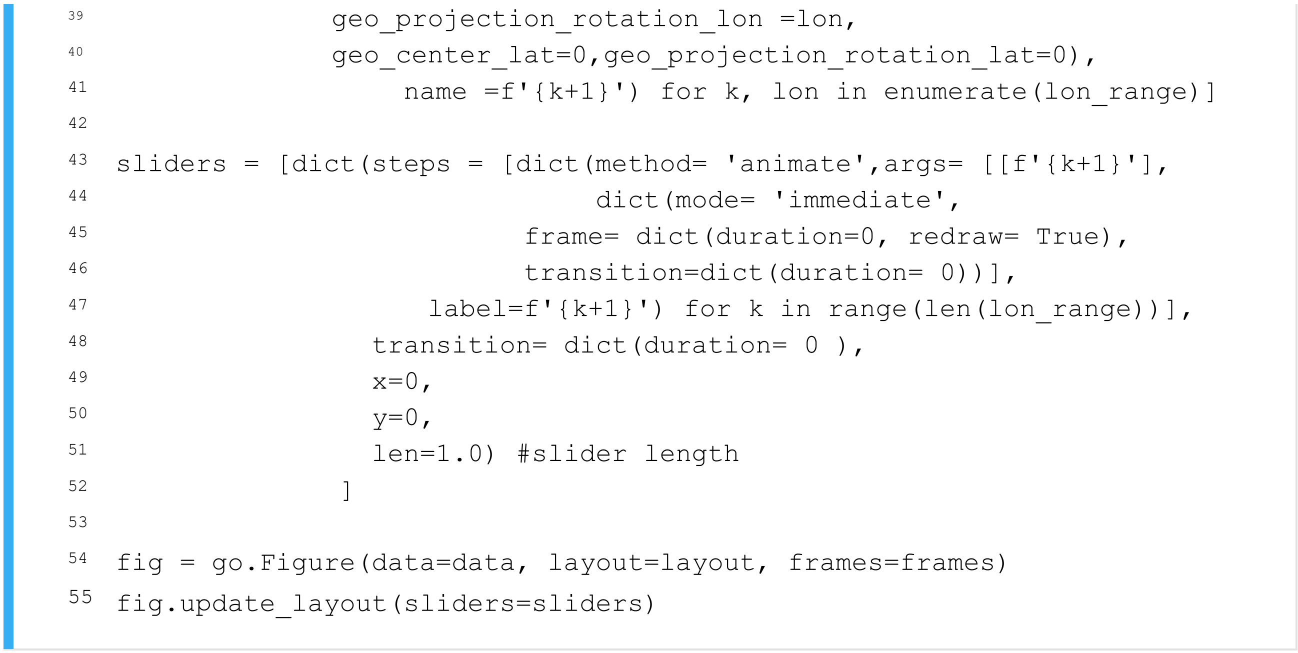
```
figure= {'data': [data],
'layout':go.Layout(xaxis=dict(range=[min(X),max(X)],title='Time'),
yaxis=dict(range=[min(Y),max(Y)],title='Sentiment'),
font={'color':'#FFFFFF'}, plot_bgcolor = '#0C0F0A',
paper_bgcolor='#0C0F0A', )}
```

Orthographic map

```
data=[go.Scattergeo(
            lat=df1[df1['Overall_sentiment']=='Positive']['Latitude'],
            lon=df1[df1['Overall_sentiment']=='Positive']['Longitude'],
            mode='markers',
            marker_color='rgb(128, 255, 0)',
            marker_size=3,
            name='Positive'),
     go.Scattergeo(
            lat=df1[df1['Overall_sentiment']=='Negative']['Latitude'],
            lon=df1[df1['Overall_sentiment']=='Negative']['Longitude'],
            mode='markers',
            marker_color='rgb(255, 51, 51)',
            marker_size=3,
            name='Negative'),
    go.Scattergeo(
            lat=df1[df1['Overall_sentiment']=='Neutral']['Latitude'],
            lon=df1[df1['Overall_sentiment']=='Neutral']['Longitude'],
            mode='markers',
            marker_color='rgb(174, 179, 179)',
            marker_size=3,
            name='Neutral')]

layout =go.Layout(template='plotly_dark',
        geo=go.layout.Geo(
            showland = True,
            showcountries = True,
            showocean = True,
            countrywidth = 0.5,
            landcolor = 'rgb(51, 102, 0)',
            oceancolor = 'rgb(0, 0, 102)',
            projection_type='orthographic',
            center_lon=50,
            center_lat=0,
            projection_rotation_lon=50
        ))
    lon_range = np.arange(-180, 180, 2)

frames =[go.Frame(layout=go.Layout(geo_center_lon=lon,
```

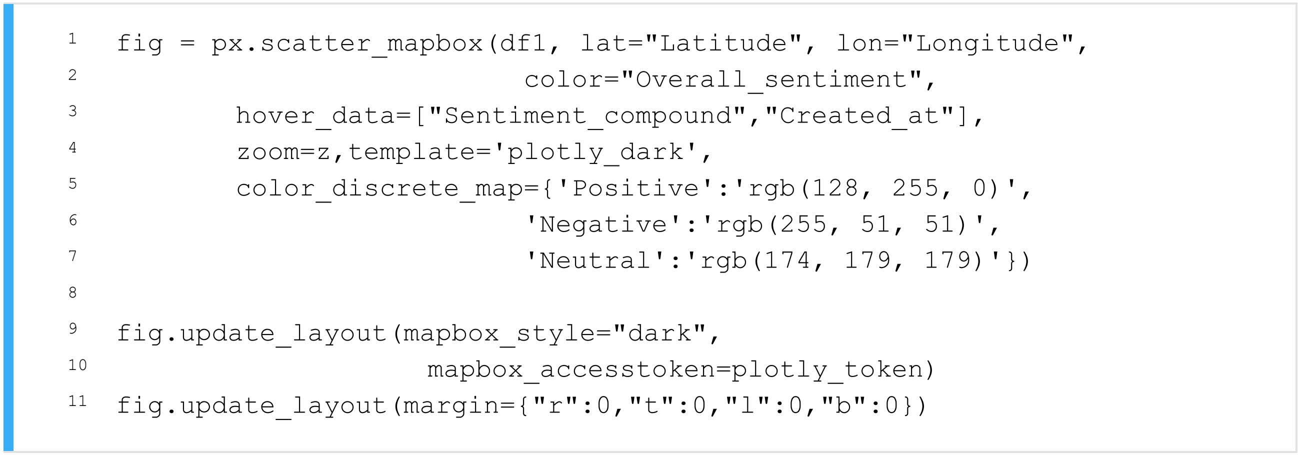


```
              geo_projection_rotation_lon =lon,
              geo_center_lat=0,geo_projection_rotation_lat=0),
                  name =f'{k+1}') for k, lon in enumerate(lon_range)]

sliders = [dict(steps = [dict(method= 'animate',args= [[f'{k+1}'],
                              dict(mode= 'immediate',
                          frame= dict(duration=0, redraw= True),
                          transition=dict(duration= 0))],
                    label=f'{k+1}') for k in range(len(lon_range))],
                transition= dict(duration= 0 ),
                x=0,
                y=0,
                len=1.0) #slider length
              ]

fig = go.Figure(data=data, layout=layout, frames=frames)
fig.update_layout(sliders=sliders)
```

Geographic distribution map:

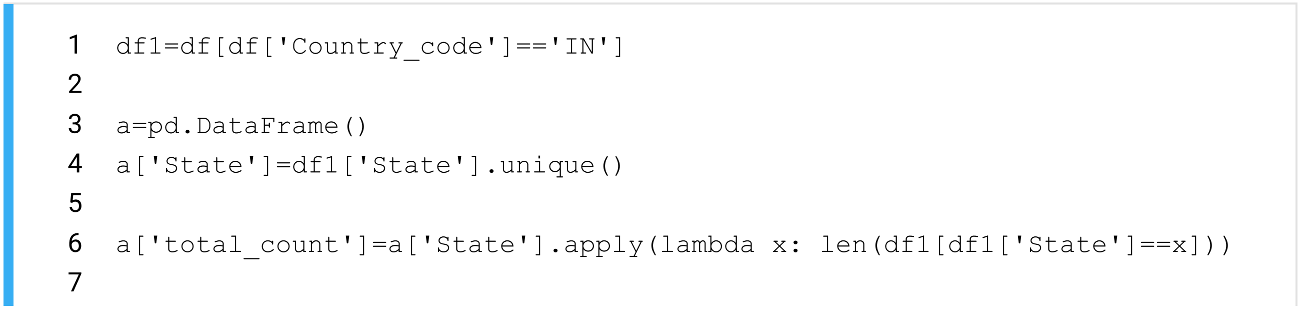


```
fig = px.scatter_mapbox(df1, lat="Latitude", lon="Longitude",
                          color="Overall_sentiment",
        hover_data=["Sentiment_compound","Created_at"],
        zoom=z,template='plotly_dark',
        color_discrete_map={'Positive':'rgb(128, 255, 0)',
                          'Negative':'rgb(255, 51, 51)',
                          'Neutral':'rgb(174, 179, 179)'})

fig.update_layout(mapbox_style="dark",
                    mapbox_accesstoken=plotly_token)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
```

Analysis by state:

i) Bar graph showing the sentiment distribution



```
df1=df[df['Country_code']=='IN']

a=pd.DataFrame()
a['State']=df1['State'].unique()

a['total_count']=a['State'].apply(lambda x: len(df1[df1['State']==x]))

```

```
a['positive_count']=a['State'].apply(lambda x: len(df1[df1['State']==x]['State'][df1[df1['State']==x]['Overall_sentiment']=='Positive']))

a['negative_count']=a['State'].apply(lambda x: len(df1[df1['State']==x]['State'][df1[df1['State']==x]['Overall_sentiment']=='Negative']))

a['neutral_count']=a['State'].apply(lambda x: len(df1[df1['State']==x]['State'][df1[df1['State']==x]['Overall_sentiment']=='Neutral']))

a.sort_values(by=['total_count'], inplace=True,ascending=False)

trace3 = go.Bar(x =a.State.head(10), y = a.positive_count.head(10), name='positive')
trace2 = go.Bar(x= a.State.head(10), y = a.negative_count.head(10), name ='negative')
trace1 = go.Bar(x = a.State.head(10), y = a.neutral_count.head(10), name ='neutral' )
data = [trace1, trace2, trace3]

layout = go.Layout(barmode = 'group',paper_bgcolor="Black",
               font=dict(family="Nunito Sans",size=15,color="#DDDDDD"))
fig= go.Figure(data = data, layout = layout)
```

ii) Bar graph for the top keywords in a State

```
df1=df[df['State']==m]
f=return_tweet_words(df1)
sort_orders = sorted(f.items(), key=lambda x: x[1], reverse=True)
x=[];y=[]
for j in range(0,10):
 x.append(sort_orders[j][0])
 y.append(sort_orders[j][1])

y1=[];y2=[];y3=[];x1=[]

for k in range(0,7):
 p=0;n=0;l=0
      for j in df1.index:
       if df1['Overall_sentiment'][j]=='Positive' and str(df1['Tweet_text'][j]).lower().count(x[k])>0:
                p+=str(df1['Tweet_text'][j]).lower().count(x[k])

       elif df1['Overall_sentiment'][j]=='Negative'and
str(df1['Tweet_text'][j]).lower().count(x[k])>0:
                n+=str(df1['Tweet_text'][j]).lower().count(x[k])

       elif df1['Overall_sentiment'][j]=='Neutral'and
str(df1['Tweet_text'][j]).lower().count(x[k])>0:
                l+=str(df1['Tweet_text'][j]).lower().count(x[k])

 if(p>0 or n>0 or l>0):
            y1.append(p)
            y2.append(n)
            y3.append(l)
            x1.append(x[k])

fig = go.Figure(go.Bar(x=x1, y=y3, name='Neutral'))
fig.add_trace(go.Bar(x=x1, y=y2, name='Negative'))
fig.add_trace(go.Bar(x=x1, y=y1, name='Positive'))
fig.update_layout(barmode='stack')
fig.update_layout(xaxis_title="Frequent Words Used",
yaxis_title="Count",
                      font=dict(family="Nunito Sans", size=15,
                    color="#DDDDDD"),paper_bgcolor="Black")
```

iii) Radar chart:

```
p=len(df1[df1['Overall_sentiment']=='Positive'])
n=len(df1[df1['Overall_sentiment']=='Negative'])
l=len(df1[df1['Overall_sentiment']=='Neutral'])


fig1 = go.Figure(data=go.Scatterpolar(
r=[p,n,l],theta=['Positive','Negative','Neutral'],
fill='toself')
    fig1.update_layout(paper_bgcolor="Black",font=dict(family="Nunito
Sans",size=15,color="#DDDDDD"),
polar=dict(bgcolor='#1e2130',
radialaxis=dict(visible=True,)),showlegend=False)

fig1.update_layout(title_text=
'Sentiment polarity of the state', title_x=0.5)
```

Analysis by Keywords

i) Top keywords used in the tweets

```
fd =return_tweet_words(df[df["Country_code"]=='IN'])
d = pd.DataFrame({'Hashtag': list(fd.keys()),
                  'Count' : list(fd.values())})
d = d.nlargest(columns = 'Count', n = 10)
fig= px.bar_polar(d, r='Count', theta='Hashtag',
              hover_data=['Hashtag','Count'],
              color='Hashtag', template="plotly_dark",
                color_discrete_sequence= px.colors.sequential.Plasma_r)
```

ii) Top occurrence of a keyword in the states

```
def search(t):
  d1=str(t).count(h.lower())
  return d1

df1=df[df["Country_code"]=='IN']
df1["count"]=df1["Tweet_text"].apply(search)
df2=df1[df1['count']>0]

df3=df2.drop(['User_Id','Created_at','Country_code','Latitude','Longitude','Tweet_text', 'Sentiment_compound', 'Overall_sentiment'],axis=1)
df3=df3.groupby('State').sum()
df3.sort_values(by=['count'], inplace=True,ascending=False)
df3=df3.rename_axis('index').reset_index()
df3.rename(columns={"index":"state"})

fig5=px.bar(df3.head(10),x='index',y='count',
color='count',color_continuous_scale=px.colors.sequential.Plasma_r,
                labels={'pop':'count'},height=400)
```