

IBM HACK CHALLENGE 2020

Sentiment analysis of covid-19 tweets

INTRODUCTION

Our sentiment analysis model facilitates a better understanding of the nation's reaction towards this global pandemic. The dashboard generated will help the government and other institutions to get a hold of the current situation. Also, our predictive analysis of relevant #tags on Twitter will boost up businesses amidst the nationwide lockdown.

Purpose:

Our model aims to provide a better understanding of the public sentiments regarding the Covid-19 situation and also provide a brief visualisation about their thoughts which can help the government to predict their reactions on the various laws adopted during the lockdown period. Our model aims to foster a positive attitude among the public regarding COVID-19 and help organisations respond in time.

LITERATURE SURVEY

Existing problem:

The Covid-19 endangers our physical health indeed, but alongside, social distancing also threatens our emotional stability. Our government is facing a problem to get the nation's reaction towards the global pandemic and their views on the various actions adopted in extension of the lockdown which has made things more complicated as they are not able to make a decision in public welfare.

Proposed Solution:

Our model on sentiment analysis of covid-19 tweets using the popular indian hashtags can help to solve the government problems. We are showing the people reactions on this pandemic situation using various graphs, plots and models. Visualization of data on our dashboard will dynamically represent the public sentiments and help us be more specific in understanding the trends. Twitter hosts abundant user tweets, which closely reflect the public's reactions towards this pandemic that's why we are collecting all the relevant tweets with hashtags to predict public sentiments.

THEORITICAL ANALYSIS

Technical tools We used python tools like tweepy and textblob to fetch the tweets from Twitter Developer Account. The modules used in our analysis model are:

Tweepy: Tweepy is the python Client for the official Twitter API.

Textblob: It is a python library and offers a simple API to access its methods and perform basic NLP tasks. TextBlob aims to provide access to common text-processing operations through a familiar interface. You can treat [TextBlob](#) objects as if they were Python strings that learned how to do Natural Language Processing.

Twitter Developer Account: The developer portal is a self-serve user interface within developer.twitter.com where developers can manage their API access and [Twitter apps](#). We have used developer account to fetch all the tweets with Covid-19 indian #tags to analyze public sentiment.

IBM Cloud: For deployment

IBM Watson Studio: For analysis, modeling and visualisation.

Numpy: It is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. We used this library to train and test our Linear Regression model.

Scikit-Learn: This is one of the most popular ML libraries for classical ML algorithms. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML. We used this library to form the Linear Regression model.

Pandas: It is a popular Python library for data analysis. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data. We used pandas for Sentiment analysis our dataset and form that Dataframe by reading the CSV file.

Matplotlib: It is a very popular Python library for data visualization. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc. We used pyplot module of matplotlib to plot visualize our dataset by plotting Piecharts, Histograms, Subplots to depict the sentiment analysis graphically.

AI Experiment Model: We gave used AI Experiment model using the csv file we have

made which depicts the overall mechanism of analysis and visualisation and also exhibit the step by step process used to select a specified model.

Linear Regression Model: Linear regression is a statistical algorithm used to predict a Y value, given X features. Using machine learning, the data sets are examined to show a relationship. The relationships are then placed along the X/Y axis, with a straight line running through them to predict further relationships.

Flask: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. We have used flask to make our dashboard dynamic as it fetches tweets from twitter and updates the ui on the dynamic visualisation.

Streaming with Tweepy: The Twitter streaming API is used to download twitter messages in real time. It is useful for obtaining a high volume of tweets, or for creating a live feed using a site stream or user stream.

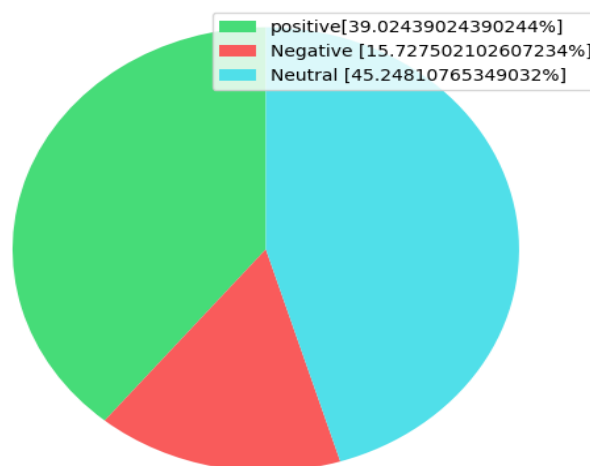
EXPERIMENTAL INVESTIGATIONS

We used various popular ML libraries for data visualisation like - pandas, numpy, matplotlib and scikit-learn. The various visualisation charts and model we developed are:

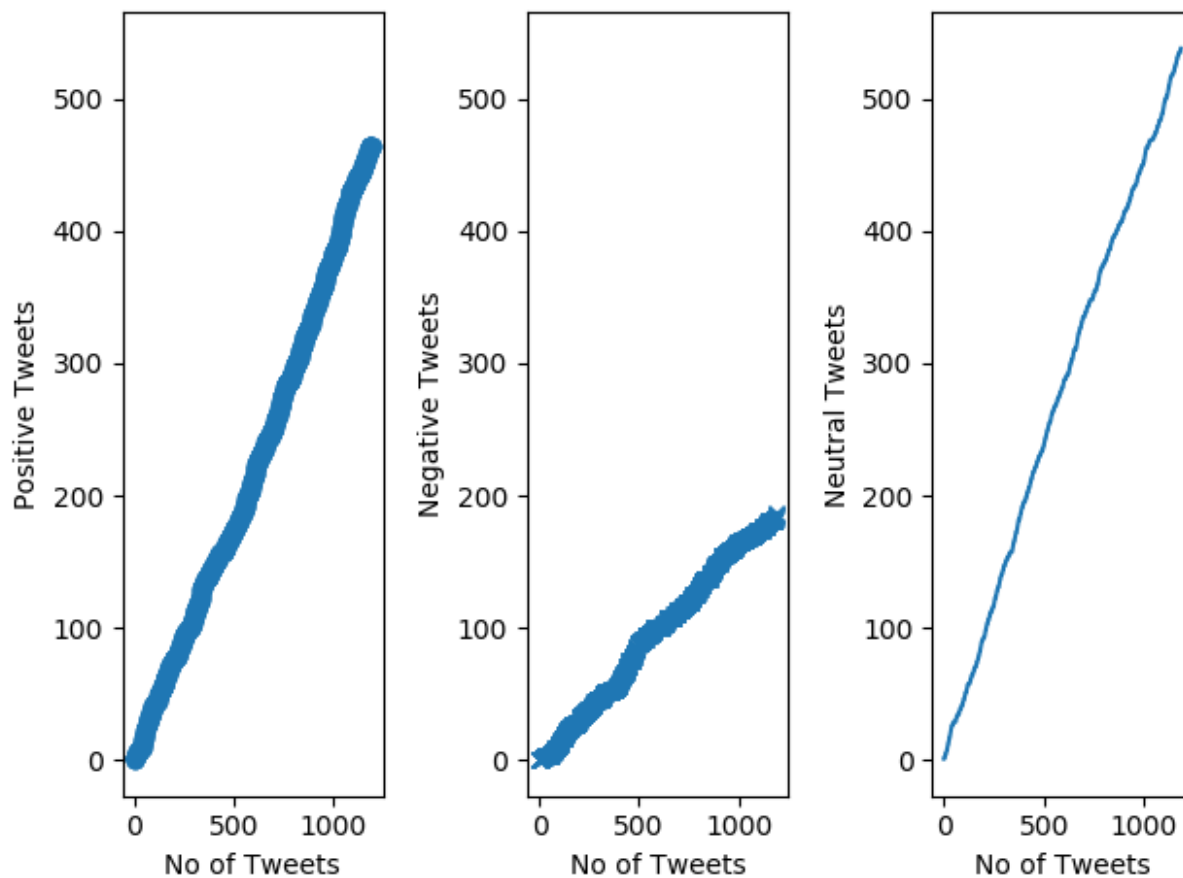
1. PieChart

We used the positive, negative and neutral tweet percentages to develop a Pie chart which shows here that the neutral tweets are more as compared to positive and negative tweets.

How people are reacting on 1189 Tweets as Positive,Negative,Neutral

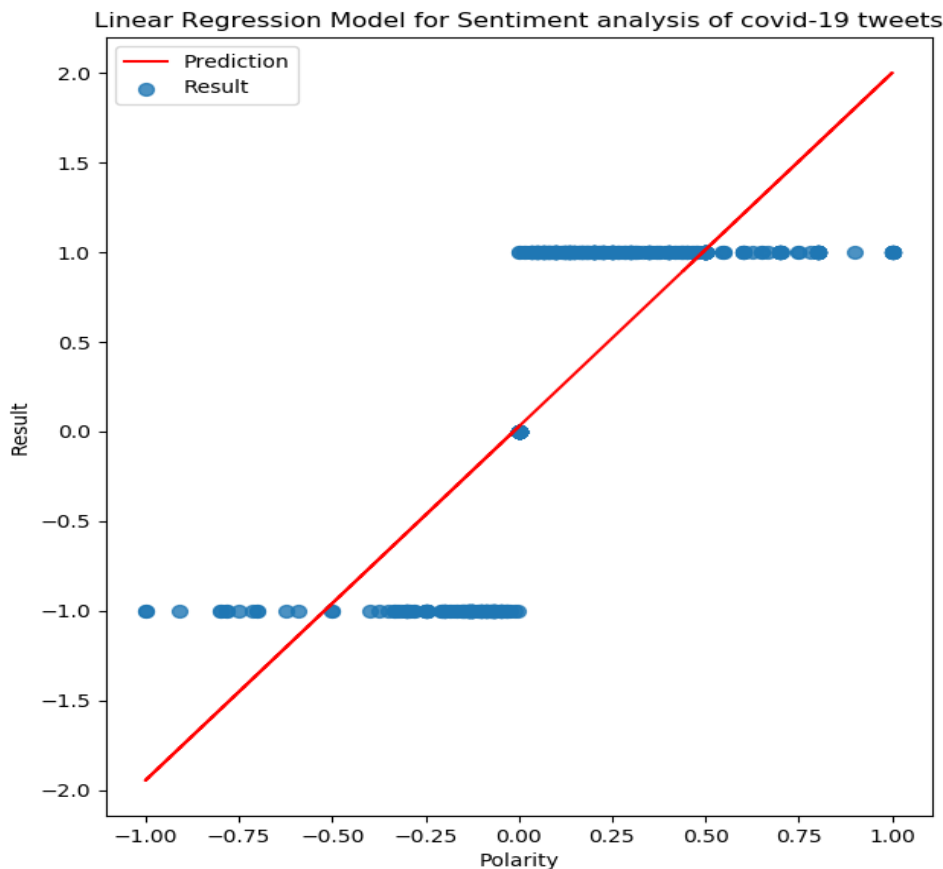


2. Subplots With the help of the libraries like matplotlib, numpy and pandas we have developed subplots



3. Linear Regression Model

We have used ML library Matplotlib to plot the linear regression model. The blue points on the graph denote the results which are positive=+1, neutral=0, negative=-1 and the red line shows the prediction line for the training values.

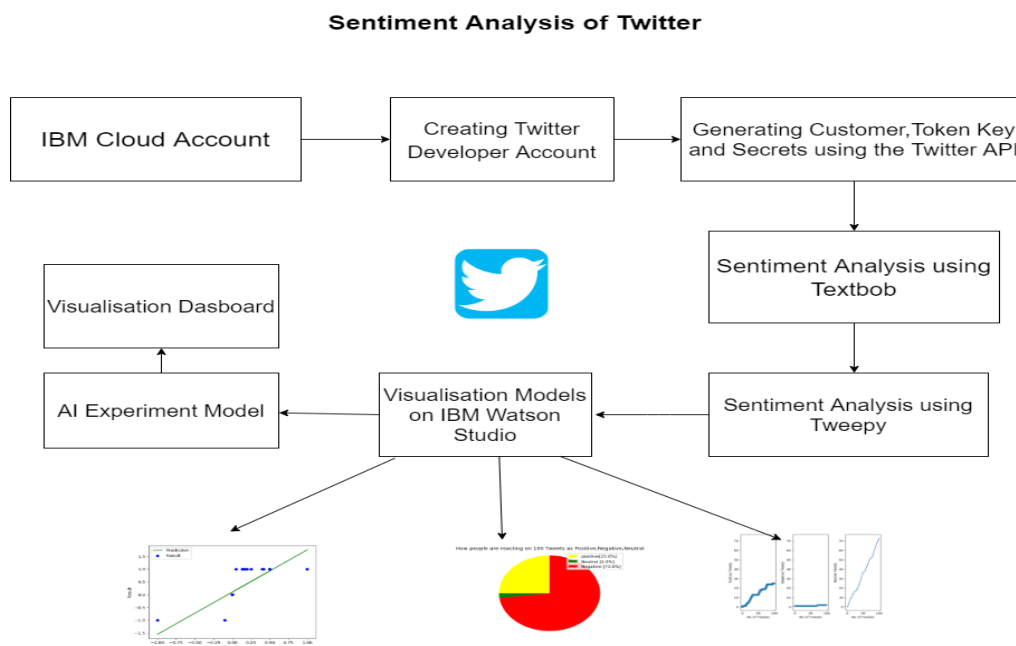


FLOWCHART-

Sentiment Analysis of Covid-19 Tweets model workflow-

1. All the team members had made the IBM Cloud account using the Email id provided by the smartinternz team . Logging to the accounts, from the catalog section we have added the IBM Watson Studio which is used for the visualisation of our model.
2. To fetch all the Tweets directly from twitter we have made the Twitter Developer Account which provided us with the api key and the necessary credentials used.
3. Using the Tweepy and Textblob we were successful to dynamically fetch the tweets from Twitter developer account .
4. In the python code we have used the indian hashtags and set the location to india which fetched the indian tweets associated to the hashtags provided.
5. Through python code we were able to construct the csv file with relevant data used for the model.

6. Using IBM Studio we made the Linear Regression model with the csv file we obtained.
7. The Sentiment Analysis visualisations included Piecharts, Histograms and Bar graph .
8. We have trained the AI -Experiment Model also using the csv file which included all the necessary pipelines and the processflow of the model.
9. Using Flask and Stream api we have done the dynamic visualisation of the model .



RESULT

Through the use of natural language processing and text analysis we have identified and extracted subjective information by reading through comments and reviews on Twitter. The first case of Covid-19 in India was reported on 30 January 2020. As after that period the real fight started in March when the first lockdown have started and the no of cases kept on increasing after that, looking at the economic crisis first unlock started from 15th June which has made the conditions more vulnerable as per the Health Ministry confirmed a total tally of 5,85,493 positive cases till date. These statistics leave an imprint on our minds in ways we are unaware of. We keep asking ourselves how we have been feeling amidst this time which is endangering our physical health alongside causing isolation and hence posing a threat to our mental health too. Our project realises how crucial it is to understand sentiments of the Citizens of India under this global pandemic. We aim to help public health officials slow the spread of the

disease by providing them with the data and analytics we will acquire. Our model's analysis will give a foresight by providing an indication of how well prepared people are for the coming changes by categorising tweets on the basis of positive , negative sentiments and neutral.

ADVANTAGES AND DISADVANTAGES

- This model will help the government to understand the sentiments of people.
- This will give a visualisation of public sentiments on the extended lockdown.
- This can create awareness in government about the public interest so that they can take decisions in their favour.
- This can create a good impact on our economy as it will highlight the all the bad points of government in front of everyone.
- This can be dangerous also because people use abusive language in tweets to criticize the government that creates an inferior environment.

APPLICATIONS

- This model provides you with a perfect slice and dice of a high volume of complex live streaming data to create interactive analytics which allows you to instantly explore the desired information in the form of live visualization dashboards covering the detailed analysis of the entire data-set.
- Dashboards continuously fetch the new data to update its graphs and summaries.
- Optimal service can be provided to the users with some recommendation system utilizing the sentiment analysis technique.
- Our model can adequately understand the critical sentiment, belief and sensibility and present them with valuable insights to promptly arrive at the appropriate measures.
- This model monitors the real-time Twitter feed for coronavirus-related tweets. Using Flask and streaming api using tweets we are presenting the dynamic representation of the public sentiments which is used to trace and filter out the misinformation floating and provide us with accurate data streaming to analyze and visualize it on live dashboards.

CONCLUSION

The Covid-19 has created disastrous impact on the public health and also have created a emotional instability because of the extending lockdowns. Thus, it is crucial to understand public sentiments under COVID-19. We are using Twitter's data extraction and data mining which will significantly help us to know the general sentiment people have towards the pandemic. Our model can adequately understand the critical sentiment, belief and sensibility and present them with valuable insights to promptly arrive at the appropriate measures. We use dashboards which continuously fetch the new data to update its graphs and summaries. So, service providers can grasp the response of the users on their services, and manufacturers can use it for marketing research. Also, optimal service can be provided to the users with some recommendation system utilizing the sentiment analysis technique.

FUTURE SCOPE

This model gives a clear understanding to public sentiments through high volume of complex live streaming data to create analytics which allow you to instantly explore the desired statistics in form of live visualization dashboards covering the detailed analysis of the entire data set. This model monitors the real-time Twitter feed for coronavirus-related tweets. The latest ML & AI heatmaps can be used to trace and filter out the misinformation floating and provide us with accurate data streaming on live dashboards. This information can help service providers to grasp the response of the users on their services, and manufacturers can use it for marketing research. Also, optimal service can be provided to the users with some recommendation system utilizing the sentiment analysis technique. It can be used in future by all the organisations as a feedback from there users to know about there requirements and their reactions of the various services so that oragnisation can work in fulfulling all the needs which have a beneficial impact on economic growth. This model can also be very beneficial for the governement in central, state and local sector to anticipate in welfare of public demands.

BIBILOGRAPHY

<https://www.ibm.com/support/knowledgecenter/SSJTVS/cxcloud/Using/Creatin gIBMID.html>.

<https://opensource.com/article/18/1/step-step-guide-git>

<https://developer.twitter.com/en>

<https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for -beginners-using-textblob/#>:

<https://textblob.readthedocs.io/en/dev/quickstart.html#create-a-textblob>

http://docs.tweepy.org/en/latest/getting_started.html

https://www.youtube.com/watch?v=Tv_5DHwIYdE&feature=youtu.be

<https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/>

<https://www.coursera.org/learn/python-plotting/home/welcome>

<https://www.ibm.com/account/reg/in-en/signup?formid=urx-34710>

https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.0.0/com.ibm.swg.ba.cognos.wig_cr.doc/c_gtstd_ica_overview.html

<https://www.youtube.com/watch?v=xBoem605XQ4>

<https://www.ibm.com/support/pages/how-upload-spreadsheet-files-cognos-analytics-11-and-access-data-dashboard>

https://www.ibm.com/support/knowledgecenter/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ug_cr_rptstd.doc/c_rs_cwa_mng_ext_dta.html

<https://newintelligence.ca/see-how-easy-it-is-to-build-a-dashboard-in-ibm-cognos-analytics/>

https://github.com/Atharva-Pandkar/sentiment_analysis

APPENDIX

Sentiment Analysis of Tweets using Tweepy and Textblob

```
1 import tweepy
2 from textblob import TextBlob
3 from csv import writer
4
5 consumer_key = 'skrrhvZmeKNuZvI0Z1sGgWQMJ'
```

```
6 consumer_key_secret =
  'RkFVEBn66lianCo1XgmXJohHQXbtkgrrPQ9KejqrU5MnVLs8b5'
7
8 access_token =
  '2261270106-vrPqTCE3iFNi6rGwUUQAhFYMLH1xXbxrgTWuumw'
9 access_token_secret =
  'd0fVkCHnChJFaYnt0eyuuCblpjACdBxemVsIi6vuwo7tf'
10
11 #positive = 1
12 #neutral = 0
13 #negative = -1
14
15 num_tweets = 0
16 num_positive_tweets = 0
17 num_negative_tweets = 0
18 num_neutral_tweets = 0
19
20 auth = tweepy.OAuthHandler(consumer_key,
  consumer_key_secret)
21
22 auth.set_access_token(access_token, access_token_secret)
23
24 api = tweepy.API(auth)
25
26 keywords = ["#coronavirus", "#lockdown",
  "#coronaupdatesindia", "#coronavirusindia"]
27
28 with open("tweet_data.csv", "w", encoding="utf-8") as
  file:
```

```

29         csv_writer = writer(file)
30         csv_writer.writerow(["Tweet Number",
    "Tweet", "Tweet Polarity", "Tweet Subjectivity",
    "Result", "Num Positive Tweets", "Num Negative Tweets",
    "Num Neutral Tweets", "Positive Percentage", "Negative
    Percentage", "Neutral Percentage"])
31
32 def get_tweets(keyword):
33     with open("tweet_data.csv", "a", encoding="utf-8")
    as file:
34         csv_writer = writer(file)
35         for tweet in
    tweepy.Cursor(api.search,q=keyword + "
    -filter:retweets", lang = "en",
    geocode='20.5936832,78.962883,10000km').items(200):
36             print(tweet.text)
37             global num_tweets
38             global num_positive_tweets
39             global num_negative_tweets
40             global num_neutral_tweets
41             num_tweets += 1
42             analysis = TextBlob(tweet.text)
43             print(analysis.noun_phrases)
44             if analysis.sentiment[0]>0:
45                 num_positive_tweets += 1
46                 csv_writer.writerow([num_tweets,
    tweet.text, analysis.sentiment[0],
    analysis.sentiment[1], "Positive", num_positive_tweets,
    num_negative_tweets, num_neutral_tweets])
47             elif analysis.sentiment[0]<0:

```

```

48             num_negative_tweets += 1
49             csv_writer.writerow([num_tweets,
tweet.text, analysis.sentiment[0],
analysis.sentiment[1], "Negative", num_positive_tweets,
num_negative_tweets, num_neutral_tweets])
50         else:
51             num_neutral_tweets += 1
52             csv_writer.writerow([num_tweets,
tweet.text, analysis.sentiment[0],
analysis.sentiment[1], "Neutral", num_positive_tweets,
num_negative_tweets, num_neutral_tweets])
53
54 for keyword in keywords:
55     get_tweets(keyword)
56
57 print(f"Total Tweets: {num_tweets}\nPositive Tweets:
{num_positive_tweets}\nNegative Tweets:
{num_negative_tweets}\nNeutral Tweets:
{num_neutral_tweets}")
58 positive_percentage = (num_positive_tweets/num_tweets) *
100
59 negative_percentage = (num_negative_tweets/num_tweets) *
100
60 neutral_percentage = (num_neutral_tweets/num_tweets) *
100
61 with open("tweet_data.csv", "a", encoding="utf-8") as
file:
62     csv_writer = writer(file)
63
64     csv_writer.writerow(["Summary", "", "", "", "", "", "", "", f"{p
ositive_percentage}", f"{negative_percentage}",
f"{neutral_percentage}"])

```

Analysis and Visualisation using IBM Watson Studio

```
1 import pandas as pd
2 import numpy as np
3 import types
4 import pandas as pd
5 from botocore.client import Config
6 import ibm_boto3
7
8 def __iter__(self): return 0
9
10 # @hidden_cell
11 # The following code accesses a file in your IBM Cloud
12 # Object Storage. It includes your credentials.
13 # You might want to remove those credentials before you
14 # share the notebook.
15 client_895e42ece5014c46aeca51ffdb5f1230 =
16     ibm_boto3.client(service_name='s3',
17
18         ibm_api_key_id='tx6AiW33Uba0xIrp5PFG--PejVbxJx1hlrFwxZRN
19         BCA6',
20
21         ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token"
22         ,
23         config=Config(signature_version='oauth'),
```

```

17     endpoint_url='https://s3.eu-geo.objectstorage.service.ne
    tworklayer.com')

18

19 body =
    client_895e42ece5014c46aeca51ffdb5f1230.get_object(Bucket
    ='try-donotdelete-pr-2klдоог21f8jew',Key='tweet_data
    (3).csv')['Body']

20 # add missing __iter__ method, so pandas accepts body as
    file-like object

21 if not hasattr(body, "__iter__"): body.__iter__ =
    types.MethodType( __iter__, body )

22

23 df = pd.read_csv(body)

24 df.head()

25 p=df.iloc[-1,8:9]['Positive Percentage']
26 float(p)

27 n=df.iloc[-1,9:10]['Negative Percentage']
28 float(n)

29 nu=df.iloc[-1,10:11]['Neutral Percentage']
30 float(nu)

31 index=df.index
32 a=len(index)-1
33 a

34 #Visualisation of Public Sentiments on Covid -19

35 import matplotlib.pyplot as plt

36 labels=['positive['+str(p)+'%]', 'Neutral
    ['+str(n)+'%]', 'Negative ['+str(nu)+'%]']

37

38 sizes=[p,n,nu]

```

```

39 colors=['#46DC78', '#50DFE9', '#F95B5B']
40 patches, texts=plt.pie(sizes, colors=colors, startangle=90)
41 plt.legend(patches, labels, loc="best")
42 plt.title('How people are reacting on ' + str(a) + '
    Tweets as Positive,Negative,Neutral')
43 plt.axis('equal')
44 plt.tight_layout()
45 plt.show()
46 %matplotlib notebook
47
48 import matplotlib.pyplot as plt
49 positive_tweets=df.iloc[:,5:6].values
50 negative_tweets=df.iloc[:,6:7].values
51 neutral_tweets=df.iloc[:,7:8].values
52 #Visualisation using Subplots
53 plt.figure()
54 ax1=plt.subplot(1,3,1)
55 plt.plot(positive_tweets, '-o')
56 plt.xlabel('No of Tweets')
57 plt.ylabel('Positive Tweets')
58 ax2=plt.subplot(1,3,2, sharey=ax1)
59 plt.plot(negative_tweets, '-x')
60 plt.xlabel('No of Tweets')
61 plt.ylabel('Negative Tweets')
62 ax3=plt.subplot(1,3,3, sharey=ax1)
63 plt.plot(neutral_tweets, '-')
64 plt.xlabel('No of Tweets')
65 plt.ylabel('Neutral Tweets')

```

```

66 plt.tight_layout()
67 mapping={'Positive':1, 'Negative':-1, 'Neutral':0}
68 df=df.replace({'Result':mapping})
69 df.head()
70 m=df.iloc[:,4:5].values
71 import matplotlib.animation as animation
72
73 n=2000
74 x=m
75 def update(curr):
76     if curr==n:
77         a.event_source.stop()
78     plt.cla()
79     bins=np.arange(-2,2,0.1)
80     plt.hist(x[:curr],bins=bins)
81     plt.axis([-2,2,0,2000])
82     plt.gca().set_title('Sampling the Result')
83     plt.gca().set_ylabel('No of tweets')
84     plt.gca().set_xlabel('Result')
85 #Visualisation using Animated Histograms
86 fig = plt.figure()
87 a = animation.FuncAnimation(fig, update, interval=100)
88 #taking 1 : positive ,
89 #         0: neutral,
90 #        -1 : negative
91 p=df.iloc[:,2:3].values
92 #Training and Testing Data
93 from sklearn.model_selection import train_test_split

```



```

94 p_train,p_test,m_train,m_test =
    train_test_split(p,m,random_state=0)\
95 df.dropna(subset=["Tweet
    Polarity","Result"],inplace=True)
96 from sklearn.linear_model import LinearRegression
97 model = LinearRegression()
98 m_train=np.nan_to_num(m_train)
99 p_train=np.nan_to_num(p_train)
100 linreg=model.fit(p_train,m_train)
101 #Calculating the score value of test and train values
102 print('R-squared score (training): {:.3f}'
103       .format(model.score(p_train,m_train)))
104 print('R-squared score (test): {:.3f}'
105       .format(model.score(p_test,m_test)))
106 import matplotlib.pyplot as plt
107 #Linear Regression Model
108 plt.figure(figsize=(7,8))
109 plt.scatter(p_train,m_train, marker= 'o', s=50,
    alpha=0.8)
110 plt.plot(p_train, linreg.coef_ * p_train +
    linreg.intercept_, 'r-')
111 plt.title('Linear Regression Model for Sentiment
    analysis of covid-19 tweets')
112 plt.xlabel('Polarity')
113 plt.ylabel('Result')
114 plt.legend(['Prediction','Result'])
115 plt.show()
116
117

```

Visualisation of model using Flask

```
1 from flask import
   Flask, render_template, url_for, request, redirect,
   make_response
2 import random
3 import json
4 from time import time
5 from random import random
6 from flask import Flask, render_template, make_response
7 app = Flask(__name__)
8 from textblob import TextBlob
9 from tweepy import OAuthHandler
10 from tweepy import API
11 from tweepy import Stream
12 from tweepy.streaming import StreamListener
13 import threading
14
15 ACCESS_TOKEN =
   '2261270106-vrPqTCE3iFNI6rGwUUQAHFYMLH1xXbxrgTWuumw'
16 ACCESS_TOKEN_SECRET =
   'd0fVkCHnChJFaYnt0eyuuCblpjACdBxemVsIi6vuwo7tf'
17 CONSUMER_KEY = 'skrrhvZmeKNuZvI0Z1sGgWQMJ'
18 CONSUMER_SECRET =
   'RkFVEBn661lianCo1XgmXJohHQXbtkgrrPQ9KejqrU5MnVLs8b5'
19
20 auth = OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
21 auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
```

```

22 api = API(auth, wait_on_rate_limit=True,
23           wait_on_rate_limit_notify=True)
24 lst=[]
25 total=[0,0,0,0]
26 text=[]
27 class Listener(StreamListener):
28     def __init__(self, output_file="ats.txt"):
29         super(Listener, self).__init__()
30         self.output_file = output_file
31     #@app.route('/data', methods=["GET", "POST"])
32     def on_status(self, status):
33         #print(status.text)
34         if not hasattr(status, "retweeted_status"):
35             analysis = TextBlob(status.text)
36             #print(analysis.sentiment[0])
37             if analysis.sentiment[0]>0:
38                 lst.append(analysis.sentiment[0])
39                 lst.append(0)
40                 lst.append(0)
41
42             elif analysis.sentiment[0]<0:
43                 lst.append(0)
44                 lst.append(analysis.sentiment[0])
45                 lst.append(0)
46             else:
47                 lst.append(0)
48                 lst.append(0)
49                 lst.append(0)

```

```
50         #return data()
51         text.append(status.text)
52     def on_error(self, status_code):
53         #print(status_code)
54         return False
55
56
57 @app.route('/', methods=["GET", "POST"])
58 def main():
59     return render_template('index.html')
60
61
62 @app.route('/data', methods=["GET", "POST"])
63 def data():
64     # Data Format
65     # [TIME, Temperature, Humidity]
66     cnt=0
67     t1 = threading.Thread(target=tr)
68     t1.start()
69     Temperature = random() * 1
70     Humidity = random() * 1
71     if len(lst)>3:
72         print(lst[cnt],lst[cnt+1],lst[cnt+2])
73         total[2]+=1
74         if lst[0]>0:
75             total[0]+=1
76         if lst[1]<0:
77             total[1]+=1
```

```

78         if lst[0] == 0 and lst[1] == 0 and lst[2] == 0:
79             total[3]+=1
80             data = [time() * 1000,
1st[cnt],lst[cnt+1],total[0],total[1],total[3],total[2],
text[0]]
81             lst.pop(0)
82             lst.pop(0)
83             lst.pop(0)
84         else:
85             data=[time()*1000,0,0,total[0],total[1],total[3],total[2]
], "NULL"]
86         if len(text)>0:
87             text.pop(0)
88             response = make_response(json.dumps(data))
89             response.content_type = 'application/json'
90             return response
91
92 def tr():
93     output = open('at.txt', 'w')
94     listener = Listener(output_file=output)
95     tra=["#Unlock1.0", "#Lockdown2.0", "#Lockdown3.0",
"#coronaupdatesindia", "#coronavirusindia", "#indiafightscorona"]
96     stream = Stream(auth=api.auth,
listener=listener,tweet_mode='extended')
97     stream.filter(languages=['en'],track=tra,locations =
[66.17,8.27,90.37,32.92],is_async=True)
98 if __name__ == "__main__":
99     app.run(debug=True)

```

100

101

IBM WATSON NOTEBOOK

https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/7f8e5778-14e7-44e3-8545-155d27144c8a/view?access_token=c4363a665658fa609ff9bc01100451c61d0f77de1856eb67ab6ab5e8419f2703

