**A PROJECT REPORT**

**ON**

**COVID SENTIMENTAL ANALYZER**

Submitted for IBM Hack Challenge 2020

**Submitted By:**

Voonna Sivani

V. Pawan Kumar Senapati

Project ID: SPS\_PRO\_807

Application ID:SPS\_CH\_APL\_20200002824

**INDEX**

**1.INTRODUCTION**

1.1 Overview

1.2 Purpose

**2.LITERATURE SURVEY**

2.1 Existing problem

2.2 Proposed solution

**3.THEORITICAL ANALYSIS**

3.1 Block Diagram

**4.EXPERIMENTAL INVESTIGATIONS**

**5.FLOW CHART**

**6.RESULT**

**7.ADVANTAGES & DISADVANTAGES**

**8.APPLICATIONS**

**9.CONCLUSION**

**10.FUTURE SCOPE**

**11.BIBILOGRAPHY**

**12.SOURCE CODE**

**1.INTRODUCTION**

**1.1 Overview:**

Social media websites have emerged as one of the platforms to raise users’ opinions. Opinion of people matters a lot to analyse how the propagation of information impacts the lives in a large-scale network like Twitter. Sentiment analysis of the tweets determine the polarity and inclination of vast population towards a specific topic, item or entity.

So, we have developed a web application that extracts  the most recent data available on different platforms like twitter, news and Wikipedia through web scraping and perform sentimental analysis on that data and return an interactive visualization dashboard.

**1.2 Purpose:**

Growth of social media has resulted in an explosion of publicly available, user generated text on the World Wide Web. These data and information can potentially be utilized to provide real-time insights into the sentiments of people.

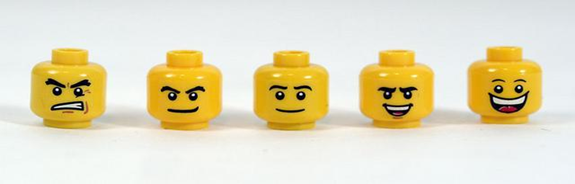
Communication and the availability of these real time opinions from people around the world make a revolution in computational linguistics and social network analysis. So ,by making an analysis of their sentiments through social media like twitter may capture millions of people's views about the pandemic and it will help the government to take decisions .

**2.LITERATURE SURVEY**

**2.1 Existing problem:**

Along with the Coronavirus pandemic, another crisis has manifested itself in the form of mass  fear and panic phenomena, filled by incomplete and often inaccurate information. There is therefore a tremendous need to address and better understand COVID-19’s informational crisis and gauge public sentiment, so that appropriate messaging and policy decisions can be implemented.

The rapid spread of Coronavirus and COVID-19 infections have created a strong need for discovering rapid analytics methods for understanding the flow of information and the development of mass sentiment in pandemic scenarios. While there are numerous initiatives analysing healthcare, preventative, care and recovery, economic and network data, there has been relatively little emphasis on the analysis of aggregate personal level and social media communications.



**2.2 Proposed solution:**

In this project, we present analysis of Twitter and News data to identify public sentiment, specifically, tracking the progress of fear, which has been associated with the rapid spread of Coronavirus and COVID-19 infections. This outlines a methodological approach to analysing Twitter data specifically for identification of sentiment, key words associations and trends in this current COVID-19 phenomena.

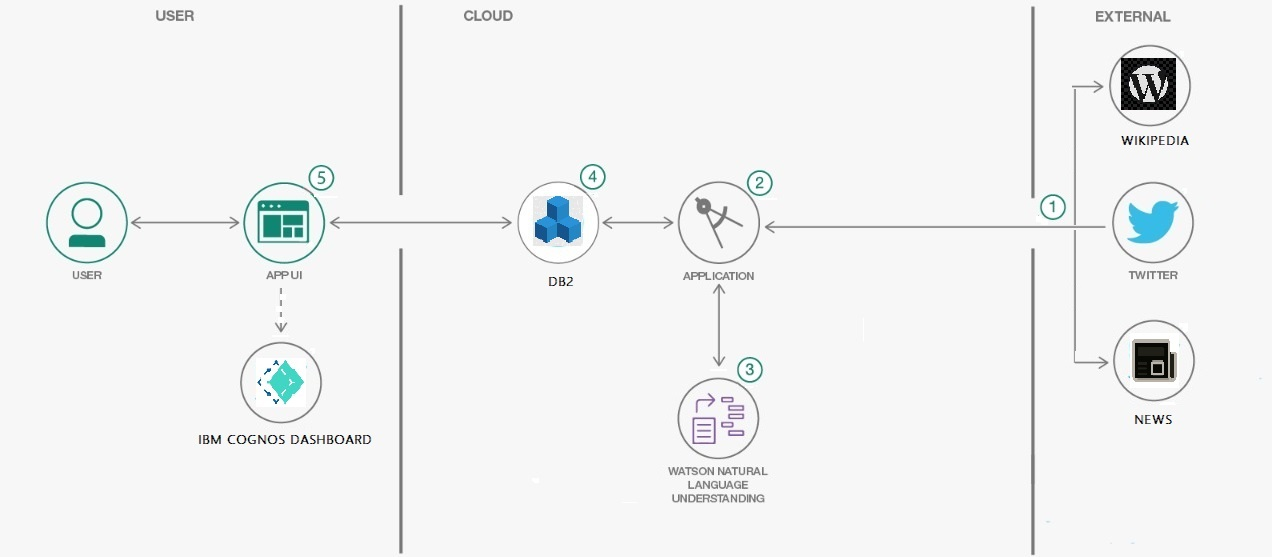
We performed Sentiment Analysis on live Twitter data and news. As a first step ,we performed web scraping from Twitter and google news to extract the most recent data on the topic of coronavirus and lockdown extension.

As a second step, this unstructured data once scraped (extracted information from web) is processed through **Watson NLU** and converted to structured data. This Structured data is then stored in **db2** database of ibm Watson.

The data stored in DB2 is used to generate the dynamic visualization dashboards.

**3.THEORITICAL ANALYSIS**

**3.1 Block Diagram:**



Steps:

1. Scraping data from Google search ,Twitter tweets and Wikipedia.

2.Feeding the data to IBM Watson Natural Language Understanding(NLU).

3.Storing the analysed data to DB2 database.

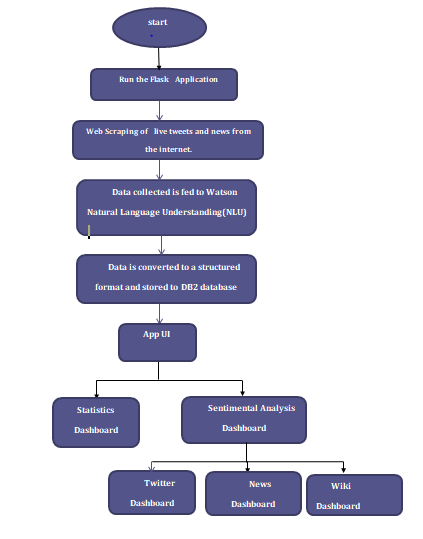
4.Visualizing the analysed results in Visualization Dashboard through App UI.

**4.EXPERIMENTAL INVESTIGATIONS**

In this process of developing the project we have undergone many investigation processes to learn and understand new concepts so that we can build this web application successfully for which we had to learn and investigate following:

* IBM Cloud.
* Flask
* Watson Natural Language Understanding service
* Db2 on IBM cloud.
* IBM Cognos Visualization Dashboard
* Using the Tweepy Package to extract tweets from twitter based on relevant hashtags(# tags).

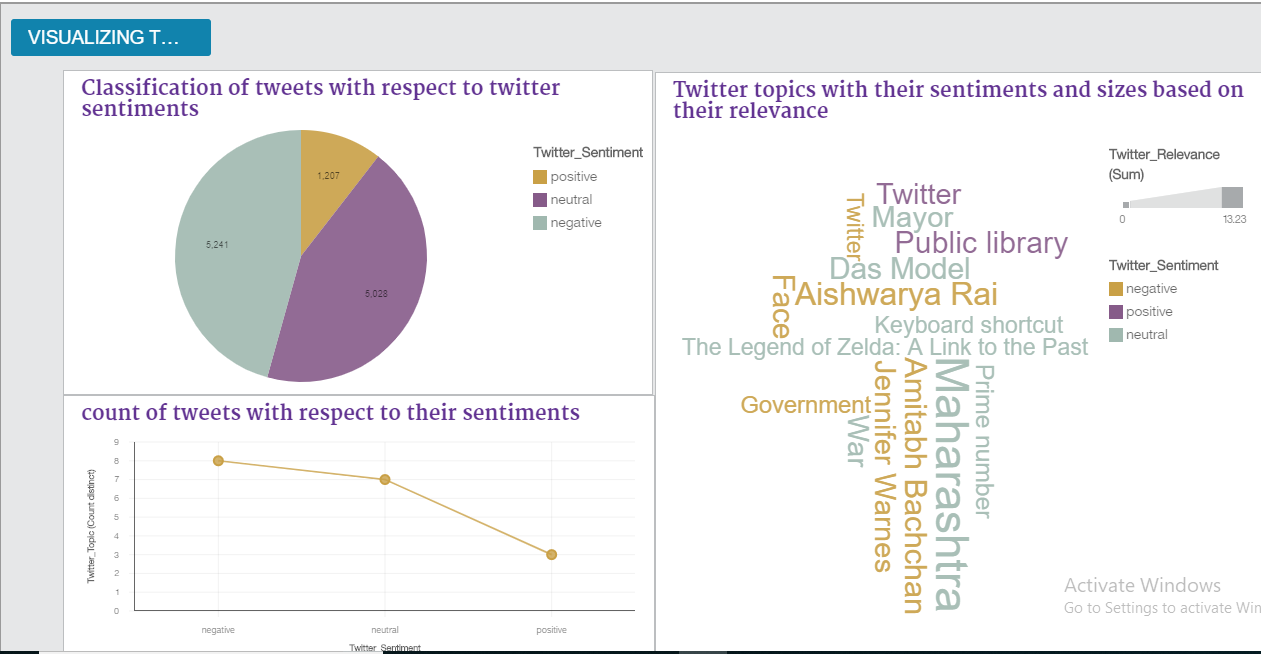
**5.FLOW CHART**



**6.RESULT**

The result of the project  is the interactive visualization dashboard from IBM Cognos Analytics Dashboard Embedded.

**Results for Twitter Analysis:**



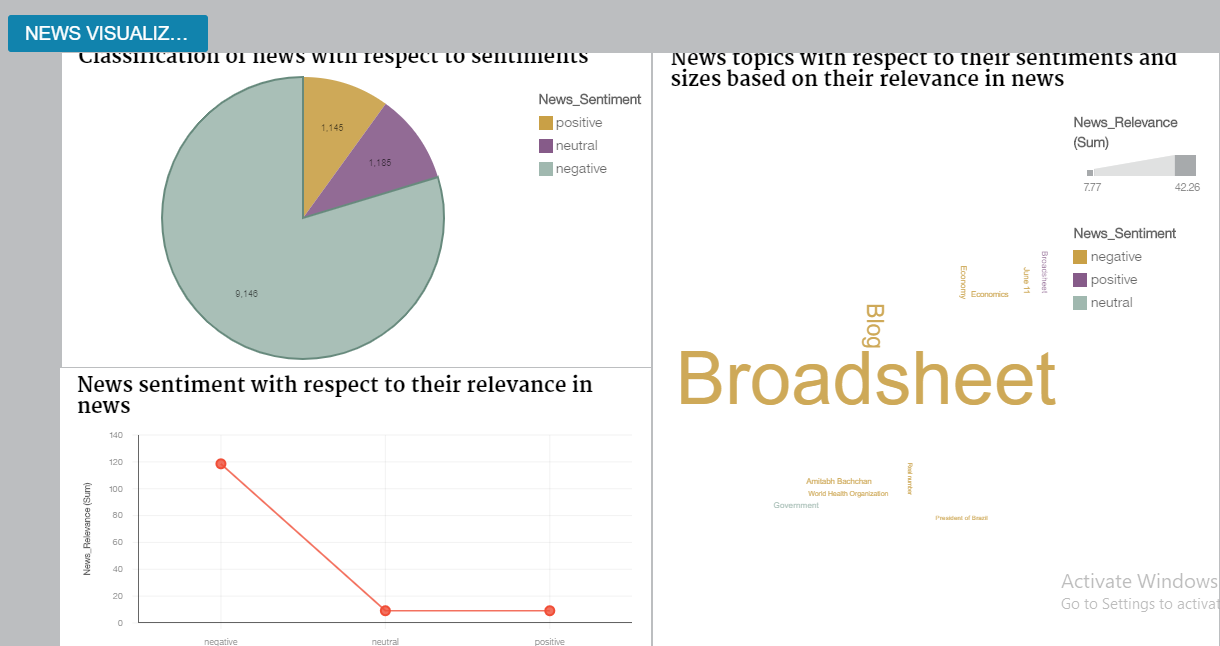
It consists of:

1.Classification of tweets according to their polarity.( positive ,negative or Neutral)

2.Most Trending Twitter topics.

3.Relevance Vs Sentiments curve.

**Results for News Analysis:**



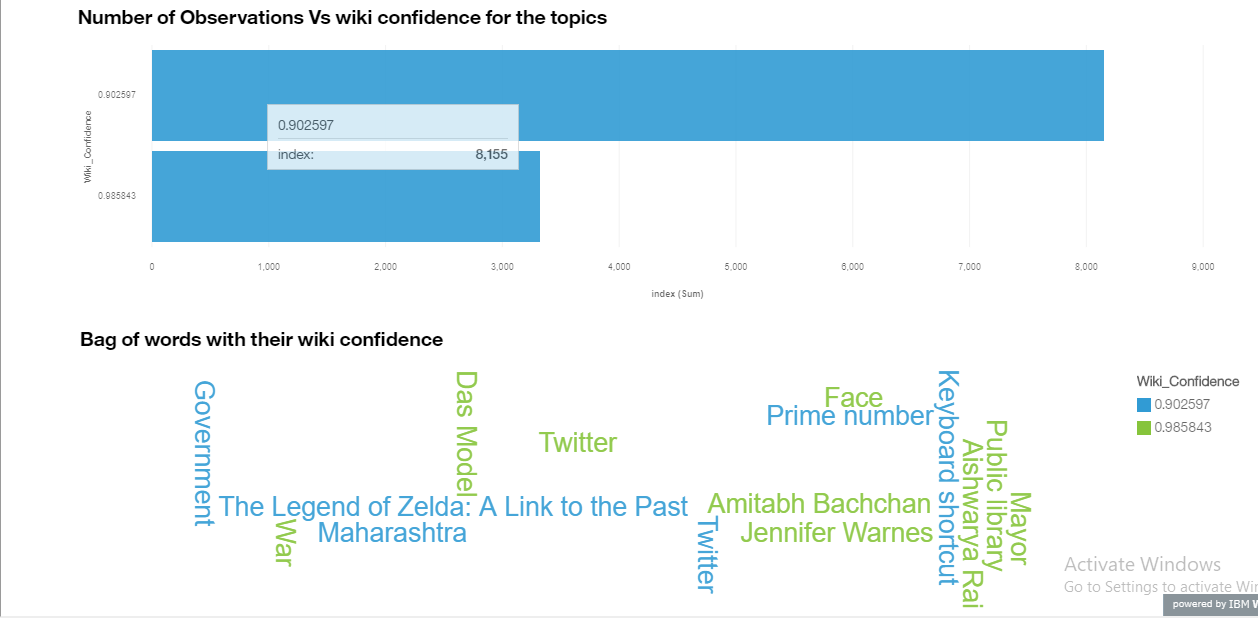
It consists of:

1.Classification of News according to their polarity.(positive ,negative or Neutral)

2.Most Trending News topics.

3.Relevance in News  Vs Sentiments curve.

**Results for wiki analysis:**



It consists of:

1.Number of Observations Vs. Wiki confidence for the topics.

2.Bag of words with their wiki confidence.

**7.ADVANTAGES & DISADVANTAGES**

**Advantages:**

1.The Dashboard gets updated each time we run the flask application thus giving live analysis.

2.The User Interface is simple, consistent and easy to use .

3.As Watson NLU is used , sentiment Analysis  accuracy is high.

4. As DB2 database is used to store data ,data is very well structured and easy to understand.

**Disadvantages:**

1.As the problem statement was about corona virus and lockdown extension ,we predefined the required hashtags (#tags) in the code itself.

**8.APPLICATIONS:**

The news media always has the loudest voices to inform the public. But we know that most major players hold political perspectives that have profound impacts on our decision making.

Apart from defeating the disease, we all should do our own parts in defeating the spread of fears and loathe.

How to stay clear-headed?

Don’t just read one side of the story, listen to more voices. Thus ,these analysis can help us to reach to a more clear conclusion by learning from more number of people.

So ,by making an analysis of their sentiments through social media like twitter may capture millions of people's views about the pandemic and it will help the government to take decisions  as per analysis made.

**9.CONCLUSION**

We have addressed issues surrounding public sentiment reflecting deep concerns about Coronavirus and COVID-19, leading to the identification of growth in fear sentiment and negative sentiment. We also demonstrated the use of exploratory and descriptive textual analytics using Watson NLU and data visualization using IBM Cognos Analytics dashboard to discover early stage insights, such as by grouping of words by levels of a specific non-text.

The Conclusion is that we identified the growth in  negative sentiment is gradual and a word cloud that  consists of all the trending topics on twitter and news.

**10.FUTURE SCOPE**

1.This project can be implemented to visualize the sentiments of people on any sort of topic by making minor changes in the code.

2.It will help to know people’s opinion on kind of decision taken by the Government.

3.Adding any kind of additional features to the dashboard is possible. Hence it is very dynamic.

**11.BIBILIOGRAPHY**

1. <https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>

2. Watson Natural Language Understanding

<https://www.youtube.com/watch?v=WDItryHZ3JI>

<https://cloud.ibm.com/apidocs/natural-language-understanding>

3. DB2 database

<https://www.youtube.com/watch?v=J9xtw0zjDLY>

<https://www.ibm.com/in-en/products/db2-database>

4. Flask using python

<https://www.youtube.com/watch?v=Z1RJmh_OqeA>

5. IBM Cognos Dashboard Embedded

<https://www.youtube.com/watch?v=FCQ6GAcyQ7E>

**12.SOURCE CODE:**

**Defining the hashtags:**

data={

    "topics": [

        {

        "coronavirus": [

            "A deadly virus that is suffering the whole world ",

            "www.who.int"

        ]

        },

        {

        "Lockdown-extension": [

            "India has declared the lockdown throughout the country for almost 3 months",

            "mohfw.gov.in"

        ]

        }

    ]

    }

**Web Scraping using Google search:**

    def scrape\_news\_summaries\_google(s):

        ua = UserAgent()

        number\_result=10

        google\_url = "https://www.google.com/search?q=" + s + "&num=" + str(number\_result)

        response = requests.get(google\_url, {"User-Agent": ua.random})

        soup = BeautifulSoup(response.text, "html.parser")

        result\_div = soup.find\_all('div', attrs = {'class': 'ZINbbc'})

        news\_items=[]

        for r in result\_div:

            # Checks if each element is present, else, raise exception

            try:

                news\_dict=dict()

                link = r.find('a', href = True)

                title = r.find('div', attrs={'class':'vvjwJb'}).get\_text()

                description = r.find('div', attrs={'class':'s3v9rd'}).get\_text()

                # Check to make sure everything is present before appending

                if link != '' and title != '' and description != '':

                    news\_dict['news\_link']=link['href']

                    news\_dict['summary']=description

                    news\_items.append(news\_dict)

            # Next loop if one element is not present

            except:

                continue

        return news\_items

**Analyzing using NLU**

def analyze\_using\_NLU(analysistext):

        """ Extract results from Watson Natural Language Understanding for each news item

        """

        res=dict()

        response = natural\_language\_understanding.analyze(

            text=analysistext,

            features=Features(

                            sentiment=SentimentOptions(),

                            entities=EntitiesOptions(),

                            keywords=KeywordsOptions(),

                            emotion=EmotionOptions(),

                            concepts=ConceptsOptions(),

                            categories=CategoriesOptions(),

                            ))

        res['results']=response

        return res

**Twitter Analysis:**

def hasTwitter(topic\_name):

        cnbcVal=0

        cnbcLinks=[]

        ET\_link=[]

        s='"'+topic\_name+'"'+' twitter'

        res= scrape\_news\_summaries\_google(s)

        return res

Twitter=[]

    for f in final\_rows:

        for name, info in f.items():

            temp=dict()

            news=hasTwitter(name)

            for n in news:

                flag=0

                if 'summary' in n:

                    summary=n['summary']

                    flag=1

                link=n['news\_link']

                temp=dict()

                if flag:

                        temp['topic\_Name']=name

                        temp['Twitter\_news\_link']=link

                        temp['Twitter\_Topic'],temp['Twitter\_Relevance'],temp['Twitter\_Sentiment']=getTechAreaNews(summary)

                        Twitter.append(temp)

**Wiki Analysis:**

def hasWiki(s):

        wikiVal=0

        wikiLinks=[]

        s=s.replace(' ','+')

        link='https://en.wikipedia.org/w/index.php?search='+s+'&title=Special%3ASearch&go=Go'

        r = requests.get(link)

        #print(r.status\_code)

        content = r.text

        return content

**Storing all the extracted data to .csv files:**

    file = open("ET\_final.csv", "r")

    ET = pd.read\_csv(file, delimiter=',')

    file = open("Wiki.csv", "r")

    Wiki = pd.read\_csv(file, delimiter=',')

    file = open("Twitter.csv", "r")

    Twitter = pd.read\_csv(file, delimiter=',')

**Defining the db2 database:**

dsn\_database = credentials\_1['db']

    dsn\_hostname = credentials\_1['host']

    dsn\_port = 50000

    dsn\_uid = credentials\_1['username']

    dsn\_pwd = credentials\_1['password']

    dsn = (

        "DRIVER={{IBM DB2 ODBC DRIVER}};"

        "DATABASE="+str(dsn\_database)+";"

        "HOSTNAME="+str(dsn\_hostname)+";"

        "PORT="+str(dsn\_port)+";"

        "PROTOCOL=TCPIP;"

        "UID="+str(dsn\_uid)+";"

        "PWD="+str(dsn\_pwd)+";").format(dsn\_database, dsn\_hostname, dsn\_port, dsn\_uid, dsn\_pwd)

    conn = ibm\_db.connect(dsn, "", "")

**Delete the OLD data from the DB2 Table:**

sql = "DELETE FROM "+dsn\_uid+".DATA\_FOR\_DASHBOARD"

    ins\_sql=ibm\_db.prepare(conn, sql)

    ibm\_db.execute(ins\_sql)

    tuple\_of\_tuples = tuple([tuple(x) for x in compiled\_rows.values])

**Inserting newly extracted data to DB2:**

i=1

    for x in compiled\_rows.values:

        vals= (i,) + tuple(x)

       # print(vals)

        sql = "INSERT INTO "+dsn\_uid+".DATA\_FOR\_DASHBOARD VALUES"+ str(vals)

        i=i+1

        ins\_sql=ibm\_db.prepare(conn, sql)

        ibm\_db.execute(ins\_sql)