Assignment 3

PSOSM

Gagandeep Singh: 2016037
Suraj Prathik Kumar: 2016101
Vyshakh: 2016120

We used python to complete all the tasks and did it in Jupyter Notebook. We used **Tweepy api** to extract the information from Twitter.

Police Twitter Handle : @**DelhiPolice**

Question 1

Part 1

We collected tweets in which @DelhiPolice was mentioned for the past few days. We were able to collect over 5000 tweets in which @DelhiPolice was mentioned using Twitter's search api.

We use the Regular expression library in python to find the **mobile numbers** that appeared in the tweets.

Finding mobile numbers

In [206]:
```
mobile_numbers={}
for tweet in tweets:
    number=re.findall('(?:\s+|)((0|(?:(\+|)91))(?:\s|-)*(?:(?:\d(?:\s|-)*\d{9})|(?:\d{2}(?:\s|-)*\d{8})|(?:\d{3}(?:\
    real_num=[]
    for num in number:
        for val in num:
            if len(val)>=8:
                real_num.append(val)
    if len(number)>0:
        mobile_numbers[tweet.id]=real_num
```

In [207]:
```
mobile_numbers
```

Out[207]:
```
{1241300650060689408: ['8595383769'],
 1241296901103710208: ['+918129307424'],
 1241281965799526400: ['+919013151515'],
 1241263877632823296: ['8168050200'],
 1241259458010869761: ['9106508873'],
 1241258824796794881: ['9990116220', '7303277701', '7065857983'],
 1241256740114632705: ['+911833502576'],
 1241256357090791426: ['+917636817859'],
 1241234198700023808: ['8750871437'],
 1241230679754465282: ['9560340354'],
 1241222732819259393: ['9971494771'],
 1241215478745296896: ['+918745815059'],
 1241214852565061632: ['+919990890928'],
 1241061653602373632: ['9173000023'],
 1241022703676657665: ['+919836515628'],
 1240993396094668802: ['7534061767'],
 1240973911837245443: ['9871222844'],
 1240971895731707904: ['9873293701'],
 1240968341633679360: ['7596056681']}
```

We use Regular expression and NLTK library in python to find the **names** that appeared in the tweets.

In [240]:
```
def get_human_names(text):
    tokens = nltk.tokenize.word_tokenize(text)
    pos = nltk.pos_tag(tokens)
    sentt = nltk.ne_chunk(pos, binary = False)

    person = []
    name = ""
    for subtree in sentt.subtrees(filter=lambda t: t.label() == 'PERSON'):
        for leaf in subtree.leaves():
            person.append(leaf[0])
        if len(person) > 1: #avoid grabbing lone surnames
            for part in person:
                name += part + ' '
            if name[:-1] not in person_list:
                person_list.append(name[:-1])
            name = ''
        person = []
```

In [242]:
```
names={}
for tweet in tweets:
    person_list = []
    person_names=person_list
    text=""
    try:
        text=tweet.retweeted_status.full_text
    except:
        text=tweet.full_text
    get_human_names(tweet.full_text)
    for person in person_list:
        person_split = person.split(" ")
        for name in person_split:
            if wordnet.synsets(name):
                if(name in person):
                    person_names.remove(person)
                    break
    names[tweet.id]=person_names
```

In [243]:
```
names
```

```
1241293549066260480: [],
1241293518259154945: [],
1241293518103969792: [],
1241293517818748928: [],
1241293482037112833: [],
1241293266709917697: [],
1241293235823116289: [],
1241293217154252800: [],
1241293201303949313: ['Jyoti Singh Pandey'],
1241293133729501186: [],
1241292973314203653: [],
1241292943845142529: ['Kapil Mishra', 'Anurag Thakur', 'Pravesh Verma'],
1241292931366916096: ['Shaheen Bagh'],
1241292905580343296: [],
1241292893039529985: [],
1241292879705673728: [],
1241292850274471936: [],
1241292751573938176: [],
```

```
In [209]: tw=api.get_status(1241258824796794881,tweet_mode="extended")

In [210]: tw.full_text

Out[210]: 'Dear @DelhiPolice \nThe name of the person \nNeha Sharma 9990116220\n7303277701 Agent \n7065857983 Rajeev Malhotra
          \nThese are the person trying to do some financial scam in the name of bima lokpal earlier in 2015 also same scam b
          usted by You, 1 of my culigue also got cheated'
```

```
In [245]: names[1241258824796794881]

Out[245]: ['Neha Sharma', 'Rajeev Malhotra']
```

We use Regular expression and NLTK library in python to find the **ZIP Codes** that appeared in the tweets.

```
In [344]: pincodes={}
          for tweet in tweets:
              text=""
              try:
                  text=tweet.retweeted_status.full_text
              except:
                  text=tweet.full_text
              text=re.sub('(?:\s+|)((0|(?:(\+|)91))(?:\s|-)*(?:(?:\d(?:\s|-)*\d{9})|(?:\d{2}(?:\s|-)*\d{8})|(?:\d{3}(?:\s|-)*\
              final_text=[]
              for i in text.split():
                  if i[0]=='@':
                      continue
                  elif len(i)>6:
                      continue
                  else:
                      final_text.append(i)
              text=' '.join(final_text)

              code=re.findall(r'^.*(?P<zipcode>\d{6}).*$', text)
              if len(code)>0:
                  pincodes[tweet.id]=code
```

```
In [345]: pincodes

Out[345]: {1241257090733105152: ['122002'],
           1241198190734393344: ['110018'],
           1241037070153334784: ['500000'],
           1240990774902194176: ['110021'],
           1240990110608281600: ['110021'],
           1240972569202667520: ['110094']}
```

```
In [356]: tw2=api.get_status(1240972569202667520,tweet_mode="extended")

In [357]: tw2.full_text
Out[357]: '@DelhiPolice \nFriday weekly market setup in ankur enclave karawal nagar Delhi 110094'
```

Finding all the people tagged in the tweet

```
In [358]: mentions={}

for tweet in tweets:
    text=""
    try:
        text=tweet.retweeted_status.full_text
    except:
        text=tweet.full_text
    tagged=[]
    for i in text.split():
        if i[0]=='@':
            tagged.append(i)
    if len(tagged)>0:
        mentions[tweet.id]=tagged
```

```
In [359]: mentions
Out[359]: {1241307073821827072: ['@sarthakraizada,',
            '@solicitorNikunj,',
            '@pradosh_shetty',
            '@DelhiPolice'],
           1241307063583748098: ['@ani_jharia07',
            '@HMOIndia',
            '@DelhiPolice',
            '@ArvindKejriwal'],
           1241307046571433985: ['@DelhiPolice', '@DcpNorthDelhi'],
           1241307011037442048: ['@HMOIndia', '@DelhiPolice', '@ArvindKejriwal'],
           1241307005639254017: ['@sarthakraizada,',
            '@solicitorNikunj,',
            '@pradosh_shetty',
            '@DelhiPolice'],
           1241306998232076289: ['@ManishJhaTweets',
            '@pokershash',
            '@DelhiPolice',
            '@DcpNorthDelhi',
            '@AamAadmiParty'],
```

Hashtags in each tweet

In [211]:
```
hashtagsDict={}#stores list of hashtags in each tweet as a dictionary (tweet_id: [hastags]
for tweet in tweets:
    hashtagsDict[tweet.id]=[]
    for tag in tweet.entities['hashtags']:
        hashtagsDict[tweet.id].append(tag['text'])
```

In [212]:
```
hashtagsDict
```

```
 1241307063583748098: [],
 1241307046571433985: [],
 1241307011037442048: [],
 1241307005639254017: ['ShaheenBagh'],
 1241306998232076289: [],
 1241306949846630403: [],
 1241306940078272514: [],
 1241306933056794624: [],
 1241306919278727170: [],
 1241306890514182145: [],
 1241306880686690304: ['ShaheenBagh', 'Coronavirus'],
 1241306878723809280: [],
 1241306818548133888: ['दिल्ली',
  'घायल',
  'कोरोना',
  'बीमार',
  'खतरनाक',
  'Shame',
  'DelhiRiots2020',
```

Part 2

We collected tweets by @DelhiPolice using Tweety api's Cursor from February 1, 2020 to March 6, 2020.

```
In [104]: tweets=[]
          count=0
          for tweet in tweepy.Cursor(api.user_timeline,id = "@DelhiPolice",tweet_mode='extended').items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweets.append(tweet)
              if tweet.created_at<startDate:
                  print("yoyo")
                  break
              count+=1
              print(count," -> ",len(tweets)," -> ",tweet.created_at)
```

230 tweets were collected within the given time period.

```
In [110]: tweets_text=[]
          for tweet in tweets:
              try:
                  tweets_text.append(tweet.retweeted_status.full_text)
              except:
                  tweets_text.append(tweet.full_text)

          tweets_text

Out[110]: ['In the run up to Holi: Huge haul of illicit liquor seized, 3 arrested in 3 different cases with total 140 cartons (6972 quarters) an
          by three PS Govindpuri, PS Okhla and PS Badarpur of SE Dist.@DelhiPolice https://t.co/a2ZhXjyFdm',
           'Conspiracy, Robbery and Murder:  Case solved by the Team of PS Amar Colony with the arrest of three accused persons. @DelhiPolice ht
           'Public Meeting: with Maulanas and respectables of Shaheen bagh to have their cooperation in maintaining peace in the area. @DelhiPol
          jy5d',
           'https://t.co/6drSLFMFLP',
           'https://t.co/ND8VAFUVj0',
           'Notorious Thak Thak Gang members arrested within hours of committing an incident with the recovery of 11 pairs of earrings,necklace,
          ,8 mobile phones and motorcycle\n@CPDelhi @LtGovDelhi @DelhiPolice \n#KeepingDelhiSafe https://t.co/zyS6omNLRz',
           '@CPDelhi visited residence of Sh. Anuj Sharma, ACP Gokul Puri, the officer who was injured facing rioters &amp;amp; yet,heroically rescu
          Sharma, DCP Shahdara,who was grievously injured trying to control the senseless violence. Delhi Police is proud of such officers https
           'Public meeting was held at Jaitpur to get their cooperation and confidence in maintaining law and order and to improve crime situati
          olice https://t.co/KqjRBqgpYg',
           "@SubtleTraveller @AtishiAAP @DelhiPolice Some unsubstantiated reports of tense situation in SouthEast &amp;amp; West District are being
          dia. It is to reiterate that these are all rumours. Don't pay attention to such rumours. Delhi Police is closely monitoring accounts s
          king action.",
           'Hey guys!\n\nNot done. You have taken the job of spreading नफ़रत so brazenly.\n\nRest assured, we are watching you all, and mighty wel
          of these are fake IDs, be sure of our capabilities to hunt you down 😎.\n\nTake this as a sweet warning!\n\n#DelhiPoliceNailsFake',
           'Hey guys!\n\nNot done. You have taken the job of spreading नफ़रत so brazenly.\n\nRest assured, we are watching you all, and mighty wel
```

A dataset was created to store information about these tweets. The dataset had multiple features describing the information about each tweet. We used **Pandas** library to create a dataframe.

Feature of the dataset:

1. **Text** : contains the text of the tweet
2. **Number of Hashtags**
3. **Hashtags** : List of all the hashtags used in that tweet
4. **Length of Tweet** : length of the text of the tweet
5. **IsRetweet**: A boolean value to depict if a tweet is a retweet or not

The dataset was then stored in a CSV for further use.

	text	Number of Hashtags	Hashtags	Length of tweet	isRetweet
1	In the run up to Holi: Huge haul of illicit li...	0	[]	252	True
2	Conspiracy, Robbery and Murder: Case solved b...	0	[]	153	True
3	Public Meeting: with Maulanas and respectables...	0	[]	159	True
4	https://t.co/6drSLFMFLP	0	[]	23	False
5	https://t.co/ND8VAFUVj0	0	[]	23	False
6	Notorious Thak Thak Gang members arrested with...	0	[]	263	True
7	@CPDelhi visited residence of Sh. Anuj Sharma,...	0	[]	308	False
8	Public meeting was held at Jaitpur to get thei...	0	[]	184	True
9	@SubtleTraveller @AtishiAAP @DelhiPolice Some ...	0	[]	324	True
10	Hey guys!\n\nNot done. You have taken the job ...	0	[]	276	True
11	Hey guys!\n\nNot done. You have taken the job ...	0	[]	300	True
12	Hey guys!\n\nNot done. You have taken the job ...	0	[]	276	True
13	Efforts are being made to arrest him .	0	[]	38	True
14	The police verified the same to be incorrect a...	0	[]	260	True
15	A section of Media has reported that Tahir Hus...	0	[]	273	True
16	Minister of State for Consumer Affairs appreci...	0	[]	271	True
17	It's the BAAGHI season.\n\nWe are #Baaghi agai...	4	[Baaghi, false, fake, FakeNews]	306	True
18	Senior Officers from Dwarka District interacte...	0	[]	221	True
19	Addl.DCP/Dwarka Dist. and Akshat Kaushal I.P.S...	0	[]	277	True
20	दिल्ली में कहीं भी हिंसा की कोई खबर नहीं है।	0	[]	130	True

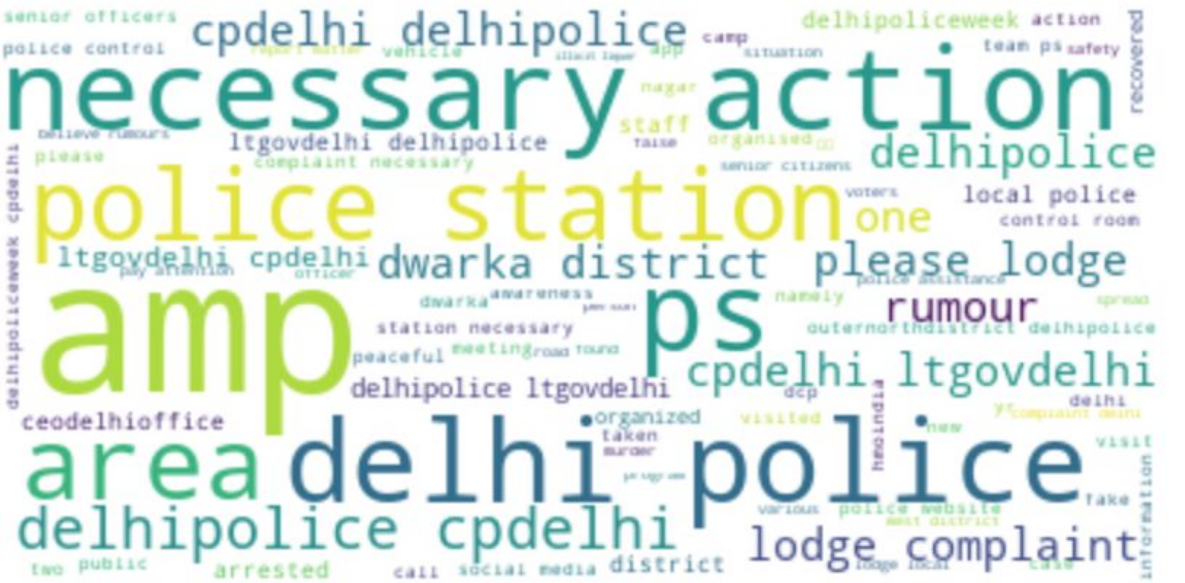
Preprocessing the Text

We used multiple libraries to preprocess our text to get the most information out of it as possible. Initially we removed all the punctuation marks and stopwords from the text using regular expression library and NLTK library.
We then used the NLTK library to stem and lemmatize the words in each tweet the

```
In [481]: lowerTextTweets=[]
          for tweet in tweets_text:
              tweet=re.sub('[,\.!?]', '', tweet)
              tweet=re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+','',tweet)
              words=tweet.split()
              withoutStop=[i for i in words if i.lower() not in stopwords.words('english')]
              result=[]
              stemmer= PorterStemmer()
              for word in withoutStop:
                  result.append(stemmer.stem(WordNetLemmatizer().lemmatize(word, pos='v')))
              tweet=' '.join(result)
              tweet=' '.join(withoutStop)
              lowerTextTweets.append(tweet.lower())
```

We then created a word cloud of the most used words to better understand the topics being addressed in the tweets.

```
wordcloud.generate(allTweets)
wordcloud.to_image()
```



Using the Gensim library we converted the text of the tweets to a dictionary and then into a bag of words, i.e., converting each tweet and the words contained in it into a vector form so as to feed it to a LDA model for topic modelling.

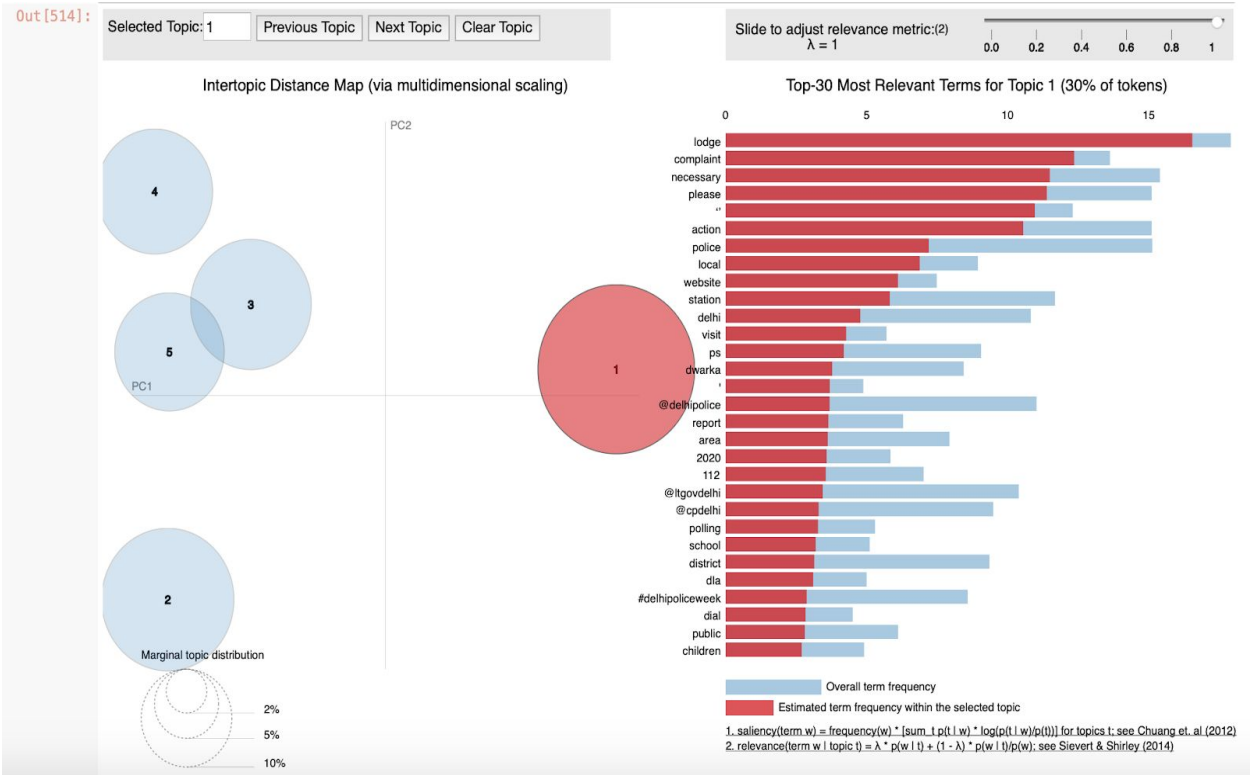
```
In [484]: dictionary = gensim.corpora.Dictionary(preProcessedTweets_list)

In [485]: bagOfWords=[dictionary.doc2bow(l) for l in preProcessedTweets_list]

In [547]: bagOfWords

Out[547]: [[(0, 1),
            (1, 1),
            (2, 2),
            (3, 1),
            (4, 1),
            (5, 1),
            (6, 1),
            (7, 1),
            (8, 1),
            (9, 1),
```

The bag of words was then converted into term frequency-inverse document frequency to train out topic modelling model.

Using the gensim library, we then created an LDA model for it to segregate each tweet into 5 topics and calculate a score for each tweet for each of the 5 different topics.

```
In [491]: lda_tfidf=gensim.models.LdaMulticore(bow_tfidf, num_topics=5, id2word=dictionary, passes=10, workers=2)

In [492]: for idx, topic in lda_tfidf.print_topics(-1):
              print('Topic: {} \nWords: {}'.format(idx, topic))

Topic: 0
Words: 0.004*"regard" + 0.004*"contact" + 0.004*"police" + 0.003*"requested" + 0.003*"officer" + 0.003*"investigating" + 0.003*"designa
+ 0.003*"delhi" + 0.003*"district"
Topic: 1
Words: 0.004*"@delhipolice" + 0.004*"area" + 0.003*"station" + 0.003*"officer" + 0.003*"house" + 0.003*"grievance" + 0.003*"&amp;" + 0.
oncerned" + 0.002*"domination"
Topic: 2
Words: 0.003*"#delhipoliceweek" + 0.003*"done" + 0.003*"organized" + 0.003*"swaroop" + 0.003*"senior" + 0.003*"citizens" + 0.002*"@ltgo
" + 0.002*"@cpdelhi" + 0.002*"dwarka"
Topic: 3
Words: 0.013*"lodge" + 0.010*"complaint" + 0.009*"necessary" + 0.009*"please" + 0.009*"'' + 0.009*"action" + 0.006*"police" + 0.006*"l
" + 0.005*"station"
Topic: 4
Words: 0.006*"@dtptraffic" + 0.005*"matter" + 0.005*"forwarded" + 0.004*"rumours" + 0.004*"action" + 0.003*"necessary" + 0.003*"police"
03*"social" + 0.003*"appeal"
```

Below is a visual representation of our LDA model which segregates tweets into 5 topics.



We then tested our model for a few unseen tweets which were tweeted by @DelhiPolice in the past few days. Below are the results provided by our model for a tweet.

```
In [534]: tweetForTest.lower()

Out[534]: "🔴 warning\n\nwe have spotted this fake notice being circulated purportedly issued by delhi police\n\nwe have not issued any such advisory
a fine on march 22 please tell your family and friends that this is false &amp;amp; fake\n\nlet's make #janatacurfewmarch22 a success "
```

```
In [548]: topicIndex=-1
topicScore=0
for index, score in sorted(lda_tfidf[vector]):
    if score>topicScore:
        topicScore=score
        topicIndex=index
    print("Score: {}\t Topic: {}".format(score, lda_bagOfWords.print_topic(index)))
print("\nMost relevant topic")
print("Score: {}\nTopic: {}".format(topicScore, lda_tfidf.print_topic(topicIndex)))

Score: 0.040340349078178406      Topic: 0.021*"@delhipolice" + 0.012*"@cpdelhi" + 0.010*"ps" + 0.007*"&amp;amp;" + 0.006*"police" + 0.006*"@ltgo
व" + 0.006*"one" + 0.006*"staff" + 0.005*"recovered"
Score: 0.040876101702451706      Topic: 0.041*"police" + 0.024*"@delhipolice" + 0.020*"@cpdelhi" + 0.019*"delhi" + 0.017*"@ltgovdelhi" + 0.0
014*"district" + 0.009*"dwarka" + 0.008*"please" + 0.008*"local"
Score: 0.04045272246003151       Topic: 0.026*"@delhipolice" + 0.022*"@cpdelhi" + 0.017*"@ltgovdelhi" + 0.015*"ps" + 0.011*"&amp;amp;" + 0.010*"
#delhipoliceweek" + 0.009*"delhi" + 0.007*"rumours" + 0.007*"dwarka"
Score: 0.040316905826330185      Topic: 0.014*"@ltgovdelhi" + 0.014*"district" + 0.013*"delhi" + 0.012*"@cpdelhi" + 0.011*"@delhipolice" + 0
08*"fake" + 0.008*"police" + 0.008*"north" + 0.007*"&amp;amp;"
Score: 0.8380139470100403        Topic: 0.048*"necessary" + 0.047*"lodge" + 0.046*"action" + 0.035*"complaint" + 0.032*"please" + 0.029*"‘’"
+ 0.012*"@dtptraffic" + 0.010*"forwarded" + 0.007*"@delhipolice"

 Most relevant topic
Score: 0.8380139470100403
Topic: 0.006*"@dtptraffic" + 0.005*"matter" + 0.005*"forwarded" + 0.004*"rumours" + 0.004*"action" + 0.003*"necessary" + 0.003*"police" + 0.
03*"social" + 0.003*"appeal"
```

The tweet in our test example is a tweet which appeals to the society to stay away from the fake news/rumour which is spreading telling people that whoever gets out of his/her home on March 22, 2020 , the days of #JantaCurfew will be charged a fine. This was a rumour spreading across social media.

Our model quite correctly segregated it in the topic of rumour and fake news with an 83% score.

Question 2

We collected tweets in which @DelhiPolice was mentioned for the past few days. We were able to collect over 5000 tweets in which @DelhiPolice was mentioned using the Twitter's search api.

```
In [4]: len(tweets)
Out[4]: 5121

In [6]: tweets
```


```
tatus_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id': None, 'in_reply_to_user_id_str': None, '
in_reply_to_screen_name': None, 'user': {'id': 384337715, 'id_str': '384337715', 'name': 'Shubhendu', 'screen_name'
: 'BBTheorist', 'location': 'New Delhi', 'description': 'Old School Guy • India & UK trained Lawyer • Counsel @ Sup
reme Court • Tea, Travel, Paintings, History • Introvert • Nerd • Conservative • TISS survivor', 'url': 'https://t.
co/v4L7YtF6Rg', 'entities': {'url': {'urls': [{'url': 'https://t.co/v4L7YtF6Rg', 'expanded_url': 'https://www.opind
ia.com/author/shubhenduanand/', 'display_url': 'opindia.com/author/shubhen…', 'indices': [0, 23]}]}, 'description':
{'urls': []}}, 'protected': False, 'followers_count': 14449, 'friends_count': 987, 'listed_count': 72, 'created_at'
: 'Mon Oct 03 14:17:16 +0000 2011', 'favourites_count': 26781, 'utc_offset': None, 'time_zone': None, 'geo_enabled'
: True, 'verified': False, 'statuses_count': 28203, 'lang': None, 'contributors_enabled': False, 'is_translator': F
alse, 'is_translation_enabled': False, 'profile_background_color': '1A1B1F', 'profile_background_image_url': 'http:
//abs.twimg.com/images/themes/theme9/bg.gif', 'profile_background_image_url_https': 'https://abs.twimg.com/images/t
hemes/theme9/bg.gif', 'profile_background_tile': False, 'profile_image_url': 'http://pbs.twimg.com/profile_images/1
239258059764981761/9uRGIu3r_normal.jpg', 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/123925805
9764981761/9uRGIu3r_normal.jpg', 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/384337715/1584297292'
, 'profile_link_color': 'D2691E', 'profile_sidebar_border_color': '000000', 'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000', 'profile_use_background_image': True, 'has_extended_profile': False, 'default_profi
le': False, 'default_profile_image': False, 'following': False, 'follow_request_sent': False, 'notifications': Fals
e, 'translator_type': 'none'}, 'geo': None, 'coordinates': None, 'place': None, 'contributors': None, 'is_quote_sta
tus': False, 'retweet_count': 215, 'favorite_count': 529, 'favorited': False, 'retweeted': False, 'possibly_sensiti
```


We then preprocessed text of the tweets for exploratory data analysis and to better understand the topics that were being discussed in these tweets.

```
In [22]: def preProcess(tweet):
             tweet=re.sub('[,\.!?]', '', tweet)
             tweet=re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+','',tweet)
             words=tweet.split()
             withoutStop=[]
             for i in words:
                 if i.lower() in stopwords.words('english') or i=='&amp;':
                     continue
                 else:
                     withoutStop.append(i)
             tweet=' '.join(withoutStop)
             return tweet.lower()
```

After the exploratory data analysis, we created a score to classify each tweet into rumour or not a rumour. The score was based on is the user who tweeted is verified or not, number of retweets of the tweet, number of followers of the user, number of likes on the tweet and the length of the tweet.

In [97]:
```
score=[]
for index, row in dataset.iterrows():
    if row['Is Verified']:
        score.append(1)
        continue
    if row['No. of Retweets']==0 and row['No. of Likes']==0:
        l=len(row["text"])
        s=((0.01*l+0.7*row["Follower Count"])/(l+row["Follower Count"]))
        score.append(s)
    else:
#         if row["Follower Count"]==0:
#             s=((0.35*row['No. of Retweets']+0.15* row['No. of Likes'])/(row['No. of Likes']+row['No. of Retweets'
#             score.append(s)
#         else:
        s=((1.5 * row['No. of Retweets'] + 0.5 * row['No. of Likes']) / (row["Follower Count"]+row['No. of Likes']+
        score.append(s)
```

Using the score for classification of tweets among rumour and not a rumour, we were able to get 973 tweets which were classified as a rumour out of 5121 tweets.

In [105]:
```
co=0
for sc in score:
    if sc<0.02:
        co+=1
co
```

Out[105]: 973

The result were stored in a dataset and saved as a CSV file named “data_Q2.csv”.

```
In [107]: dataset["Is Rumour"]=isRumour
          dataset["Score"]=score
          dataset.head(30)
```

Out[107]:

	text	Tweet_Id	User_Id	No. of Retweets	No. of Likes	Is Verified	Follower Count	Is Rumour	Score
1	My friends @sarthakraizada, @solicitorNikunj, ...	1241307073821827072	724492284092383232	215	0	False	183	0	0.810302
2	@ani_jharia07 @HMOIndia @DelhiPolice @ArvindKe...	1241307063583748098	1105633420397277184	0	0	False	896	0	0.656695
3	@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट मे...	1241307046571433985	1088048352	88	0	False	72	0	0.825000
4	This protest is Now illegal. If you can't foll...	1241307011037442048	1105633420397277184	1	0	False	896	1	0.001672
5	My friends @sarthakraizada, @solicitorNikunj, ...	1241307005639254017	525133945	215	0	False	46	0	1.235632
6	@ManishJhaTweets @pokershash @DelhiPolice @Dcp...	1241306998232076289	1218935059819851776	0	0	False	65	0	0.339779
7	शाहीनबाग की महिलाएं जनता कर्फ्यू में शामिल नही...	1241306949846630403	101990582	228	0	False	772	0	0.342000
8	यदि मेरी चिंता अनुचित है तो मुझे निश्चित रूप स...	1241306940078272514	1212347139344322561	1	0	False	9610	1	0.000156
9	@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट मे...	1241306933056794624	417763872	88	0	False	875	0	0.137072
10	यदि मेरी चिंता अनुचित है तो मुझे निश्चित रूप स...	1241306919278727170	1212347139344322561	1	2	False	9610	1	0.000260
11	@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट मे...	1241306890514182145	330778397	88	0	False	555	0	0.205288
12	So 2 ppl at #ShaheenBagh have tested +ive for ...	1241306880686690304	1024260706424635393	17	0	False	47	0	0.398438
13	@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट मे...	1241306878723809280	43912419	88	0	False	813	0	0.146504
14	#दिल्ली - 36 घंटे में 55 मरे,800 #घायल\n \n...	1241306818548133888	1140297456053211136	0	0	False	1364	0	0.581786
15	@MumbaiPolice @DelhiPolice @rsprasad please se...	1241306808616194049	52382339	0	0	False	82	0	0.271944
16	Why step outside when you know the risk? \n\nघर...	1241306800990945282	109774946	94	0	False	1810	0	0.074055

```
In [364]: dataset.to_csv("data_Q2.csv",index=False)
```

We the collected a tweet rumour dataset from online resources to train our model and check the credibility of our score.

```
In [108]: rumourDf=pd.read_csv("aug_complete.csv")
```

```
In [110]: rumourDf.head(30)
```

Out[110]:

	id	text	created_at	label
0	3.240000e+17	RIP to the 8 year old girl who died in the Bos...	15/04/2013 19:51	1
1	3.240000e+17	Danny Amendola is going to donate $100 dollars...	16/04/2013 07:58	1
2	3.240000e+17	an 8 year old girl died, running for her class...	15/04/2013 19:14	1
3	3.240000e+17	An 8 year old child was one of the deaths in b...	16/04/2013 01:06	1
4	3.240000e+17	8 years old waiting for his father to finish t...	16/04/2013 08:32	1
5	3.240000e+17	An baby angel now, this was the little 8 year ...	15/04/2013 18:15	1
6	3.240000e+17	8 year old child died today. #prayforboston	15/04/2013 15:33	1
7	3.240000e+17	Police Commissioner says 3rd explosion happene...	15/04/2013 13:55	1
8	3.240000e+17	More info on the 8-year-old that died. His mot...	16/04/2013 07:44	1
9	3.240000e+17	8 yr old boy dead in #boston terror attack.. h...	16/04/2013 07:29	1
10	3.240000e+17	Boston police chief says explosion took place ...	15/04/2013 13:52	1
11	3.240000e+17	Counterterrorism officials found what they bel...	15/04/2013 17:43	1
12	3.240000e+17	proud-atheist: Westboro Baptist Church to pick...	15/04/2013 20:48	1
13	3.240000e+17	Wow. One of the confirmed dead, is an 8 year o...	15/04/2013 15:45	1
14	3.240000e+17	Google has created a people finder for the #Bo...	15/04/2013 14:25	1

We then vectorized each tweet in the dataset to feed it to a machine learning model.

```
In [145]: vectorMatrix=CountVectorizer()
          messageVector = vectorMatrix.fit_transform(rumourDf['text'])
```

```
In [146]: tfidf = TfidfTransformer()
          message_tfidf = tfidf.fit_transform(messageVector)
```

The data was then split into training and testing data with a random split of 67% for training data and 33% for testing data.

```
In [147]: msg_train, msg_test, label_train, label_test = train_test_split(message_tfidf, rumourDf['label'], test_size=0.33)
```

We then trained a Naive Bayes model with the training data and then tested the model on the test data. The model predicted whether a tweet is rumour or not with an accuracy of 97% on the test data.

```
In [148]: rumour_detect_model = MultinomialNB().fit(msg_train, label_train)

In [149]: predictions = rumour_detect_model.predict(msg_test)
```

```
In [151]: print(classification_report(label_test,predictions))
```

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1238
1	0.96	0.95	0.95	397
accuracy			0.98	1635
macro avg	0.97	0.97	0.97	1635
weighted avg	0.98	0.98	0.98	1635

We then used this model to test the predictions made by the score we created.
We again vectorized all the tweets we collected and transformed them using the machine learning model and predicted if the tweet is a rumour or not and compared the results with the result given by our score. It predicted **tweet not being a rumour** correctly with an accuracy of **81%** and **tweet being a rumour** with an accuracy of **33%**, thus giving an average of precision as **57%** and a weighted average of **72%.**

Comparing the score classification and NB classification

```
In [152]: tweet_text_vector=vectorMatrix.transform(dataset["text"])

In [153]: tweet_text_tfidf=tfidf.transform(tweet_text_vector)

In [154]: predictNB=rumour_detect_model.predict(tweet_text_tfidf)

In [155]: print(classification_report(dataset["Is Rumour"],predictNB))
```

	precision	recall	f1-score	support
0	0.81	1.00	0.89	4148
1	0.33	0.01	0.01	973
accuracy			0.81	5121
macro avg	0.57	0.50	0.45	5121
weighted avg	0.72	0.81	0.73	5121

1. **Tweets where model and prediction score did well**
   a. Tweet id: 1241300953183084545
   The tweet is very likely to be a rumour as its quite impossible that the police would practice some kind of sexual assault on the victim.

```
In [380]: twtw=api.get_status(1241300953183084545,tweet_mode="extended")
          twtw.full_text

Out[380]: 'Sexual assault by delhi police. Police refusing to file an FIR calling the victim a prostitute. Batton marks repo
          rtedly found on her body. Call the numbers below and demand urgent action. @DCPSouthDelhi @DelhiPolice @HuffPostWo
          men @JagoriSafeDelhi https://t.co/WsN0Gv9Qmj'
```

```
text                Sexual assault by delhi police. Police refusin...
Tweet_Id                                          1241300953183084545
User_Id                                                    3782810959
No. of Retweets                                                     6
No. of Likes                                                        2
Is Verified                                                     False
Follower Count                                                   8530
Is Rumour                                                           1
Score                                                      0.00117123
Prediction                                                      True
```

b. Tweet id: 12410110601752944 65
The tweet talks about delhi police paying the damage for violence in jamia.

text	Will Delhi Police pay for damage to Jamia Libr...
Tweet_Id	1241011060175294465
User_Id	1591906044
No. of Retweets	22
No. of Likes	0
Is Verified	False
Follower Count	1793
Is Rumour	1
Score	0.0181818
Prediction	True

c. Tweet id: 1241049960721920000
The tweet talks about closing all religious places immediately and tries to divide people in the name of religion. Hence, it is predicted as a rumour.

```
In [385]: twtw=api.get_status(1241049960721920000,tweet_mode="extended")
          twtw.full_text

Out[385]: '@BesuraTaansane @Anubhav95020225 @MumbaiPolice @DelhiPolice @myogiadityanath सबसे ज्यादा ईरान तुर्की में इस्लाम ही चपेट में आया और देश लौटने को बेहाल है, काबा कब का सुनसान हो गया मक्का में कोई माई का लाल नहीं दिख रहा,पाकिस्तान में तम्बू में दुबक के बैठे हैं और वायरस से फटती न हीं!!मंदिर,church,gurudwara shut down immediately!!'
```

text	@BesuraTaansane @Anubhav95020225 @MumbaiPolice...
Tweet_Id	1241049960721920000
User_Id	788997521649479681
No. of Retweets	0
No. of Likes	1
Is Verified	False
Follower Count	297
Is Rumour	1
Score	0.00167785
Prediction	True

d. Tweet id: 1241197471205748736
The tweet talks about punishment to the lawyers who did some meaningful talk on some topic. This might be a rumour as the tweet does not specify why the lawyers were punished.

```
In [387]: twtw=api.get_status(1241197471205748736,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[387]: '@VirenderBaisoy1 @shivana32207580 @rsprasad @Mynation_MP @vaastavngo @DelhiPolice Police वकीलों की #DhankiBaat पर भारी चोट ।'
```

text	@VirenderBaisoy1 @shivana32207580 @rsprasad @M...
Tweet_Id	1241197471205748736
User_Id	1061589188477210625
No. of Retweets	2
No. of Likes	0
Is Verified	False
Follower Count	302
Is Rumour	1
Score	0.00986842
Prediction	True

e. Tweet id: 1241304662063509504
The tweet only talks about the importance of staying home in the time of pandemic. Hence, it is not classified as a rumour.

```
In [390]: twtw=api.get_status(1241304662063509504,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[390]: 'Why step outside when you know the risk?\n\nघर से बाहर मत निकलो न \n#CoronaStopKaroNa \n\nStay indoors, while we are at work for your safety! \n\n#HelpUsToHelpYou #JanataCurfewMarch22\n#Covid_19 \n\n@CPDelhi @LtGovDelhi @DelhiPolice https://t.co/oJMExKiWmx'
```

text	Why step outside when you know the risk?\n\nघर...
Tweet_Id	1241304662063509504
User_Id	1025417394850816000
No. of Retweets	94
No. of Likes	0
Is Verified	False
Follower Count	452
Is Rumour	0
Score	0.258242
Prediction	False

2. Tweets where model did not do well

a. Tweet id: 1241304642094379008

The tweet most likely seems a political stunt in the name of COVID-19 to empty the protest at Shaheen Bagh. It is not a responsibility of the general public to order Delhi Police to arrest these people. Moreover, there is no confirmation that 2 people at Shaheen Bagh are suffering from coronavirus. Hence, it should be classified as a rumour but it is not.

```
In [392]: twtw=api.get_status(1241304642094379008,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[392]: 'So 2 ppl at #ShaheenBagh have tested +ive for #Coronavirus , Yet\n1) they are defiant\n2) they continue their pub
          lic gathering\nNow these protestors are officially criminals\nArrest them ASAP ! @DelhiPolice'
```

text	So 2 ppl at #ShaheenBagh have tested +ive for ...
Tweet_Id	1241304642094379008
User_Id	148687950
No. of Retweets	17
No. of Likes	0
Is Verified	False
Follower Count	81
Is Rumour	0
Score	0.260204
Prediction	False

b. Tweet id: 1241304662541623297

The tweet is a false information about two girls who came from Dubai but are not cooperating in the tense situation because of coronavirus. This again is a rumour but is not classified as one.

```
In [393]: twtw=api.get_status(1241304662541623297,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[393]: '@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट में दो लड़कियां दुबई से आई है। RWA के कहने पर भी ना मेडिकल टेस्ट करवा रही है ना ही घर में रह रही हैं। 10
          0 पर शिकायत करवाई तो PCR वाले का कॉल आया कि इसमें हम क्या करेंगे !\nPCR Official No - +918129307424\n#Covid_19 @DcpNorthDelhi
          '
```

text	@DelhiPolice शालीमार बाग /कनिष्क अपार्टमेंट मे...
Tweet_Id	1241304662541623297
User_Id	3401368630
No. of Retweets	88
No. of Likes	0
Is Verified	False
Follower Count	19
Is Rumour	0
Score	1.23364
Prediction	False

c. Tweet id: 1241304735233085440

The tweet is ordering/ informing the police about a 6th guy who was involved in the Nirbhaya case, which was a rumour.

```
In [396]: twtw=api.get_status(1241304735233085440,tweet_mode="extended")
          twtw.full_text

Out[396]: 'Hello @DelhiPolice , It looks like this person has more information than you guys on the Nirbhaya case. Maybe you
          guys want to question him and find out about the 6th guy. Then we would know if he is lying or if there is any tru
          th behind it. https://t.co/CdrzfyDsKh'
```

text	Hello @DelhiPolice , It looks like this person...
Tweet_Id	1241304735233085440
User_Id	1444205084
No. of Retweets	0
No. of Likes	0
Is Verified	False
Follower Count	156
Is Rumour	0
Score	0.265677
Prediction	False

d. Tweet id: 1241304812559294465

```
In [414]: twtw=api.get_status(1241304812559294465,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[414]: 'So 2 ppl at #ShaheenBagh have tested +ive for #Coronavirus , Yet\n1) they are defiant\n2) they continue their pub
          lic gathering\nNow these protestors are officially criminals\nArrest them ASAP ! @DelhiPolice'
```

text	So 2 ppl at #ShaheenBagh have tested +ive for ...
Tweet_Id	1241304812559294465
User_Id	147229311
No. of Retweets	17
No. of Likes	0
Is Verified	False
Follower Count	17398
Is Rumour	1
Score	0.00146425
Prediction	False

e. Tweet id: 1241300750753345538
The tweet says that people will be fined if they come out of their homes on janta curfew day which was a rumour as stated by delhi police on their twitter handle. But the model did not classify it as a rumour.

```
In [415]: twtw=api.get_status(1241300750753345538,tweet_mode="extended")
          twtw.retweeted_status.full_text

Out[415]: 'सोशल मीडिया पर खबर वायरल हुई कि 22 मार्च को दिल्ली में बिना किसी बड़ी वजह से अगर कोई घूमता दिखा तो उसपर दिल्ली पुलिस 11000Rs. का जुर्माना लगा
```

text	सोशल मीडिया पर खबर वायरल हुई कि 22 मार्च को दि...
Tweet_Id	1241300750753345538
User_Id	1062665231858573312
No. of Retweets	1
No. of Likes	0
Is Verified	False
Follower Count	267
Is Rumour	1
Score	0.00559701
Prediction	False

3. **Ways to improve our data and model**
    a. As many tweets were written in hindi and the model was only trained in english language, the model behaves unpredictably when it faces a foriegn language. We could train the model using multiple languages to make it predict better.

    b. The score we calculated was a basic score based on the user who tweeted is verified or not, number of retweets of the tweet, number of followers of the user, number of likes on the tweet and the length of the tweet. We could use better scoring techniques like doing textual and image analysis on tweets and use these features to score a tweet.

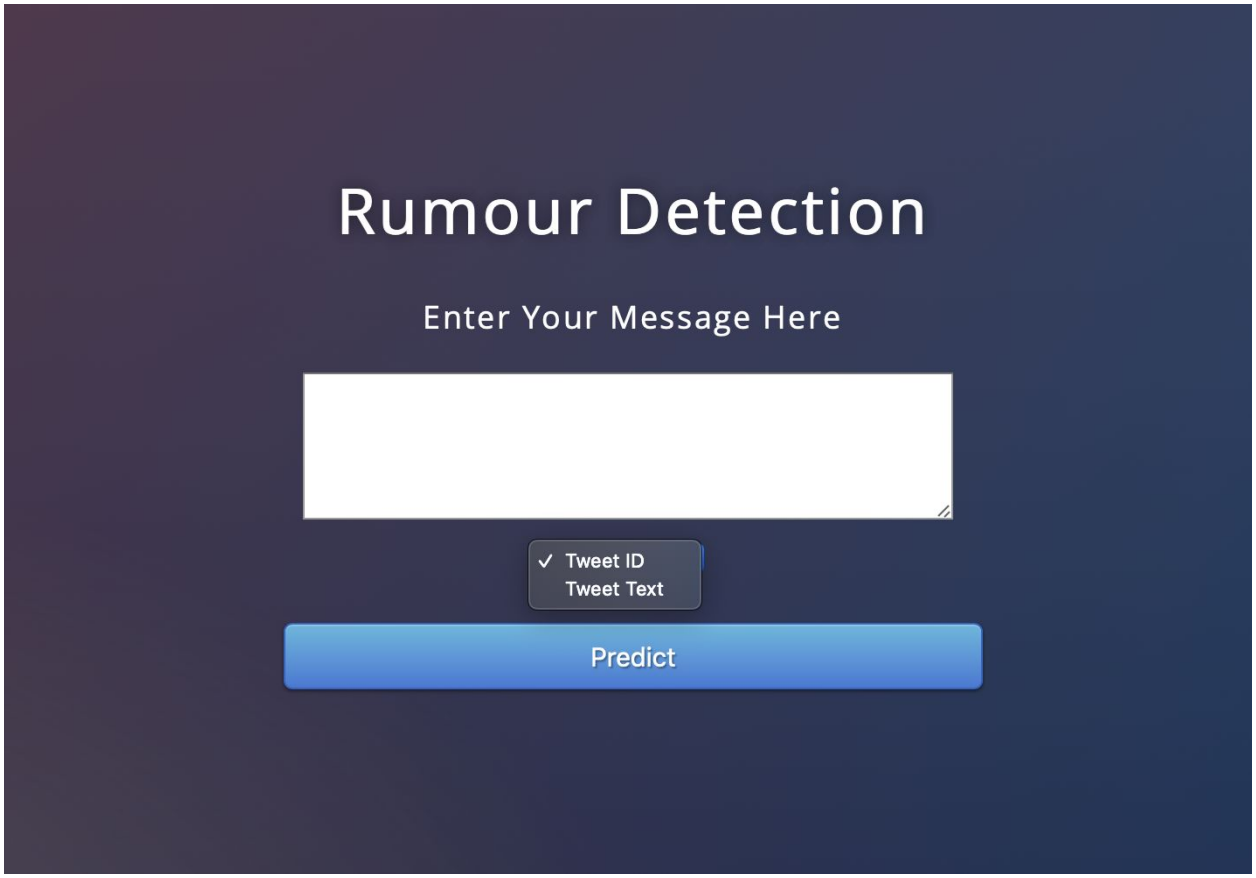
    c. The data we used to train the model might not have been enough to train the model. Moreover it did not have many features to train the model. Choosing or creating data with multiple features would help our model predict better.

Model Deployment

We created a webapp using **Flask** framework which helps the user predict whether the tweet is a rumour or not.
Using **Pickle** library, we dumped our model into the memory for future use on our webapp.

```
In [361]: import pickle

In [362]: pickle.dump(rumour_detect_model, open('model.pkl', 'wb'))
          pickle.dump(vectorMatrix, open('vectorMatrix.pkl', 'wb'))
          pickle.dump(tfidf, open('tfidf.pkl', 'wb'))
```

On the webapp, the user has an ability to provide the webapp a tweet id or simply the whole text of the tweet for it to predict whether it is a rumour or not.



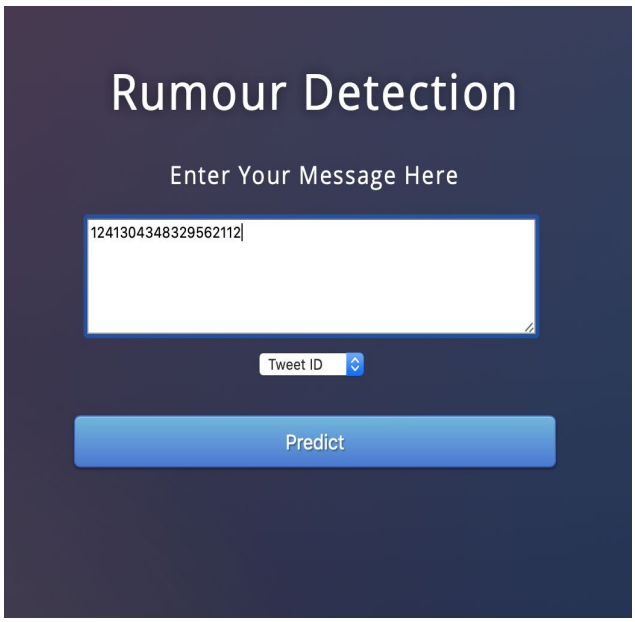
The app loads the model we dumped into the memory. The app runs the text of the tweet through the models to classify the tweet as a rumour or not.

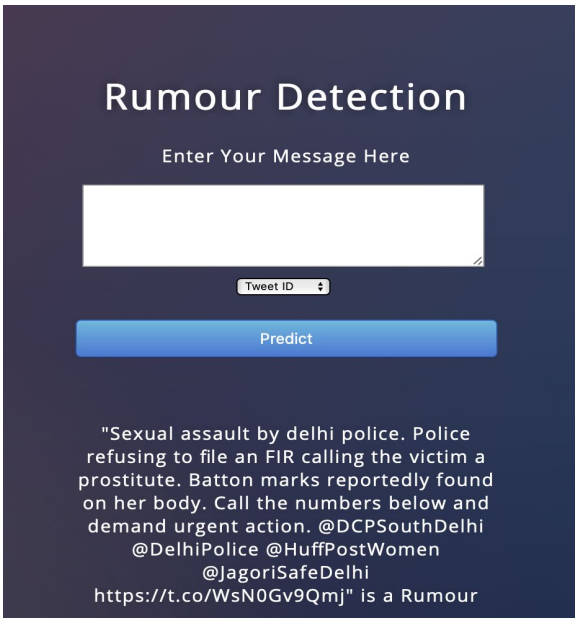
```
model = pickle.load(open('../model.pkl', 'rb'))
vectorMatrix = pickle.load(open('../vectorMatrix.pkl', 'rb'))
tfidf = pickle.load(open('../tfidf.pkl', 'rb'))
```

```
selection = [request.form['select']]

message = [request.form['message']]
text=[]

val=0
if selection[0] == 'tweetId':
    try:
        val=int(message[0])
    except:
        return render_template('index.html', prediction_text='Invalid Tweet ID!'
    tweet=""
    try:
        tweet= api.get_status(int(message[0]), tweet_mode="extended")
    except:
        return render_template('index.html', prediction_text='Tweet does not exi
    try:
        text.append(tweet.retweeted_status.full_text)
    except:
        text.append(tweet.full_text)
else:
    text=message
```

```
#nsforming data
tweet_text=text[0]
vector = vectorMatrix.transform(text)
data = tfidf.transform(vector).toarray()

#predictiong through the model
predictions = model.predict(data)

if predictions[0]==0:
    return render_template('index.html', prediction_text='\"{}\"{} is {}'.format(tw
else:
    return render_template('index.html', prediction_text='\"{}\"{} is {}'.format(tw
```

The app shows the output on the frontend.





References:

1. https://stackoverflow.com/questions/20290870/improving-the-extraction-of-human-names-with-nltk/24119115
2. https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0
3. https://towardsdatascience.com › topic-modeling-and-latent-dirichlet-alloca...
4. http://regexlib.com