TEXT ANALYTICS

INDIAN RAILWAY PRIVATISATION

SENTIMENT ANALYSIS USING TWITTER API

SUBMITTED BY:
RASHMI NAIR
PGDM RESEARCH & BUSINESS ANALYTICS
ROLL NO. 40

A) FINDING THE LOCATION AND NAME OF THE USER

1) Importing the required Libraries

```
In [98]: #Getting Twitter Data
         #Access twitter API in python
         !pip install tweepy
         import os
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         import itertools
         import collections

         import tweepy as tw
         import nltk
         from nltk.corpus import stopwords
         import re
         import networkx

         import warnings
         warnings.filterwarnings("ignore")

         sns.set(font_scale=1.5)
         sns.set_style("whitegrid")
```

```
Requirement already satisfied: tweepy in c:\users\administrator\anaconda3\lib\site-packages (3.9.0)
Requirement already satisfied: requests[socks]>=2.11.1 in c:\users\administrator\anaconda3\lib\site-packages (from tw
eepy) (2.22.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\administrator\anaconda3\lib\site-packages (from t
weepy) (1.3.0)
Requirement already satisfied: six>=1.10.0 in c:\users\administrator\anaconda3\lib\site-packages (from tweepy) (1.14.
0)
```

2) Reading the Keys

```
In [99]: #Reading Keys
         consumer_key= 'slwO72gLyBj5lgwKwhUwsS0eO'
         consumer_secret= 'AA1rFwgi0Ay3i3fj5kIqht5ZQhFMzmcoOm3tfNv9A3g5SFsOKy'
         access_token= '1305890390961250305-Mr3jQ5E2YyX21KzjlWIfyi5ij909ji'
         access_token_secret= 'A4av2McT5BsDDMmM0hxZVp1gmWc1ciUStGuskKCfBGknr'

         auth = tw.OAuthHandler(consumer_key, consumer_secret)
         auth.set_access_token(access_token, access_token_secret)
         api = tw.API(auth, wait_on_rate_limit=True)
```

3) Searching for the tweets

```
In [100]: #searching tweets
          # Define the search term and the date_since date as variables
          search_words = "indian railway privatisation"
          date_since = "2020-05-01"
```

Interpretation:
Here we are searching for tweets related to Indian Railway Privatisation from the data 01st May 2020. To search these words we have created two objects as search_words and date_since respectively.

4) Collecting the tweets

```
In [101]: # Collect tweets
          tweets = tw.Cursor(api.search,
                             q=search_words,
                             lang="en",
                             since=date_since).items(1000)
          tweets

Out[101]: <tweepy.cursor.ItemIterator at 0x55c72cf488>
```

Interpretation:
Here we are collecting 1000 items or tweets on Indian Railway Privatisation from 01st May, 2020 using the Twitter API. And the language in which the tweets must be is mentioned as en which is English.

5) Listing down the tweets

```
In [102]: # Collect tweets
          tweets = tw.Cursor(api.search,
                             q=search_words,
                             lang="en",
                             since=date_since).items(1000)

          # Collect a list of tweets
          [tweet.text for tweet in tweets]

Out[102]: ['RT @TribalArmy: Indian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation https://t.co/0eRZx0aXMA',
           'RT @SaheeRa25385705: All India Agitation against Privatisation in Indian Railway \non Puna Pact Day, 24th Sept.,2020\n#StopPrivatisationSaveR…',
           'RT @VinayTi29822876: अब तो जग जाए मोदी जी\n\nIndian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation\n#25सितम्बर_…',
           'RT @basantchaturve9: Indian Railway is our pride, Stop privatisation\n#SaveRailwaySaveNation\n@PMOIndia @RailMinIndia  @PiyushGoyal @ShivaGop…',
           '@PMOIndia You hv also said in the BHU campus there will be no privatisation in Indian railway,but today we r seeing… https://t.co/6f0McaPBBv',
           'RT @basantchaturve9: Indian Railway is our pride, Stop privatisation\n#SaveRailwaySaveNation\n@PMOIndia @RailMinIndia  @PiyushGoyal @ShivaGop…',
           'RT @VinayTi29822876: अब तो जग जाए मोदी जी\n\nIndian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation\n#25सितम्बर_…',
           'RT @Sandeepnewsman: Indian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation\n#25सितम्बर_भारतबंद\nhttps://t.co/d6Kk…',
           'RT @mulnivaasi: Indian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation',
           'RT @VinayTi29822876: अब तो जग जाए मोदी जी\n\nIndian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySaveNation\n#25सितम्बर_…',
           'RT @RAJGOVINDVDV: अब तो जग जाए मोदी जी\n\nIndian Railway is our pride, \nStop Imidently privatisation.\n#SaveRailwaySave
```

Interpretation:
We have listed down the text of all the recent 1000 tweets which were made since 01st May, 2020. The recent 1000 tweets are being collected. All these tweets have a URL attached to them in order to specify the site. The tweets also include certain special characters which have to be removed.

6) Filtering tweets

```
In [103]: #filtering retweets
          new_search = search_words + " -filter:retweets"
          new_search

Out[103]: 'indian railway privatisation -filter:retweets'
```

Interpretation:
As we have noticed in the previous step there were certain special characters which were appearing in our tweets many times but had no meaning such as a smiley, hashtag etc. Our aim is to find the sentiments with the text involved in the tweets. So we are now removing all those irrelevant aspects in the tweet using filtering retweets option. Here we have created a new object named new_search in which we are storing the filtered tweets.

7) Viewing the filtered

```
In [104]: tweets = tw.Cursor(api.search,
                             q=new_search,
                             lang="en",
                             since=date_since).items(1000)

          [tweet.text for tweet in tweets]
```

```
Out[104]: ['@PMOIndia You hv also said in the BHU campus there will be no privatisation in Indian railway,but today we r seei
          ng… https://t.co/6f0McaPBBv',
           '@narendramodi sir AISCSTREA Agitations against PRIVATION  in Indian Railway "STOP PRIVATISATION"- Save for poor
          pe… https://t.co/DK8JXIlzPi',
           'Impact of privatisation by modi govt.\n1. Increased fual and gas prices.\n2. User paying monthly minimum price fo
          r mo… https://t.co/XDNvt320JO',
           '@SurajKrBauddh @BhimArmyChief All India Agitation against Privatisation in Indian Railway \non Puna Pact Day, 24t
          h S… https://t.co/86R1RDH0uG',
           '@narendramodi Plz stop privatisation in Indian railway',
           'SAVE NATION SAVE INDIAN \nRAILWAY \nSTOP PRIVATISATION OF INDIAN RAILWAYS \nPMO INDIA \nPIYUSHGOYAL  RAILWAY MINI
          STER… https://t.co/h31DPBbA4o',
           '@PiyushGoyalOffc respected sir with due respect I plead on behalf of Indian youth to give opportunity of jobs in
          ra… https://t.co/D7XxA7vz7m',
           'All India Agitation against Privatisation in Indian Railway \non Puna Pact Day, 24th Sept.,2020… https://t.co/8wc
          IawkTVe',
           '&gt;Indian railway, PSU, AIRPORT, AIR INDIA ka privatisation krwaenge.\n\nIndia developing country  s Under devel
          oping bnayenge',
           '@ANI The initial privatised part of Indian railways are station and coach cleaning contracts, catering, retiring
          ro… https://t.co/7snklM1jYC',
```

Interpretation:
Here we can see that all the irrelevant items (which are not useful for analysis) in the tweets have been removed such as special characters, hashtags etc. This helps us to clearly understand the tweets without any hindrances. Our focus lies only on the words in the tweets rather than being influenced by the special characters.

8) Finding the name and location of people tweeting

```
In [105]: #name and location of tweets
          tweets = tw.Cursor(api.search,
                             q=new_search,
                             lang="en",
                             since=date_since).items(1000)

          users_locs = [[tweet.user.screen_name, tweet.user.location] for tweet in tweets]
          users_locs
```

```
Out[105]: [['KISHOREJICONTR1', ''],
           ['RadhaRamanRana1', ''],
           ['RuhJain7990', ''],
           ['SagarDeepak15', 'Rajasthan, India'],
           ['ssmr1984', 'Jaipur, India'],
           ['RajaniK13899528', 'Varanasi '],
           ['SarikaJ39573528', 'अजमेर, भारत'],
           ['9ed028d0ad1943a', ''],
           ['mritunjay92', 'Noida, India'],
           ['SHARMAR09415918', ''],
           ['SaheeRa25385705', ''],
           ['BimalChanda10', ''],
           ['PHOOLSINGHBAIR5', 'kazipet , Warangal'],
           ['Ramashi07224317', 'Sheikhpura, India'],
           ['Ramashi07224317', 'Sheikhpura, India'],
           ['JayTiwa48025890', ''],
           ['AmarSahani20795', 'Bhadohi, India'],
           ['TheUnitedIndia2', ''],
           ['ssmr1984', 'Jaipur, India'],
```

Interpretation:
The names of the users are on the left side along with their locations on the right. Here is the list of tweets various locations of India namely Jaipur, Udaipur, Noida, Delhi, Varanasi, Ahmedabad, Kalyani, Warangal, Hyderabad, Kolkata, Rewa (Madhya Pradesh) etc. We can see that Privatisation of Indian Railways has impacted the whole Nation and not only people of a particular states are mentioning about it. The Whole Nation is talking about the same.

9) Listing them in Dataframe

```
In [106]: #create pandas dataframe
          tweet_text = pd.DataFrame(data=users_locs,
                                    columns=['user', "location"])
          tweet_text
```

Out[106]:

	user	location
0	KISHOREJICONTR1	
1	RadhaRamanRana1	
2	RuhJain7990	
3	SagarDeepak15	Rajasthan, India
4	ssmr1984	Jaipur, India
...	...	...
379	shiva_congress	Hyderabad, India
380	AfzalKhan_INC	Gujarat, India
381	OfficialRenju	
382	NayanJain_25	Madhya Pradesh, India
383	arunplakkat	Palakkad / Bengaluru

384 rows × 2 columns

Interpretation:
We have converted all the user's name and location in a structured format of rows and columns called as dataframe for better and easy understanding.

B) FINDING THE FREQUENCY OF WORDS

1) Removing URL from Tweets

```
In [108]: #removing URL
          def remove_url(txt):
              """Replace URLs found in a text string with nothing
              (i.e. it will remove the URL from the string).

              Parameters
              ----------
              txt : string
                  A text string that you want to parse and remove urls.

              Returns
              -------
              The same txt string with url's removed.
              """

              return " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", txt).split())
```

Interpretation:
As we have to go ahead with finding the frequency of words in the tweets. The first step involved is removing all the URL's attached to the tweets.

2) Listing down the tweets with no URL's

```
In [109]: #clean tweets
          all_tweets_no_urls = [remove_url(tweet) for tweet in all_tweets]
          all_tweets_no_urls[:]

Out[109]: ['PMOIndia You hv also said in the BHU campus there will be no privatisation in Indian railwaybut today we r seein
          g',
           'narendramodi sir AISCSTREA Agitations against PRIVATION in Indian Railway STOP PRIVATISATION Save for poor pe',
           'Impact of privatisation by modi govt1 Increased fual and gas prices2 User paying monthly minimum price for mo',
           'SurajKrBauddh BhimArmyChief All India Agitation against Privatisation in Indian Railway on Puna Pact Day 24th S',
           'narendramodi Plz stop privatisation in Indian railway',
           'SAVE NATION SAVE INDIAN RAILWAY STOP PRIVATISATION OF INDIAN RAILWAYS PMO INDIA PIYUSHGOYAL RAILWAY MINISTER',
           'PiyushGoyalOffc respected sir with due respect I plead on behalf of Indian youth to give opportunity of jobs in r
          a',
           'All India Agitation against Privatisation in Indian Railway on Puna Pact Day 24th Sept2020',
           'gtIndian railway PSU AIRPORT AIR INDIA ka privatisation krwaengeIndia developing country s Under developing bnaye
          nge',
           'ANI The initial privatised part of Indian railways are station and coach cleaning contracts catering retiring r
```

Interpretation:
Here we are creating a new object named all_tweets_no_urls which stores all the tweets with no URL's.

3) Splitting each word of tweet

```
In [110]: # Split the words from one tweet into unique elements
          all_tweets_no_urls[0].lower().split()

Out[110]: ['pmoindia',
           'you',
           'hv',
           'also',
           'said',
           'in',
           'the',
           'bhu',
           'campus',
           'there',
           'will',
           'be',
           'no',
           'privatisation',
           'in',
           'indian',
           'railwaybut',
           'today',
           'we',
           'r',
           'seeing']
```

Interpretation:
Here we are taking the 1st tweet with index as 0. All the words in the 1st tweet are listed. This is just a method to understand how python will be working for finding out the frequency of the words in all tweets. We are getting the words in lower cases.

4) Another way of getting words in tweet

```
In [111]: # Create a list of lists containing lowercase words for each tweet
          words_in_tweet = [tweet.lower().split() for tweet in all_tweets_no_urls]
          words_in_tweet[:2]

Out[111]: [['pmoindia',
            'you',
            'hv',
            'also',
            'said',
            'in',
            'the',
            'bhu',
            'campus',
            'there',
            'will',
            'be',
            'no',
            'privatisation',
            'in',
            'indian',
            'railwaybut',
            'today',
            'we',
            'r',
            'seeing'],
           ['narendramodi',
            'sir',
            'aiscstrea',
            'agitations',
            'against',
            'privation',
            'in',
            'indian',
            'railway',
```

Interpretation:
Here we can see that we are getting first two tweets from all the tweets and splitting their words.

5) Creating counts of the words

```
In [112]: # List of all words across tweets
          all_words_no_urls = list(itertools.chain(*words_in_tweet))

          # Create counter
          counts_no_urls = collections.Counter(all_words_no_urls)

          counts_no_urls.most_common(20)
```

```
Out[112]: [('indian', 359),
           ('railway', 352),
           ('is', 251),
           ('stop', 228),
           ('our', 222),
           ('pride', 204),
           ('privatisation', 138),
           ('of', 103),
           ('imidently', 97),
           ('piyushgoyal', 94),
           ('privatisationsaverailwaysavenation', 80),
           ('railminindia', 79),
           ('in', 74),
           ('the', 72),
           ('india', 63),
           ('privatisationsaverailwaysavenationpmoindia', 55),
           ('against', 52),
           ('shivagopalmish1', 48),
           ('saverailwaysavenation', 44),
           ('on', 42)]
```

Interpretation:
Here we are creating a list of all tweets without URL by creating object named all_words_no_urls. Then for counting the frequency of words we are creating an object named counts_no_urls. Then we are listing down first 20 words with the highest frequency. Therefore, we come across words such as stop, saverailwaysavenation, against etc. But here we can see that there are various words which are common and do not convey a particular meaning to us such as is,our,of,in the etc. These are called stopwords. Therefore we go ahead with the process of removing stop words.

6) Creating dataframe for counts

```
In [113]: clean_tweets_no_urls = pd.DataFrame(counts_no_urls.most_common(20),
                                              columns=['words', 'count'])

          clean_tweets_no_urls
```

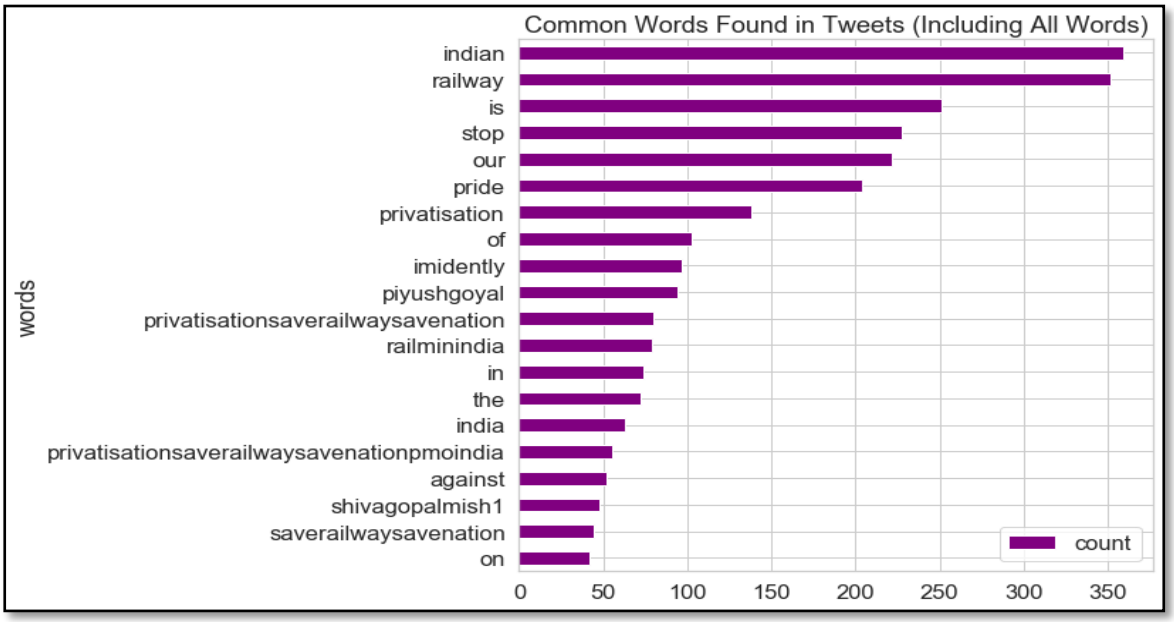
Out[113]:

	words	count
0	indian	359
1	railway	352
2	is	251
3	stop	228
4	our	222
5	pride	204
6	privatisation	138
7	of	103
8	imidently	97
9	piyushgoyal	94
10	privatisationsaverailwaysavenation	80
11	railminindia	79
12	in	74
13	the	72
14	india	63

Interpretation:
We are creating a dataframe of most frequently appearing words.

7) Visualisation

```
In [114]: #Common Words Plot
          fig, ax = plt.subplots(figsize=(8, 8))

          # Plot horizontal bar graph
          clean_tweets_no_urls.sort_values(by='count').plot.barh(x='words',
                                y='count',
                                ax=ax,
                                color="purple")

          ax.set_title("Common Words Found in Tweets (Including All Words)")

          plt.show()
```



Interpretation:
Here we are visualising the first 20 frequently appearing words. But as we can see we also have words such as is, our, of, in the etc. These are called stopwords. Therefore we go ahead with the process of removing stop words so that we can get a clear understanding and do not deviate from the objective of the analysis.

8) Identifying and listing stopwords

```
In [115]: #removing stopwords
          nltk.download('stopwords')

          [nltk_data] Downloading package stopwords to
          [nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data...
          [nltk_data]   Package stopwords is already up-to-date!

Out[115]: True

In [116]: stop_words = set(stopwords.words('english'))

          # View a few words from the set
          list(stop_words)[0:10]

Out[116]: ['who', 'in', "didn't", 'was', 'that', 'these', 'no', "you've", 'our', 'on']
```

Interpretation:
Here we can see that nltk library is being used to identify, list and remove stopwords. For this we are creating object named stopwords and listing first 10 words.

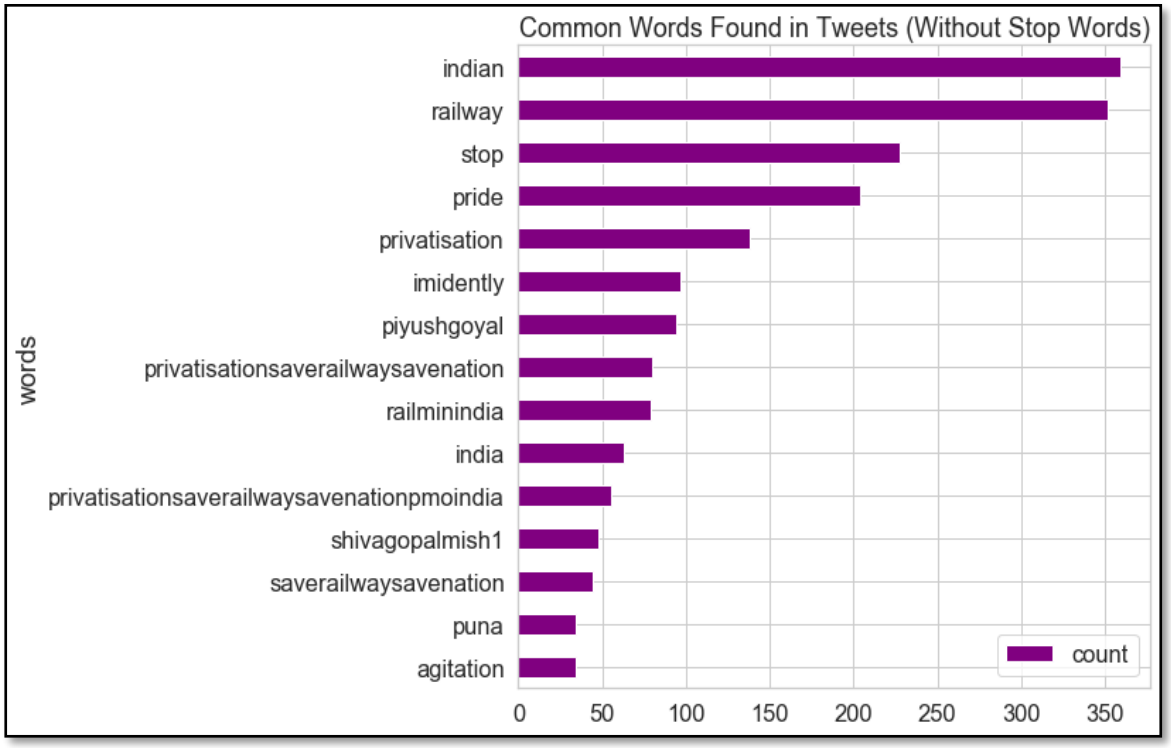
9) Verifying whether removed

```
In [117]: #words in tweet
          words_in_tweet[0]

Out[117]: ['pmoindia',
           'you',
           'hv',
           'also',
           'said',
           'in',
           'the',
           'bhu',
           'campus',
           'there',
           'will',
           'be',
           'no',
           'privatisation',
           'in',
           'indian',
           'railwaybut',
           'today',
           'we',
           'r',
           'seeing']
```

Interpretation:
Here we are printing the 1st tweet and seeing if it is clean or not. Therefore, we can see that no stop words are there.

10) Removing stopwords from all tweets and creating final visualisation

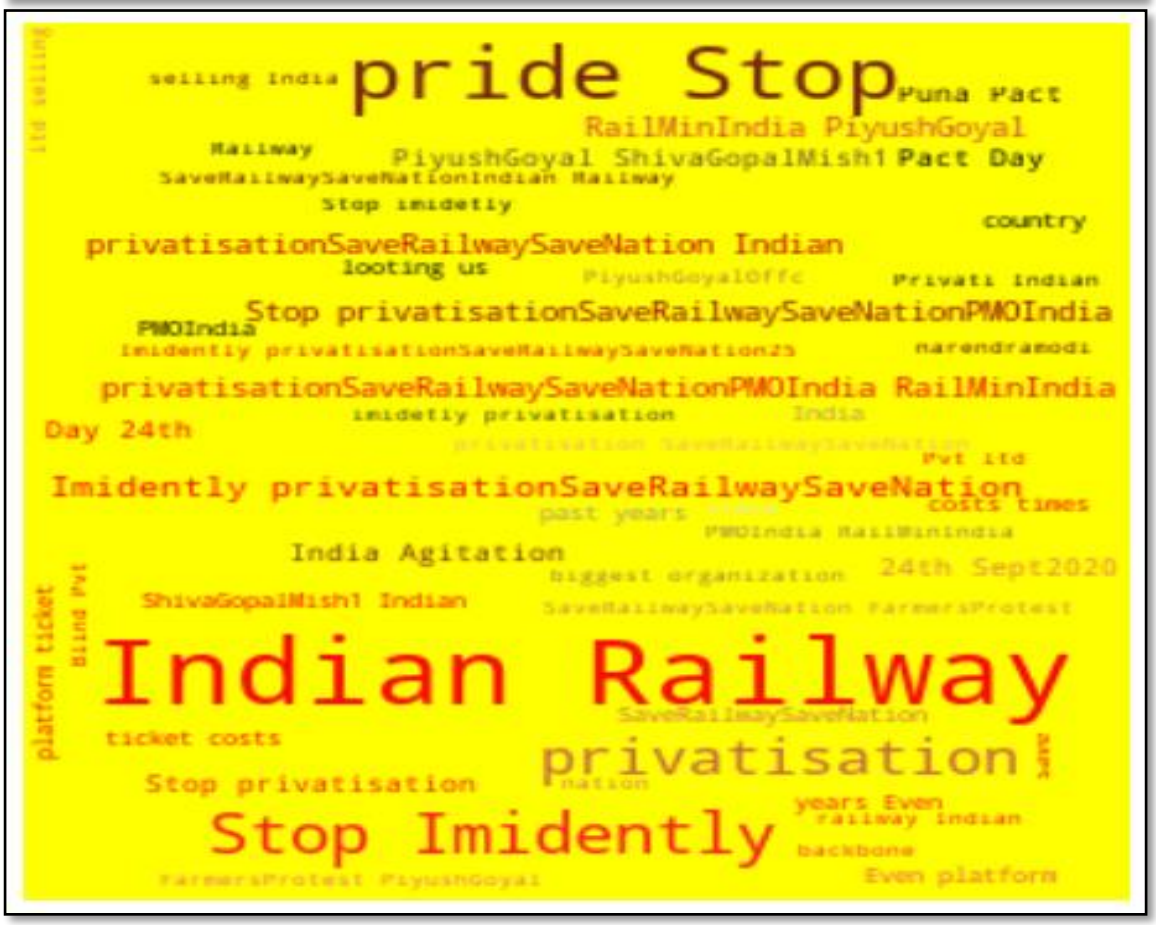
```
In [119]: #removing stopwords
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
tweets_nsw = [[word for word in tweet_words if not word in stop_words]
              for tweet_words in words_in_tweet]

tweets_nsw[0]

#removing collection words
all_words_nsw = list(itertools.chain(*tweets_nsw))

counts_nsw = collections.Counter(all_words_nsw)

counts_nsw.most_common(15)

clean_tweets_nsw = pd.DataFrame(counts_nsw.most_common(15),
                                columns=['words', 'count'])

fig, ax = plt.subplots(figsize=(8, 8))

# Plot horizontal bar graph
clean_tweets_nsw.sort_values(by='count').plot.barh(x='words',
                      y='count',
                      ax=ax,
                      color="purple")

ax.set_title("Common Words Found in Tweets (Without Stop Words)")

plt.show()

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```



Interpretation:
From here we can see that words such as Indian, railway, saverailwaysavenation, stop, agitation, pride all indicate that people have a negative thinking towards Indian Railway Privatisation. For further we also plot a word cloud for more clarity.

11) Plotting a word cloud

```
In [140]: #plot the wordcloud
          allwords = ' '.join([tweet for tweet in all_tweets_no_urls])
          mask = np.array(Image.open("Desktop/mickey.jpg"))
          color= ImageColorGenerator(mask)
          wordcloud = WordCloud(width=400, height=400,
                                max_words=50,mask=mask, stopwords=STOPWORDS,   background_color ='yellow', random_state=42).genera
          plt.figure(figsize = (8, 10), facecolor = None)
          plt.imshow(wordcloud.recolor(color_func=color),interpolation="bilinear")
          plt.axis("off")
          plt.show()
```



Interpretation:

From the word clod output we get a clarity that majority people all over India are not happy with Railway Privatisation. The common words include a new campaign named saverailwayssavenation and also words such as agitation and stop privatisation which further explains the same. It is clearly seen that people of India believe Railways are their pride and are against railway privatisation. They are also addressing some eminent personalities like Our Honorable PM of India as PMOIndia and Railway Minister Piyush Goyal. There are also words such as ticket costs as people feel that if privatisation takes place it will make tickets more expensive.

C) SENTIMENT ANALYSIS

1) Creating polarity values for each tweet

```
In [144]: # Create textblob objects of the tweets
          sentiment_objects = [TextBlob(tweet) for tweet in all_tweets_no_urls]
```

```
In [146]: # Create list of polarity valuesx and tweet text
          sentiment_values = [[tweet.sentiment.polarity, str(tweet)] for tweet in sentiment_objects]

          sentiment_values
```

```
Out[146]: [[0.0,
            'PMOIndia You hv also said in the BHU campus there will be no privatisation in Indian railwaybut today we r seein
          g'],
           [-0.4,
            'narendramodi sir AISCSTREA Agitations against PRIVATION in Indian Railway STOP PRIVATISATION Save for poor pe'],
           [0.0,
            'Impact of privatisation by modi govt1 Increased fual and gas prices2 User paying monthly minimum price for mo'],
           [0.0,
            'SurajKrBauddh BhimArmyChief All India Agitation against Privatisation in Indian Railway on Puna Pact Day 24th
          S'],
           [0.0, 'narendramodi Plz stop privatisation in Indian railway'],
           [0.0,
            'SAVE NATION SAVE INDIAN RAILWAY STOP PRIVATISATION OF INDIAN RAILWAYS PMO INDIA PIYUSHGOYAL RAILWAY MINISTER'],
           [-0.125,
            'PiyushGoyalOffc respected sir with due respect I plead on behalf of Indian youth to give opportunity of jobs in
          ra'],
           [0.0,
            'All India Agitation against Privatisation in Indian Railway on Puna Pact Day 24th Sept2020'],
           [0.0,
```

Interpretation:
Here we are identifying the polarity values for each tweet where majority of tweets have negative polarity.

2) Creating dataframe

```
In [147]: #converting into dataframe
          sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity", "tweet"])

          sentiment_df
```

Out[147]:

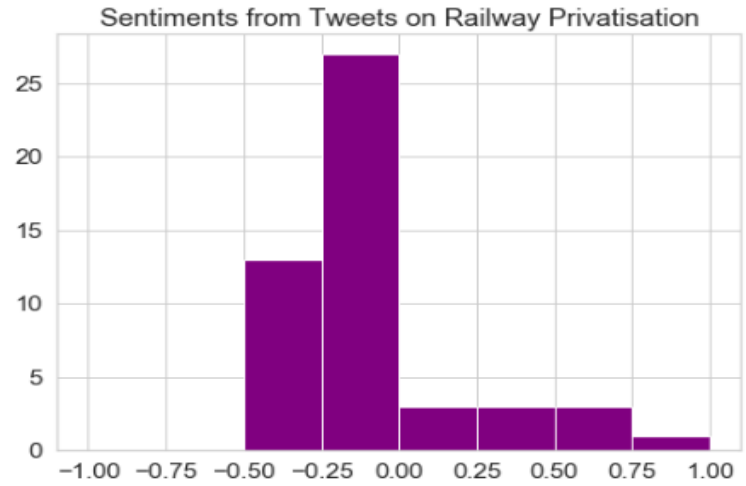
	polarity	tweet
0	0.00	PMOIndia You hv also said in the BHU campus th...
1	-0.40	narendramodi sir AISCSTREA Agitations against ...
2	0.00	Impact of privatisation by modi govt1 Increase...
3	0.00	SurajKrBauddh BhimArmyChief All India Agitatio...
4	0.00	narendramodi Plz stop privatisation in Indian ...
...	...	...
379	-0.25	Indian Railway has been looting us for the pas...
380	-0.25	Indian Railway has been looting us for the pas...
381	-0.25	Indian Railway has been looting us for the pas...
382	-0.25	Indian Railway has been looting us for the pas...
383	-0.25	Indian Railway has been looting us for the pas...

384 rows × 2 columns

Interpretation:
Now we are creating dataframe of tweets witheir respective polarity values for making it more clear and easy to interpret.

3) Removing polarity with 0 and plotting histogram

```
In [148]: # Remove polarity values equal to zero
          sentiment_df = sentiment_df[sentiment_df.polarity != 0]
          fig, ax = plt.subplots(figsize=(8, 6))

          # Plot histogram with break at zero
          sentiment_df.hist(bins=[-1, -0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1],
                            ax=ax,
                            color="purple")

          plt.title("Sentiments from Tweets on Railway Privatisation")
          plt.show()
```



Interpretation:
The Histogram shows the polarity of tweets. Here majority of the tweets have negative polarity. The highest number of tweets lies between the polarity ranges of 0 to -0.25. If the tweets have polarity value positive which shows that people have positive sentiments. And if polarity is negative, it shows that people have negative sentiments towards the subject. Here we can come with a conclusion that people have negative sentiments towards the Railway Privatisation.

4) Finding complete sentiment

```
In [149]: testimonial = TextBlob(str(sentiment_objects))

In [150]: testimonial.sentiment
Out[150]: Sentiment(polarity=-0.10875000000000001, subjectivity=0.3475657894736843)

In [151]: testimonial.sentiment.polarity
Out[151]: -0.10875000000000001

In [152]: testimonial.sentiment.subjectivity
Out[152]: 0.3475657894736843
```

Interpretation:
Here we find the polarity and subjectivity of the tweets which comes out to be as -0.108 and 0.34 respectively. Here we come to a conclusion clearly that majority of people have negative sentiments towards Railway Privatisation.

CONCLUSION:

From this we can conclude that majority of people all over India have a negative sentiment towards Railway Privatisation. They have come up with a campaign named SaveRailwaySaveNation on Twitter. TheY considered Indian Railways as their pride. They fear that Railway Privatisation would make travelling expensive because of hike in ticket prices by Corporates handling railways.