Name – Himanshu Soni

Student ID – 29389089

Tutor – Xue Sun(Erik)

FIT5147 - Assignment 1

Topic – Exploratory analysis of all CNN's tweets on Twitter over the past year using Latent dirichlet allocation(LDA).

Problem description/Aim – My aim is to collect, wrangle and analyze all the tweets posted by CNN on twitter in the past year and answer some general questions which I(and hopefully others as well) have regarding CNN's tweets.

Some of the questions I wish to answer –

1. Does the timing of a tweet affect the number of likes/retweets it receives?
2. Does a particular topic/keywords gets you more likes/retweets than others?
3. Some people say that word 'Trump' is a clickbait for CNN and I wish to find out if it's true.
4. Is CNN obsessed with TRUMP?

Motivation - It is estimated that there are millions of tweets generated every day on twitter which gives us data scientists enormous amount of data to analyze. This analysis can help us in understanding the user behavior and interests which can be used for targeted advertising.

CNN has over 42 million followers worldwide covering plethora of topics in their tweets. They also tweet quite regularly with a total of 238000 tweets since it's inception on twitter. This is the reason I choose them for this explanatory project.

This analysis would also help me analyzing future data as well as understand the concepts of NLP which I can use for applications such as targeted advertising.

DATA WRANGLING

Step 1 – DATA collection –

- My first step involved collecting all the tweets and the information related to them such as date, time, likes and retweets for each of them.
- For this I developed a script in python using SELENIUM and CHROMEDRIVER packages which captured the ID's of the tweets autonomously over a period of 10 hours.
- Each tweet ID consists of an 18 digits number like 818154306004287488 with there being over 22,000 tweet ID's collected in this step.

Twitter API

- Twitter has an API which can be used to get the information about a tweet after setting the API and the following details.

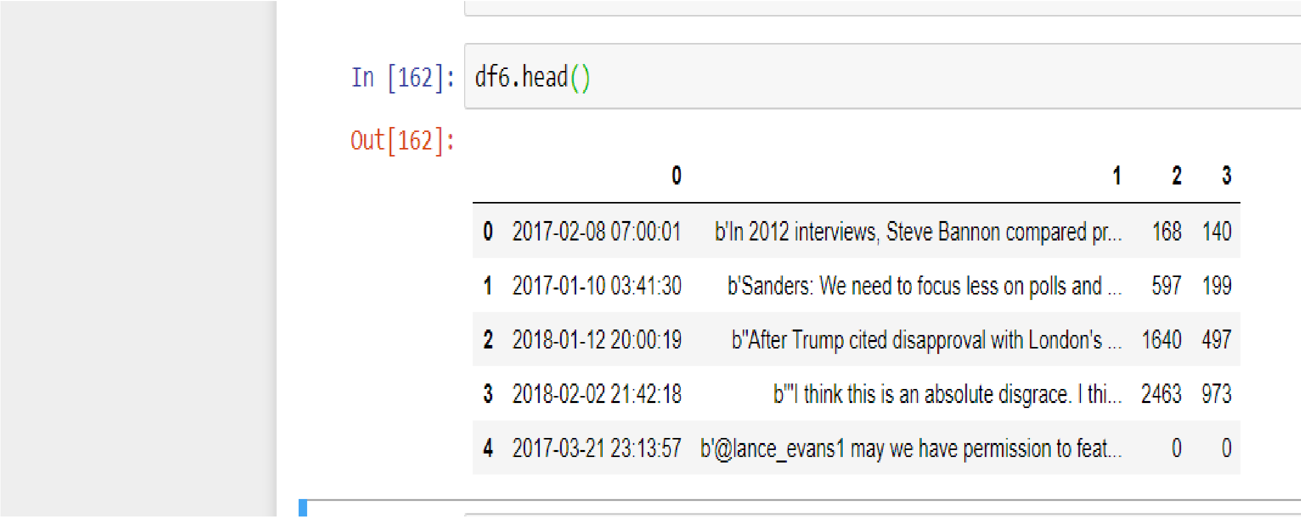
  CONSUMER_KEY or API key
  CONSUMER_SECRET or API secret
  OAUTH_TOKEN
  OAUTH_TOKEN_SECRET.

- Then I used the twitter's API and the TWEEPY package in python to capture the required info about the tweets listed above.
  The function used was

  tweepy.API(auth).get_status(int(id_no))

- This again takes around 10 hours to capture all the information as the tweepy API can run for just 15 mins at a time and has to be put to sleep every 15 mins.

- This was then stored in a csv file using pandas which looked like this (total 22000 rows)



```
In [162]: df6.head()
Out[162]:
```

	0	1	2	3
0	2017-02-08 07:00:01	b'In 2012 interviews, Steve Bannon compared pr...	168	140
1	2017-01-10 03:41:30	b'Sanders: We need to focus less on polls and ...	597	199
2	2018-01-12 20:00:19	b"After Trump cited disapproval with London's ...	1640	497
3	2018-02-02 21:42:18	b"'I think this is an absolute disgrace. I thi...	2463	973
4	2017-03-21 23:13:57	b'@lance_evans1 may we have permission to feat...	0	0

Where column 0 – datetime, 1 – tweet, 2-likes, 3- retweets for each tweet.

Step 2 - Data wrangling

In this step, I added three more columns namely TOPIC, TIME_SETTING and MENTIONS_TRUMP.

i. TOPIC – for getting the topic of the tweet, I first generated a regular expression or RE for rejecting the garbage values from the tweets text.

Example of a tweet - 'b\'police officer stands to lose job after he was caught on dashcam footage telling a woman "we only kill black people\\xe2\\x80\\xa6 https://t.co/i6youjpcsr\''

The regular expression I used was

Pattern = ‘(\\\\xe2\\\\\w{3}\\\\\w{3}|https://[a-zA-Z0-9./]+)’ and using it with re.sub(pattern,’’,string) and re.findall(r’\w+’,string), I was able to clean the data.

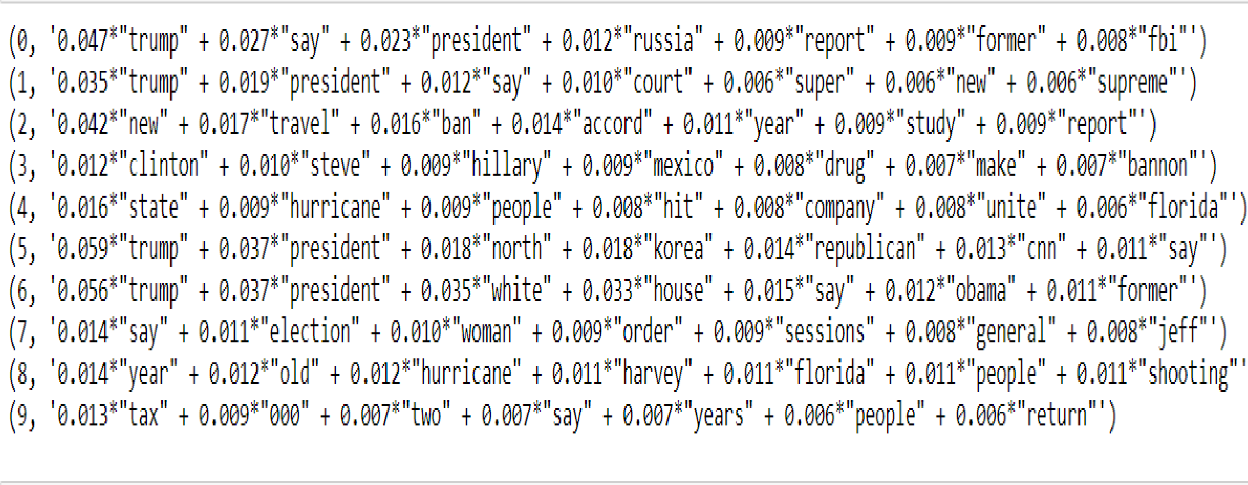
Example after cleaning - ['b','police','officer','stands','to','lose','job','after','he','was','caught','on','dashcam','footage','telling','a','woman','we','only','kill','black','people'].

I was thus able to generate the corpus for all the tweets after performing the following steps using the NLTK package in python

- Stop word removal – removing all stopwords and words shorter than two characters
- Stemmatization and lemmatization – converting the words to their base form.

The I proceeded to use latent dirichlet allocation(LDA) to generate the 10 topics (20 passes to bifurcate the tweets into.

The topics were



```
(0, '0.047*"trump" + 0.027*"say" + 0.023*"president" + 0.012*"russia" + 0.009*"report" + 0.009*"former" + 0.008*"fbi"')
(1, '0.035*"trump" + 0.019*"president" + 0.012*"say" + 0.010*"court" + 0.006*"super" + 0.006*"new" + 0.006*"supreme"')
(2, '0.042*"new" + 0.017*"travel" + 0.016*"ban" + 0.014*"accord" + 0.011*"year" + 0.009*"study" + 0.009*"report"')
(3, '0.012*"clinton" + 0.010*"steve" + 0.009*"hillary" + 0.009*"mexico" + 0.008*"drug" + 0.007*"make" + 0.007*"bannon"')
(4, '0.016*"state" + 0.009*"hurricane" + 0.009*"people" + 0.008*"hit" + 0.008*"company" + 0.008*"unite" + 0.006*"florida"')
(5, '0.059*"trump" + 0.037*"president" + 0.018*"north" + 0.018*"korea" + 0.014*"republican" + 0.013*"cnn" + 0.011*"say"')
(6, '0.056*"trump" + 0.037*"president" + 0.035*"white" + 0.033*"house" + 0.015*"say" + 0.012*"obama" + 0.011*"former"')
(7, '0.014*"say" + 0.011*"election" + 0.010*"woman" + 0.009*"order" + 0.009*"sessions" + 0.008*"general" + 0.008*"jeff"')
(8, '0.014*"year" + 0.012*"old" + 0.012*"hurricane" + 0.011*"harvey" + 0.011*"florida" + 0.011*"people" + 0.011*"shooting"'
(9, '0.013*"tax" + 0.009*"000" + 0.007*"two" + 0.007*"say" + 0.007*"years" + 0.006*"people" + 0.006*"return"')
```

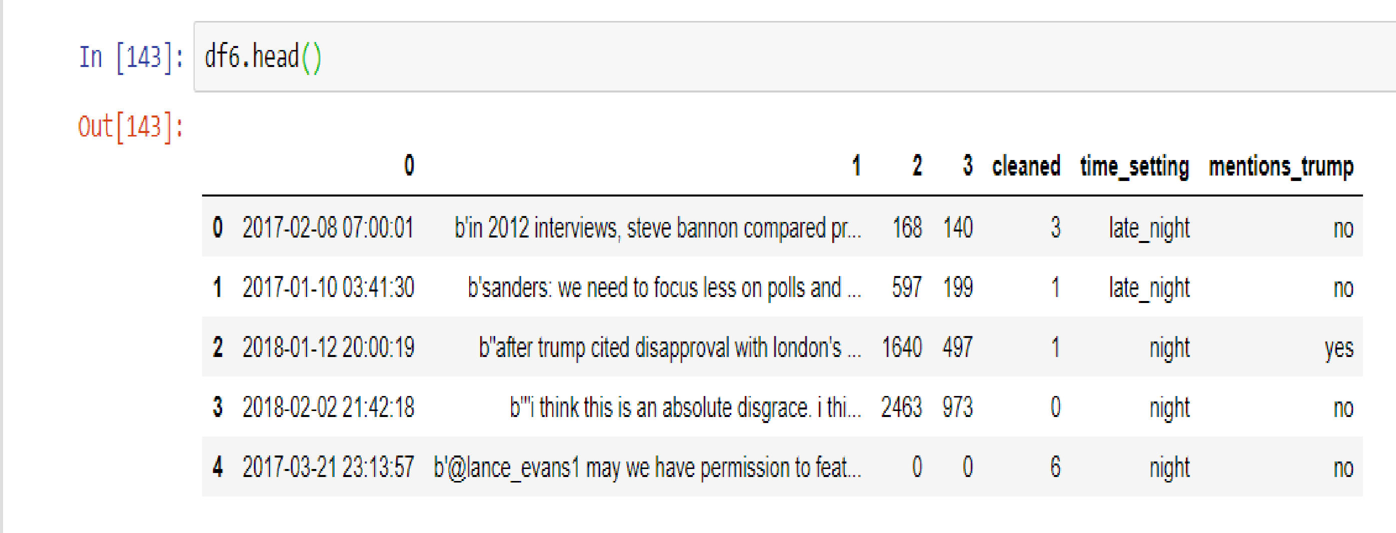
Thus, each tweet was given a topic using this algorithm.

- TIME SETTING - The aim is to group the tweets according the day setting. The categories are Morning – 8AM-12 PM, Afternoon – 12PM – 4PM, Evening – 4 PM to 8 PM, Night - 8 PM to 12AM and Late night – 12AM to 8 AM.

This was accomplished by utilizing the datetime attribute ex - 2017-02-08 07:00:01 which is categorized to late night.

- MENTIONS TRUMP – This column's value is 1 if the word trump appears in it and 0 otherwise.

After the wrangling steps, the data looks like this.(colum 2 -likes,3- retweets,cleaned-mentions topic)



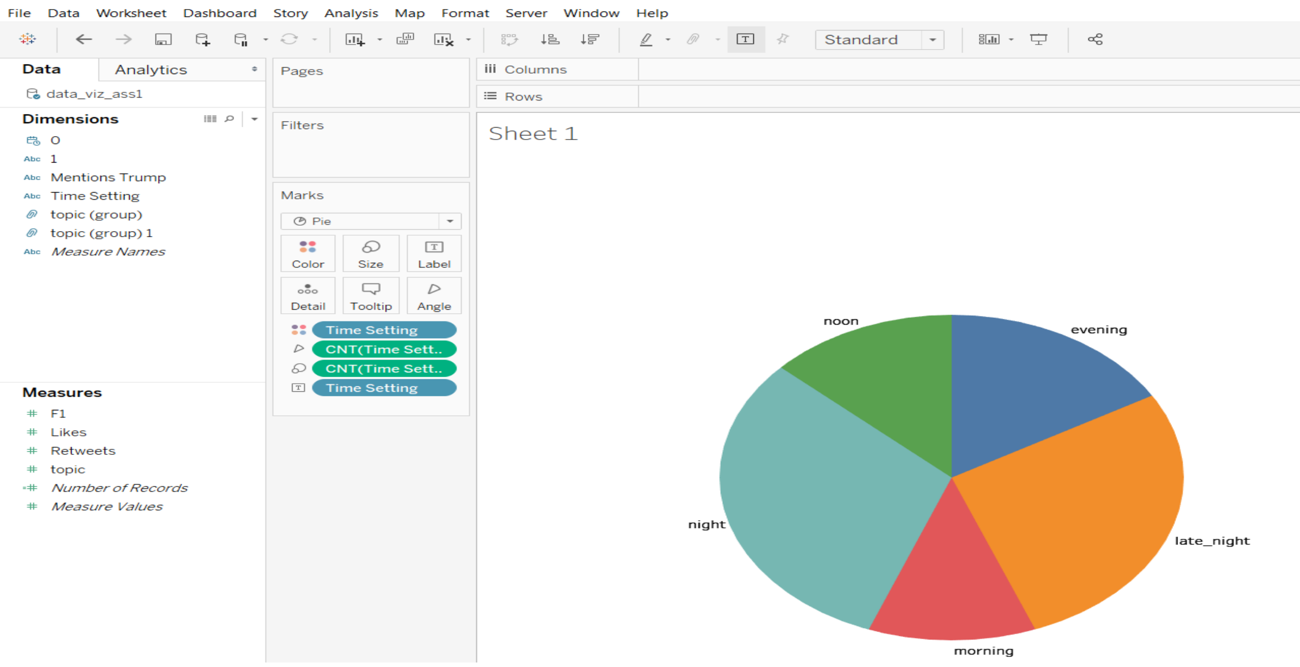
```
In [143]: df6.head()
```

Out[143]:

	0	1	2	3	cleaned	time_setting	mentions_trump
0	2017-02-08 07:00:01	b'in 2012 interviews, steve bannon compared pr...	168	140	3	late_night	no
1	2017-01-10 03:41:30	b'sanders: we need to focus less on polls and ...	597	199	1	late_night	no
2	2018-01-12 20:00:19	b"after trump cited disapproval with london's ...	1640	497	1	night	yes
3	2018-02-02 21:42:18	b"'i think this is an absolute disgrace. i thi...	2463	973	0	night	no
4	2017-03-21 23:13:57	b'@lance_evans1 may we have permission to feat...	0	0	6	night	no

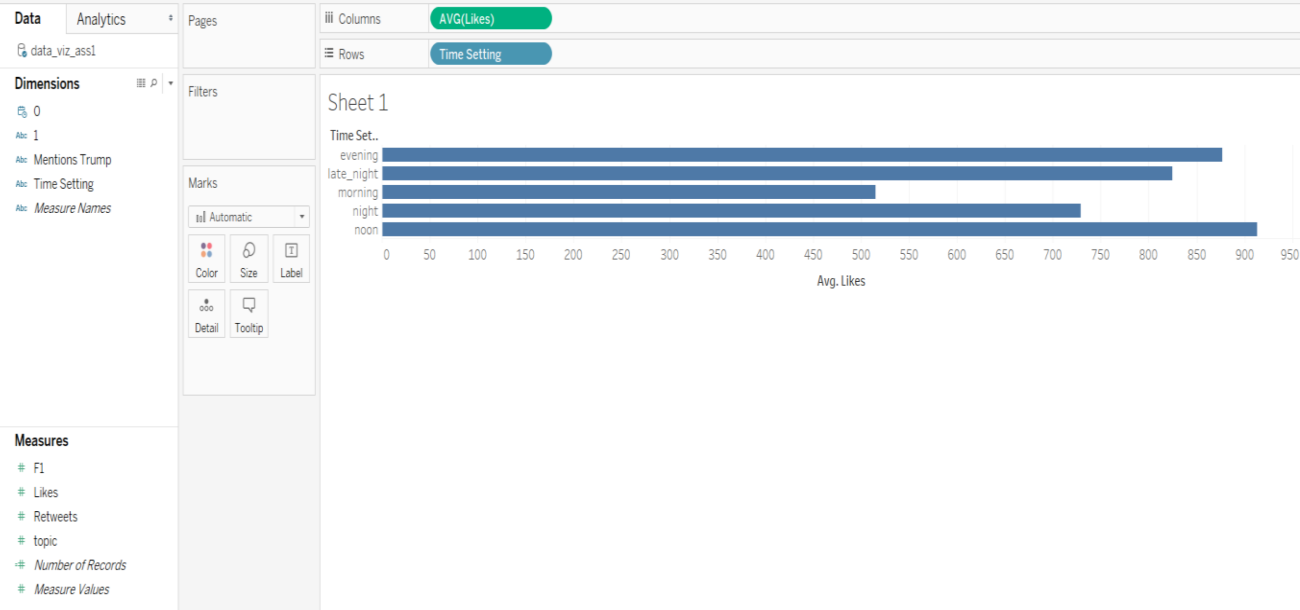
DATA EXPLORATION

1. Impact of time –
   Firstly, we see in which time setting most tweets are tweeted using a pie chart in tableau.



This shows that most of the tweets are tweeted during the night and late night categories i.e between 8PM to 8AM.

Now let's see which category gets the most average likes and retweets

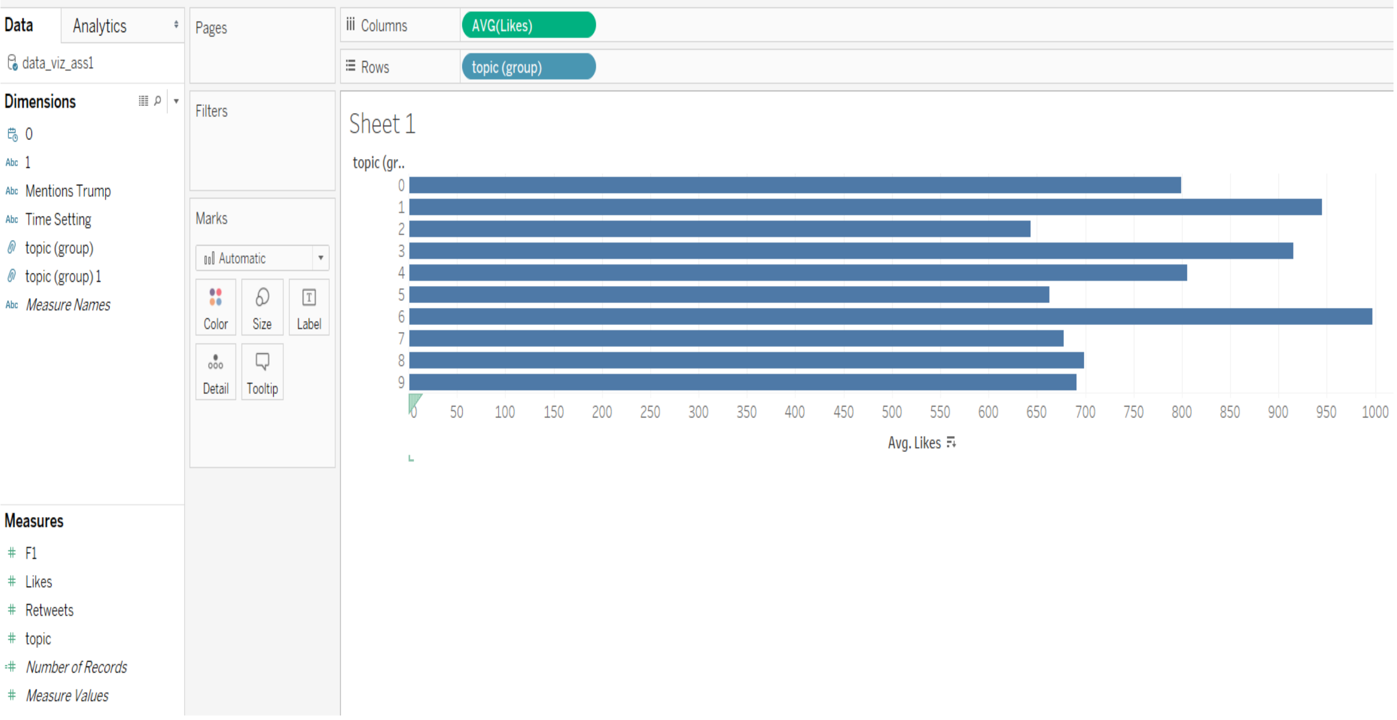


Columns AVG(Likes)

Rows Time Setting

This bar graph shows that on average a tweet in the noon receives twice the number of likes than the tweets in the morning.

2. Impact of topic – All the tweets have been categorized into 10 topics. Let's see it's impact.

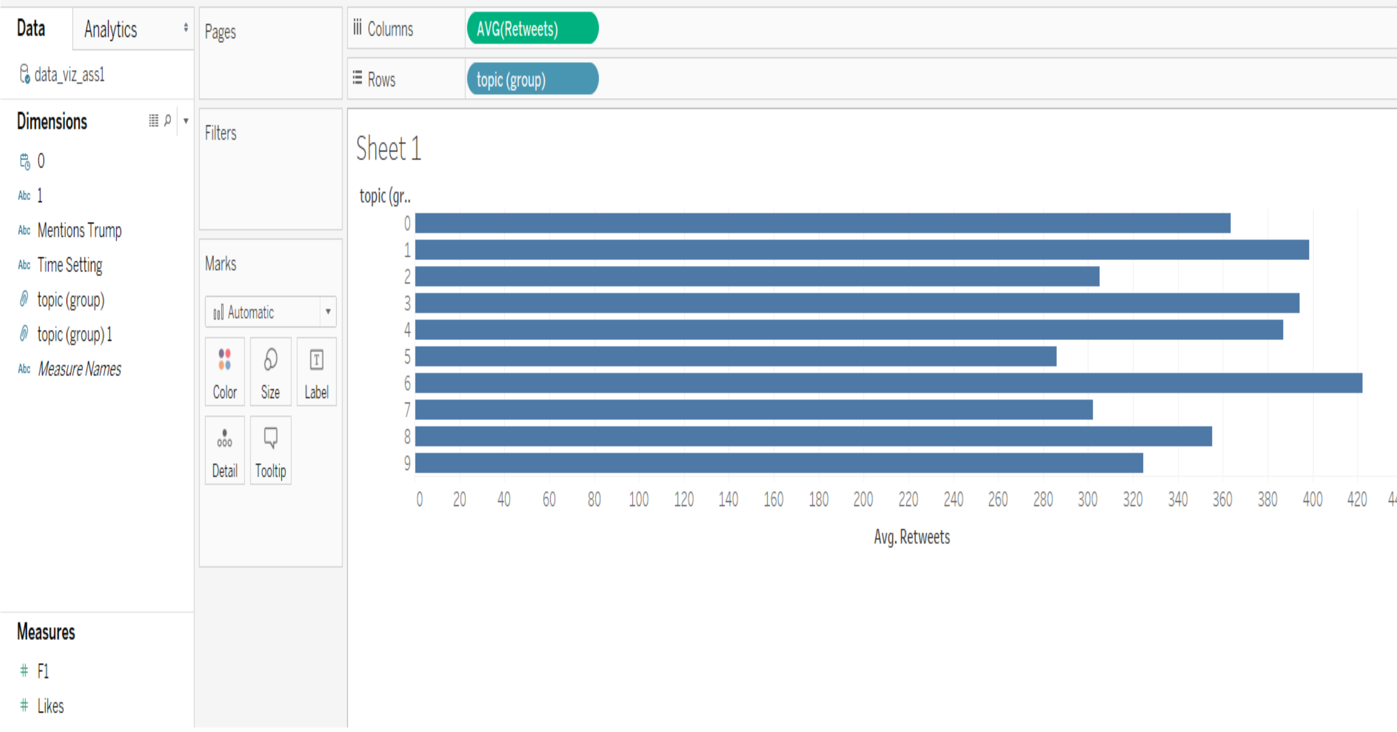


Columns AVG(Likes)

Rows topic (group)

From the graph it is evident that topic 1, 3 and 6 receives the most number of likes on average.

If we see the words associated from these topics from the previous page, the words are ‘TRUMP,’PRESIDENT’,’HILLARY’,’CLINTON’,’WHITE’,’HOUSE’. So it can be said that Political news gains the most number of likes on average.



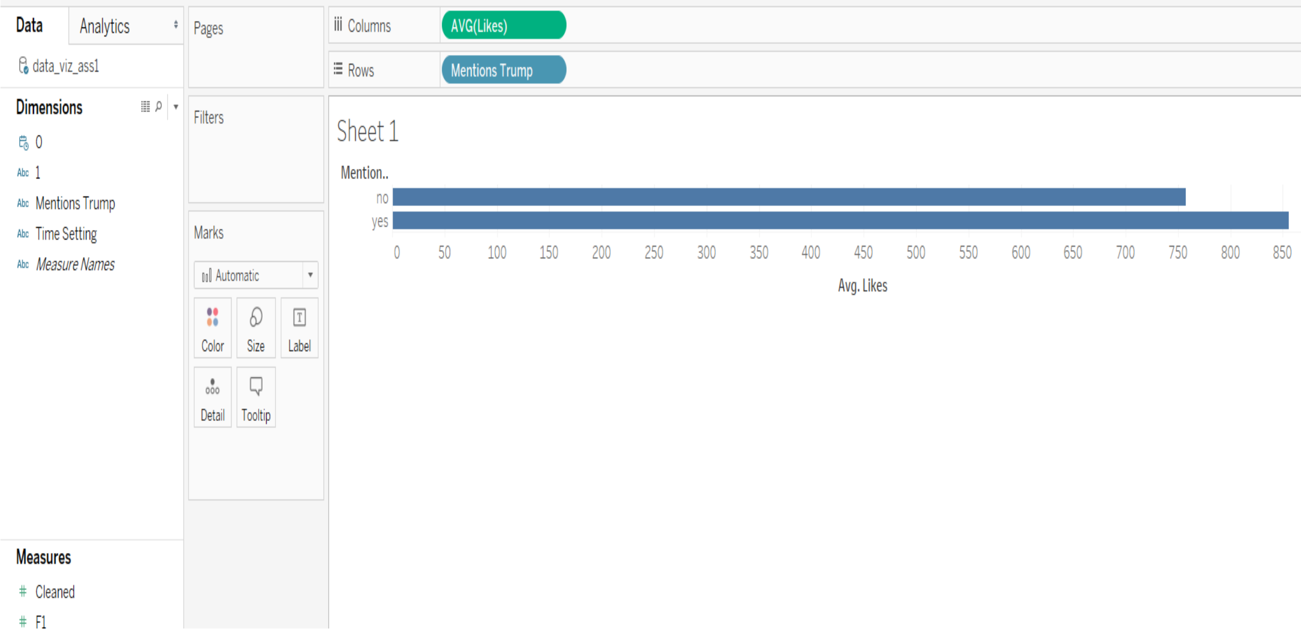
Data | Analytics
data_viz_ass1
Dimensions: 0, 1, Mentions Trump, Time Setting, topic (group), topic (group) 1, Measure Names
Measures: F1, Likes
Columns: AVG(Retweets)
Rows: topic (group)
Marks: Automatic

Similar trend can be seen across average retweets per group.

3. Is the word ‘TRUMP’ a clickbait?

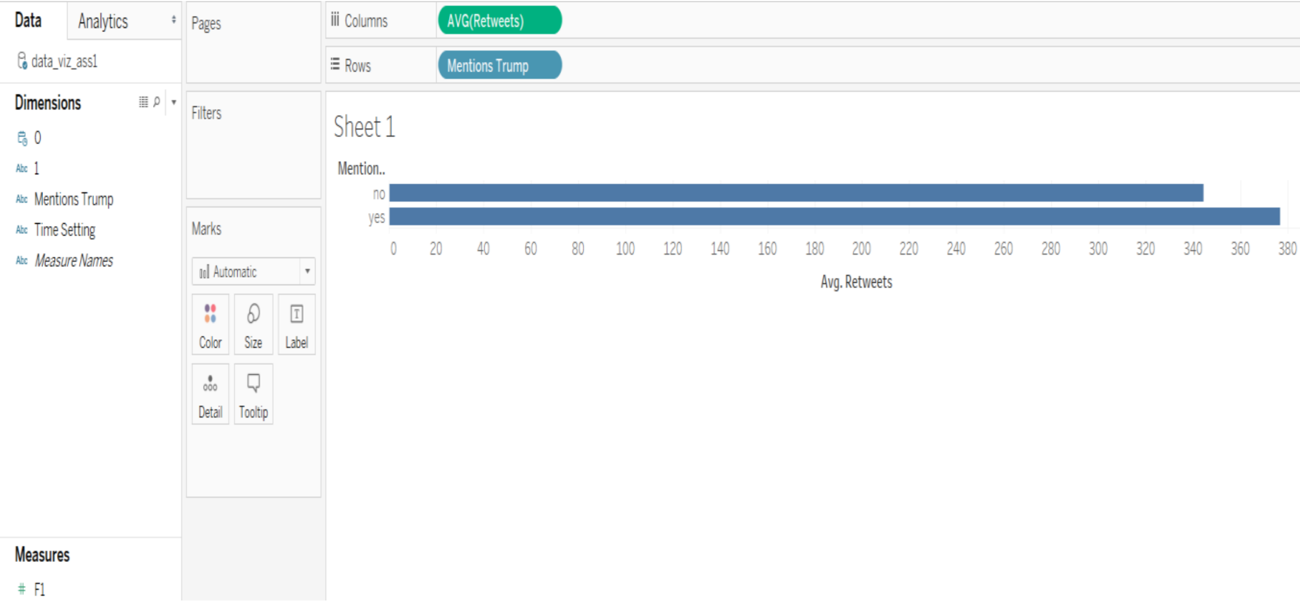
   Clickbait is a text or thumbnail link that is designed to entice users to follow that link and read, view, or listen to the linked piece of online content.

   To investigate this, lets see the average likes a tweet receives If it has the word ‘trump’ in it.



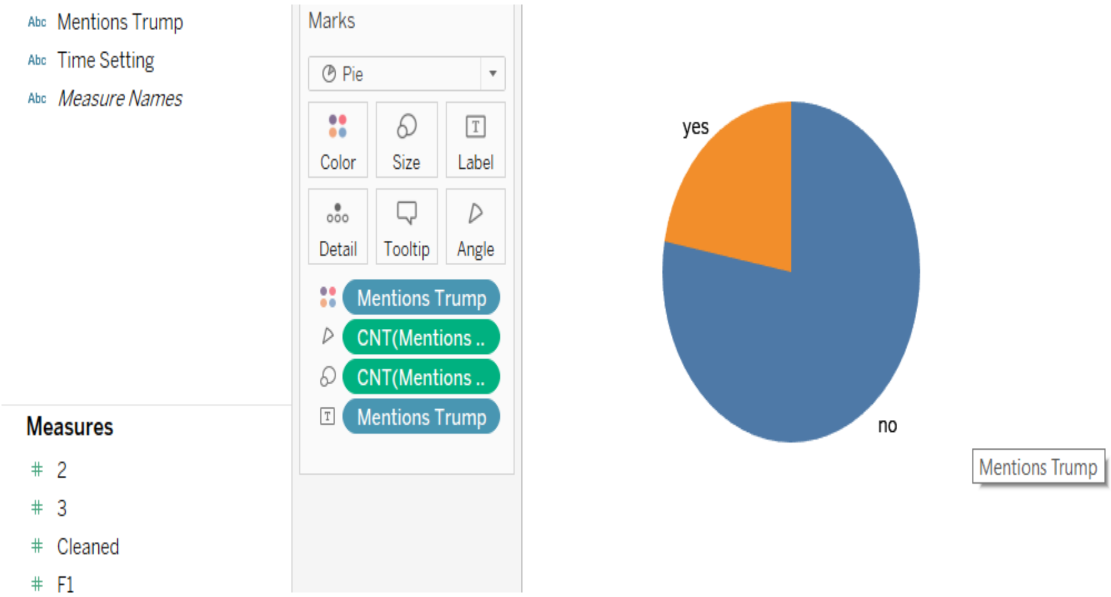
Data | Analytics
data_viz_ass1
Dimensions: 0, 1, Mentions Trump, Time Setting, Measure Names
Measures: Cleaned, F1
Columns: AVG(Likes)
Rows: Mentions Trump
Marks: Automatic

This graph shows that tweets that do mention Trump receives approximately 15% more tweets than the ones who do not mention Trump.



Similar trend is observed for the retweets as tweets with the word Trump on an average receive 12% more retweets.

4. Is CNN obsessed with Trump?



This pie chart shows us that almost 25% of the tweets by CNN use the word Trump in one way or another.

Thus, it is fair to say CNN is a bit obsessed with Trump.

Statistical analysis

1. Are the number of likes and retweets related?

   After running the code on R, it looks like these two columns are highly correlated wit h a value of 0.96.

   We also saw the same result in earlier analysis as both these columns follow the simi lar trend.

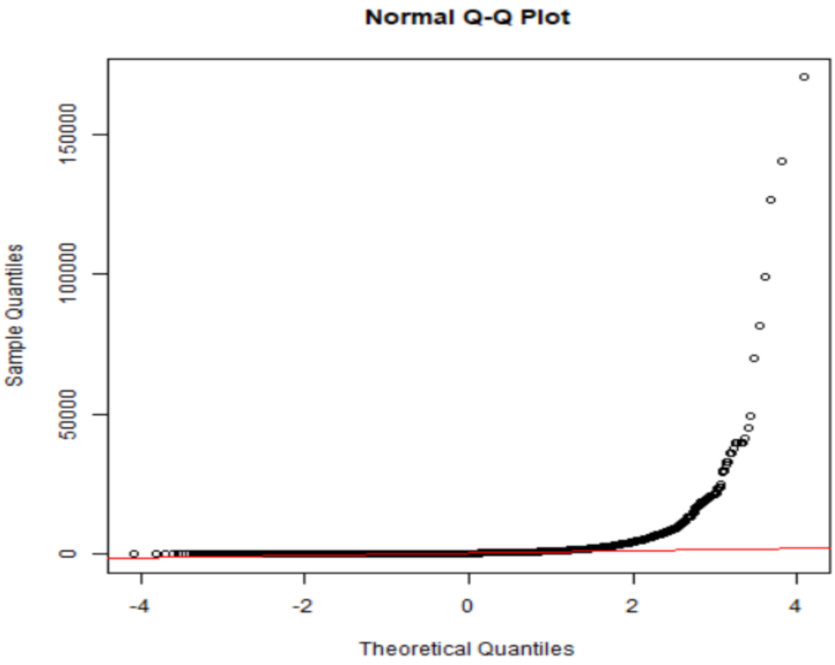
   This makes sense as well as we see normally see tweets with more likes has more co mments/retweets.

   Thus, it is possible to use a linear regression model to predict the number of likes or retweets using the other columns.

2. The mean value of the column LIKES is 778 and the median value is 331.

   This tell us that the values aren't normally distributed.

   We can also see the same with the Q-Q plot below(using R).



The similar trend is observed for retweets.

Conclusion and Reflection

Challenges - This project has helped in understanding the various issues/challenges faced by the data scientist such as –

1. The collection of data has become significantly difficult after the Cambridge Analytica – Facebook scandal as the Twitter API has restricted the max download tweets to 3200.
   Hence, I had to develop a new python script to circumvent this restriction.

2. The time involved in collecting and cleaning the data is way more than doing the analysis or implementing the machine learning algorithm. It took over 20 hours for me to collect and clean the data and a few hours for the exploration and the ML part.

Learning from the data – This project has allowed me to learn a few things as well as answer all the questions I had at the beginning of the project –

1. One shocking revelation to me is that CNN posts most of the tweets in night and late night whereas they get the best reactions in the afternoon time period. Based on the data they should post more tweets in the afternoon to engage their audience.

2. The timing of a tweet can affect the reaction/involvement of the people and that people are more active on twitter in the afternoon. This was a strange result for me as most of the people are at work during this time which seems counterintuitive.

3. People enjoy reading topics which are polarizing on Twitter such as Donald Trump and Politics.

4. It seems like CNN is completely fascinated with Trump as one out of 4 tweets posted by them involves trump in some capacity.

Learning from the project – I learnt the following things from the project which I didn’t know -

1. Usage of Selenium, Chromedriver and twitter’s API to autonomously capture tweets.

2. Application of latent latent dirichlet allocation(LDA) algorithm in NLP clustering.

3. Usage of regular expression in cleaning the data.

Suggested Improvements – Some of the techniques if implemented can reduce time as well as improve the analysis –

1. Image classification to classify clickbait images as they generally have higher likes and retweets.

2. Implementing parallel processors to decrease the time involved in collecting and cleaning the data.

References

1. https://github.com/bpb27/twitter_scraping – Used for scrapping twitter Id's

2. https://developer.twitter.com – generating auth Id and key for API

3. https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21 - implementing LDA in python for NLP clustering.

4. https://stackoverflow.com – for general codes.

Tools used –

1. Python 3 – for data collection, wrangling and applying LDA.

2. Tableau – For explanator analysis.

3. R – for statistical analysis.