

SID - 16103104

NAME - Lovedeep Singh

Big Data Analytics

Climate Sentiment Analysis on twitter tweets

Note: Free api of twitter has a limit on number of tweets that can be scrapped for a particular search query. Hence, this assignment contains the maximum number of tweets that can be scrapped at a time within a past span(at max, limit of free api)

Here's the code and corresponding visuals

Import the required libraries and packages

```
In [50]: 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
from textblob import TextBlob

import warnings
warnings.filterwarnings("ignore")

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

Set the keys for the api

```
In [51]: consumer_key = 'cG8iirfbDnQolqLU7SgCNjDXU'
consumer_secret = 'zXKHM2oSD5xMykQT1TtiJUC8lfEgLn17pN54X8vnFnnr4KUmV4'
access_token = '2858315816-aV2Q4LmxVY144TMkemk1PMeau2eemWAZRkDGyRe'
access_token_secret = 'jPbGh8Qe8trhLKOGseczZ0DRib12NtHcI8ZpP7HxB8pGg'
```

```
In [52]: 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)
```

Lets start the coding

```
In [53]: 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())
```

```
In [ ]: # Create a custom search term and define the number of tweets
search_term = "#climate+change -filter:retweets"

tweets = tw.Cursor(api.search,
                    q=search_term,
                    lang="en").items()

# Remove URLs
tweets_no_urls = [remove_url(tweet.text) for tweet in tweets]
```

```
In [60]: # Create textblob objects of the tweets
sentiment_objects = [TextBlob(tweet) for tweet in tweets_no_urls]

sentiment_objects[0].polarity, sentiment_objects[0]
##(-0.2, TextBlob("InsuranceBureau Hey Yoohoo Hey InsuranceBureau M
aybe sometime before today and everyday from now on you sh"))
```

```
Out[60]: (0.016666666666666666,
TextBlob("The emergence of lowcarbon gentrification as a process
to change the social and spatial composition of urban dis"))
```

```
In [61]: # Create list of polarity valuesx and tweet text
sentiment_values = [[tweet.sentiment.polarity, str(tweet)] for twee
t in sentiment_objects]

sentiment_values[0]
```

```
Out[61]: [0.016666666666666666,
'The emergence of lowcarbon gentrification as a process to change
the social and spatial composition of urban dis']
```

```
In [62]: # Create dataframe containing the polarity value and tweet text
sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity",
"tweet"])

sentiment_df.head()
```

```
Out[62]:
```

	polarity	tweet
0	0.016667	The emergence of lowcarbon gentrification as a...
1	0.000000	Climate Change and TerrorismClimateChange terr...
2	-0.200000	MIT study finds Climate Change impacts everyda...
3	0.325000	Global Hunger Index 2019 The Challenge of Hung...
4	0.000000	Join us tomorrow with Liza Troshka Senior Sust...

Here, we are taking

neutral ==> polarity == 0

negative ==> polarity < 0

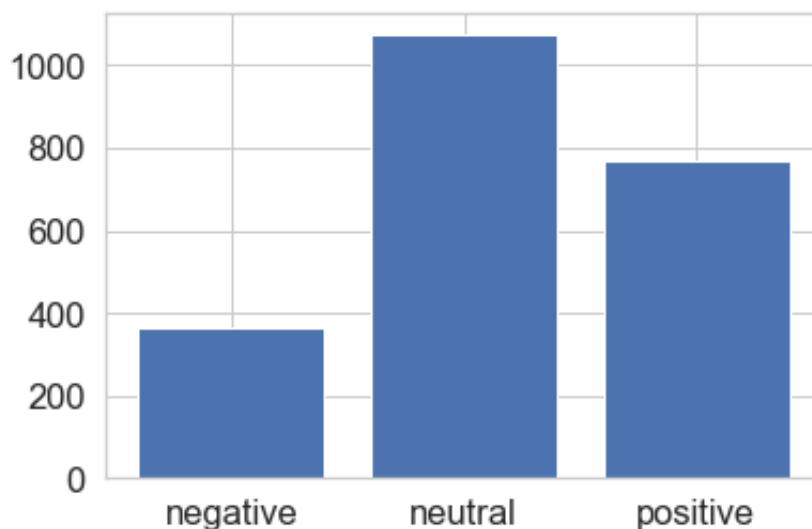
positive ==> polarity > 0

```
In [63]: neutral = sentiment_df.query('polarity == 0.0').polarity.count()  
negative = sentiment_df.query('polarity < 0.0').polarity.count()  
positive = sentiment_df.query('polarity > 0.0').polarity.count()
```

Lets plot the graph for the same

```
In [64]: plt.bar(['negative', 'neutral', 'positive'], [negative, neutral, po  
sitive])
```

Out[64]: <BarContainer object of 3 artists>

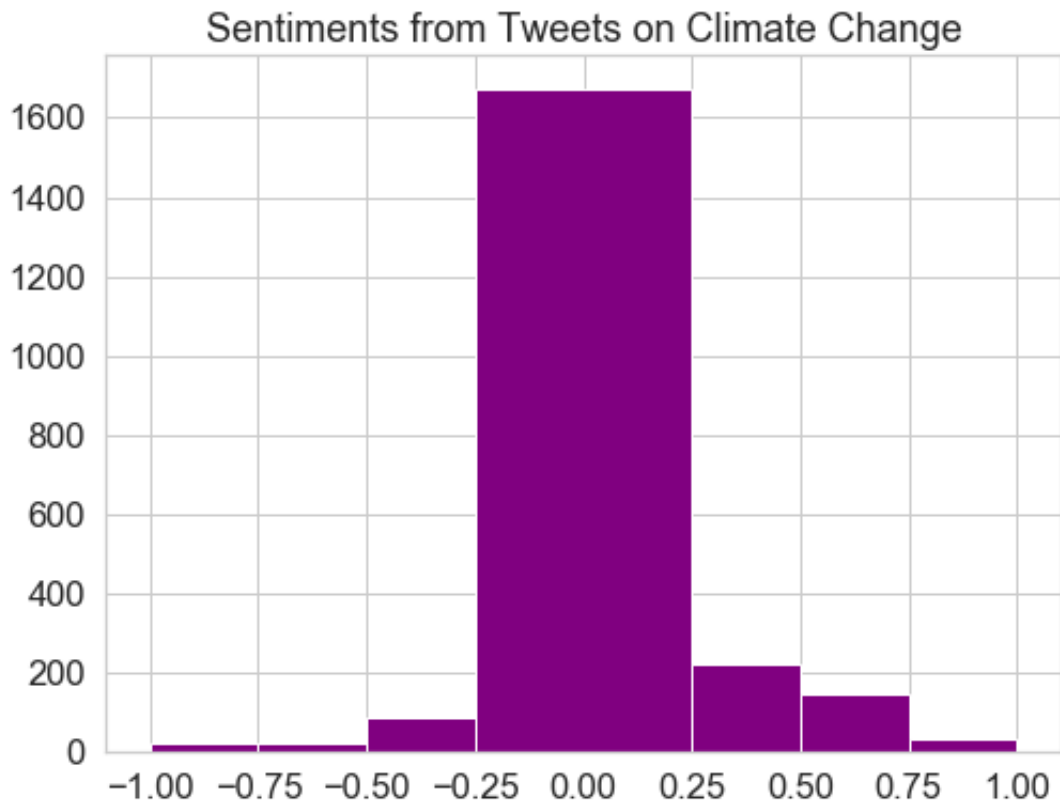


We can also plot the graph by taking intervals in quartiles

```
In [65]: fig, ax = plt.subplots(figsize=(8, 6))

# Plot histogram of the polarity values
sentiment_df.hist(bins=[-1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1]
,
                  ax=ax,
                  color="purple")

plt.title("Sentiments from Tweets on Climate Change")
plt.show()
```

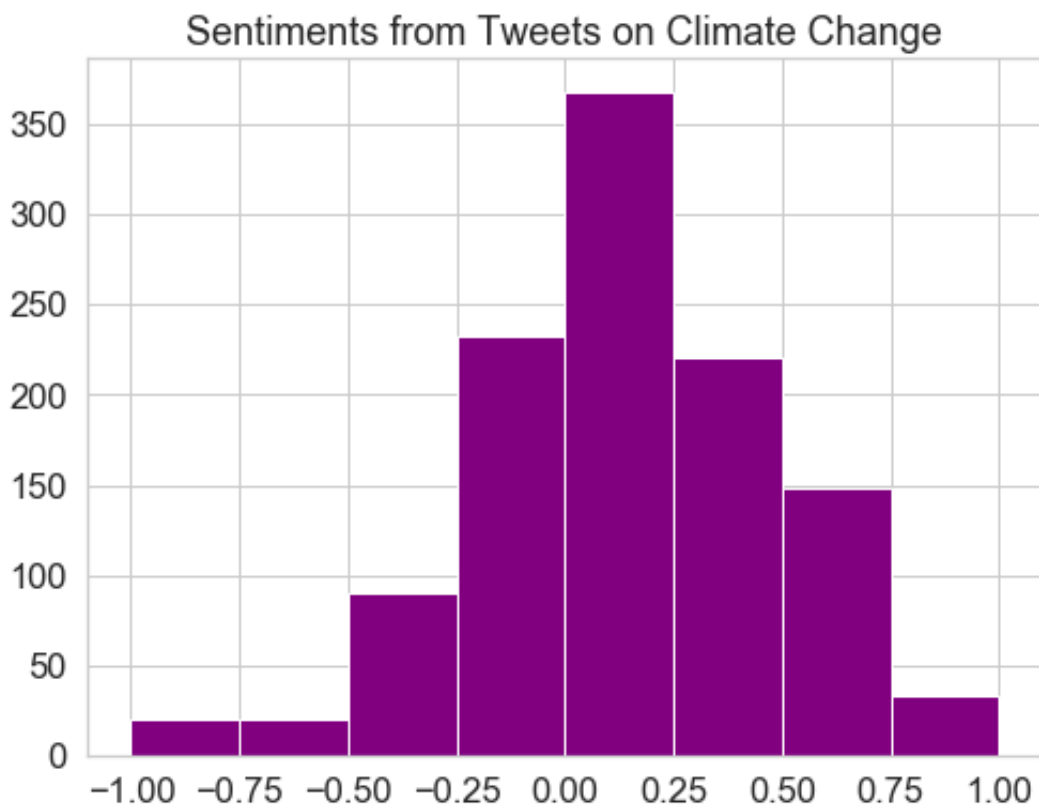


Splitting the middle interval, removing the polarity value of 0

```
In [66]: # 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 Climate Change")
plt.show()
```



In []: