UBC SAUDER SCHOOL OF BUSINESS

BAIT508 Group Project: Social Media Analytics

Section Number:

BA2 Group 23

Team Members Information:

Full Name	Student ID	Email	Role in project
Junda Liu	88015821	swufeljd@gmail.com	1. Drafted Jupyter Notebook and the report; 2. Coded for Part A and B
Zhihao Hu	76403963	hzh835567098@163.com	1.Collected json file data for author information; 2.Coded for part C and D

UBC SAUDER SCHOOL OF BUSINESS

CATALOGUE

UBC SAUDER SCHOOL OF BUSINESS

Definition of words

The table below displays the name and meaning for the words used in our codes:

Name	Data type	Explanation
streaming_results	dictionary	A dictionary to contain the information of 10K collected tweets
author_information	list	A list to contain all the author information
words1	list	A list to contain all the texts words in 10K collected tweets
words2	list	A list to contain all the texts words excluded stop words in 10K collected tweets

Part A. Keyword Selection and Data Collection

This part serves as a foundation for the preliminary analysis, data visualization, textual analysis and analytical driven insights sections in the following parts. In this part, we collected 10K tweets with the keywords "ElonMusk", stored the data in json file, and collected author IDs from the tweets. Based on the author IDs we collected, we further collected author information.

1. Pick one keyword

The keyword we pick is ElonMusk.

2. Collect 10K recent tweets on the selected keyword.

We collected 10K tweets during the past week under the keyword "ElonMusk" (from 2022-09-28 05:19:00 to 2022-10-05 15:19:00).

Then we stored our results in a json file named "elon.json" in zip file .

3. From the collected tweets, get the list of unique author IDs.

In order to provide foundation for textual analysis and preliminary analysis, we collected a list of unique author IDs from the tweets, which could be used to specifically identify the information of the users.

- Firstly we use an empty list to contain the IDs and use for loop and `append` method to collect IDs from `streaming_result`.
- Then we used set function to get the unique . The codes are as follows:

```
authors_id=[]
for i in range(len(streaming_result["tweets"])):
    authors_id.append(streaming_result["tweets"][i]["author_id"])
uniq_authors_id = []
for i in authors_id:
    if i not in uniq_authors_id:
        uniq_authors_id.append(i)
pprint(authors_id)
pprint(len(uniq_authors_id))
```

4. Collect the author information of those author IDs.

Using the unique author IDs as the keys, we collected author information for each ID, and stored it in a json file named "author_info.json". The codes are as follows:

```
import json
aut_info = []
        for ids in authors_id:
            try:
                aut_info.append(tc.fetch_author_info(ids))
                b = json.dumps(aut_info,indent=2)
                with open("author_info.json","a") as f:
                    f.write(b)
            except tweepy.TooManyRequests:
                print("TooManyRequests")
                time.sleep(15*60)
### Cited from:Anonymous Calc and Yi-Hsuan Chen, (2022,October 2),Deal with 429 Too Many Requests,Piazza
```

Here is an example of the author information:

{"id": "1872465721", "created_at": "2013-09-16T17:32:22.000Z", "public_metrics": { "followers_count": 131, "following_count": 80, "tweet_count": 50566, "listed_count": 7}, "verified": false, "description": "I like to have informative discussions with informed people.", "name": "Sajida Wasim", "username": "WasiMOra }

It consists of user name, true name of the person, description and public metrics. The public metrics contain information about the publicity of the authors, and they are used as input for preliminary analysis to evaluate the influence of authors.

Part B. Preliminary Analysis

1. What are the ten most popular words *with* and *without* stop words?

(1) For ten most popular words with stop words

We took the following steps to obtain the 10 most popular words with stop words:

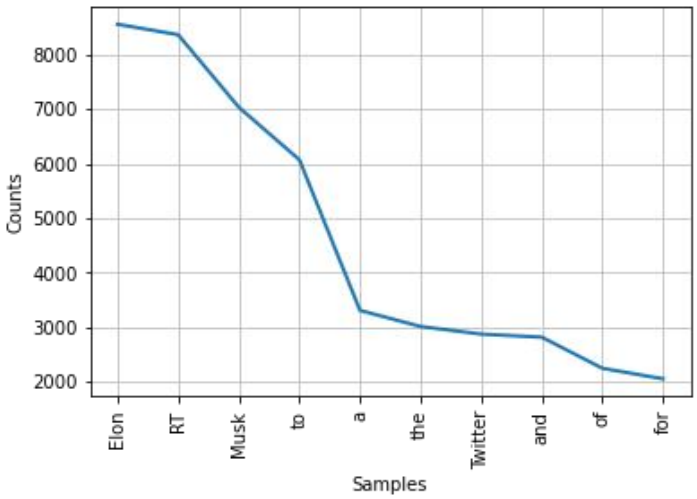
- Use a for loop to traverse the whole `streaming_results` dictionary to get the list of the text information of all tweets and store them in a list named "`words1`". The codes are as follows:

```
%matplotlib inline
# most common words
text=[]
for i in range(len(streaming_result["tweets"])):
    text.append(streaming_result["tweets"][i]["text"])
words=""
for j in range(len(text)):
    words=words+" "+text[j]
words1= words.split()
```

- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package. The codes are as follows:

```
# get frequent words
freq = nltk.FreqDist(words1)
freq.plot(10)
pprint(text)
Counter(words1).most_common(10)
```

Here is a plot of the frequency of words with stop words:



The 10 most popular words with stop words are:

Words	Frequency(times)
Elon	8,563
RT	8,371
Musk	7,039
to	6,073
a	3,304
the	3,008
Twitter	2,869
and	2,812
of	2,240
for	2,047

Analysis: We can see that the popular words with stop words are Elon (8,563), RT(8,371), Musk(7,039), to(6,073), a(3,304), the(3,008), Twitter(2,869), and(2,812), of(2,204) and for(2,047). As there are so many stop words involved, we cannot gain too many insights from the popular words, therefore we should exclude the stops words to facilitate our analysis.

(2) For ten most popular words without stop words

We took the further steps to obtain the 10 most popular words without stop words:

- Create a stop words list to contain all the stop words we want to exclude. As the keyword itself will appear a lot in the tweets collected under it, we also add "Elon" and "Musk" into the stop words list in this step, and this allows us to gain more popular words beyond the keyword itself.

```
import pickle
with open('data/stopwords.pkl', 'rb') as f:
    stopwords = pickle.load(f)
```

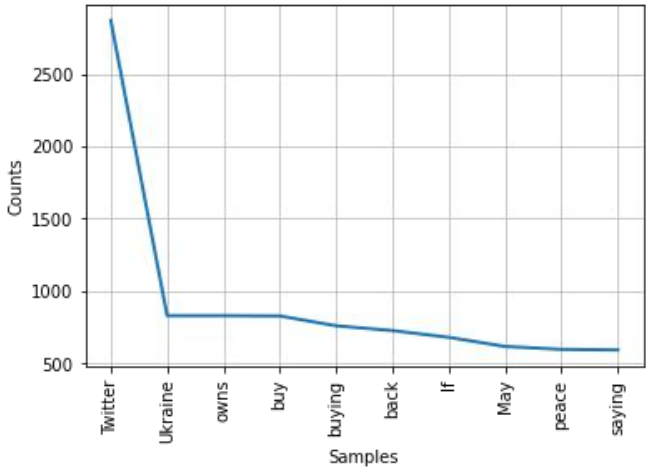
- Use for loop to traverse all the elements in the list containing all the words from texts and exclude stop words, save the remaining words in a list named "words2".

```
import nltk

text1 = [] # our accumulator list

for w in words1:
    if w not in stopwords and len(w) > 1 and w not in ['Elon','Musk','de','RT','u,']:
        text1.append(w)
print(text1)
```

- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package.

```
# get frequent words
a = Counter(text1).most_common(10)
print(a)
freq1 = nltk.FreqDist(text1)
freq1.plot(10);
```

Here is a plot of the frequency of words without stop words:



The 10 most popular words without stop words are:

Words	Frequency(times)
Twitter	2,869
Ukraine	828
owns	828
buy	826
buying	758
back	725
if	678
May	615
peace	595
saying	591

Analysis: From the table, we can deduce that there are two topics under discussion for the keyword "ElonMusk", **the acquisition of Twitter and Ukraine war**. The words "Twitter", "buying", "buy" and "owns" refer to acquisition of Twitter, and the words "Ukraine" and "peace" refer to Ukraine war.

2. What are the ten most popular hashtags (#hashtag)?

We took the following steps to obtain the 10 most popular hashtags:

- Use for loop to select the hashtags in the `words1` list.

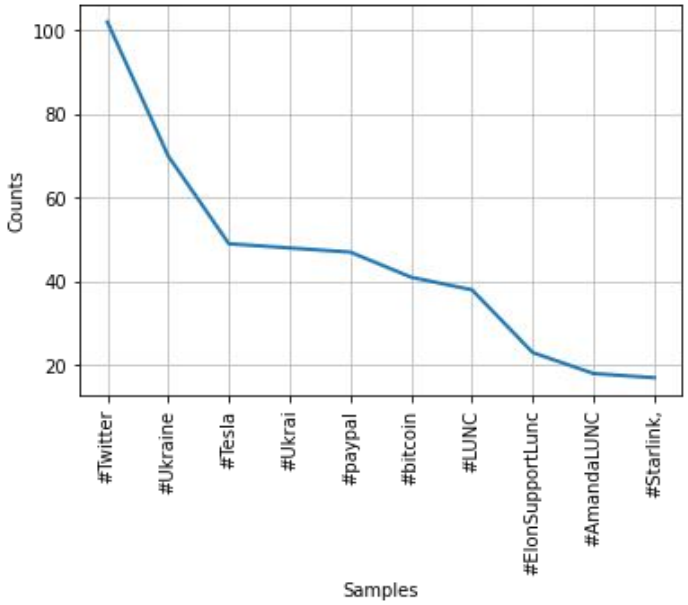
```
hashtag=[]
for i in range(len(words1)):
    if "#" in words1[i] and words1[i] not in ["#Musk","#ElonMusk","#Elon"]:
        hashtag.append(words1[i].strip("…"))# to exclude the strings attached to the hashtags
print(hashtag)
```

- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package.

```
# get frequent wordsn and plot
freq2 = nltk.FreqDist(hashtag)
freq2.plot(10);
Counter(hashtag).most_common(10)
```

- In this question, we also exclude the hashtags related to the keyword such as “Elon”, “Musk” and “ElonMusk” to obtain more hashtags.

Here is a plot of the frequency of hashtags:



The 10 most popular hashtags are:

Hashtags	**Frequency(times)**
Twitter	102
Ukraine	70
Tesla	49
Ukari	48
paypal	47
bitcoin	41
LUNC	38
ElonSupportLunc	23
AmandaLUNC	18
Starlink	17

Analysis: There are **some new topics** discussed under the hashtags such as paypal, bitcoin, and Starlink, these are the areas that Elon Musk takes interests in. It is also noteworthy that LUNC is a topic that is frequently discussed in the hashtag.This is a new area that Elon Musk shows support for.

3. What are the ten most frequently mentioned usernames (@username)?

We took the following steps to obtain the 10 most popular hashtags:

- Use for loop to select the usernames in the `words1` list.
- Use `strip` method to exclude the "@" and ":" strings in the beginning and the end of the usernames.

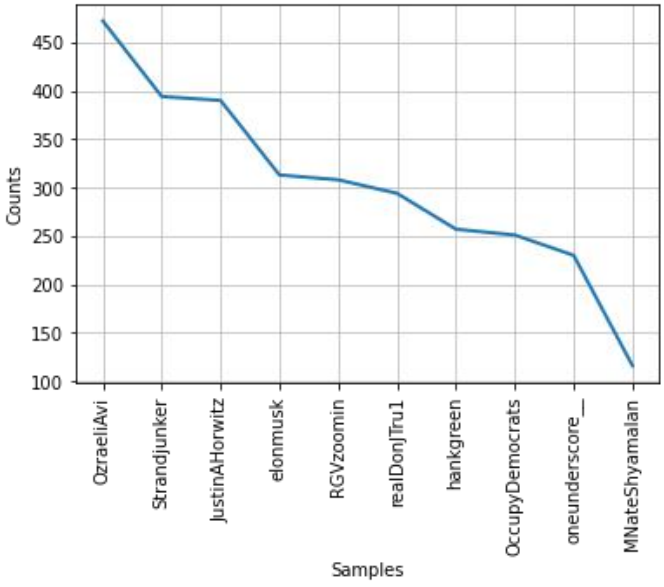
```
usernames=[]
for i in range(len(words1)):
    if "@" in words1[i] and words1[i] != "@" and words1[i] not in ["@elonmusk:"]:
        usernames.append(words1[i].strip("@").strip(":"))
print(usernames)
```

- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package.

```
# get frequent words
print(Counter(usernames).most_common(11))
freq3 = nltk.FreqDist(usernames)
freq3.plot(10);
```

- In this question, we also exclude username "@elonmusk" to obtain more usernames.

Here is a plot of the frequency of username:



The 10 most popular hashtags are:

Hashtags	Frequency(times)
OzraeliAvi	472
JustinAHorwitz	394
Strandjunker	390
RGVzoomin	308
realDonJTru1	294
hankgreen	257

OccupyDemocrats	251
oneunderscore__	230
MNateShyamalan	116
DougJBalloon	111

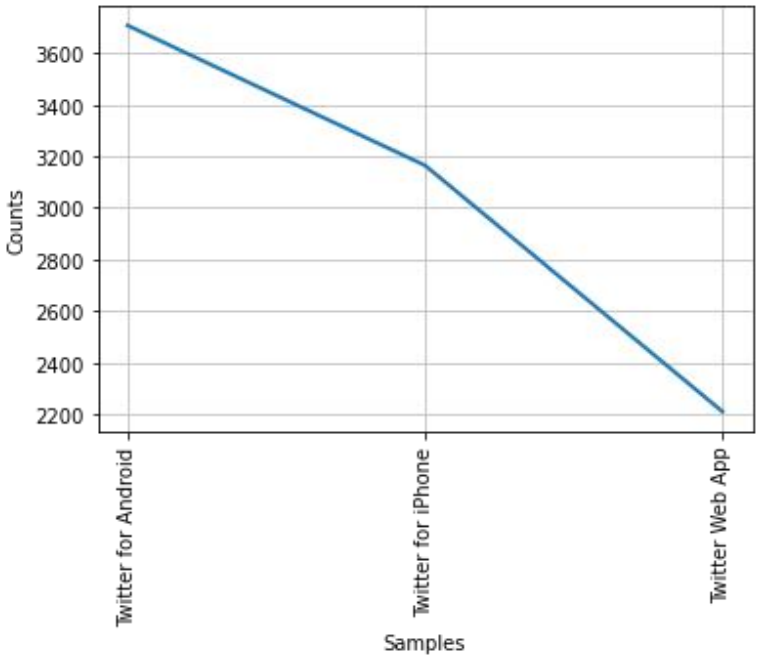
4. Which are the three most common sources of the tweets?

We took the following steps to obtain the 3 most popular sources of tweets:

- Use for loop to select the sources in the `words1` list.
- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package.

```
sources_of_tweets=[]
for i in range(len(streaming_result["tweets"])):
    sources_of_tweets.append(streaming_result["tweets"][i]["source"])
Counter(sources_of_tweets).most_common(3)
# get frequent words
freq4 = nltk.FreqDist(sources_of_tweets)
freq4.plot(3);
```

Here is a plot of the frequency of sources of tweets:



The 3 most common sources of tweets are:

Hashtags	**Frequency(times)**
Twitter for Android	3,707
Twitter for iPhone	3,164
Twitter Web App	2,209

From the table we can see that most tweets (around 70%) comes from mobile terminals such as android and iPhone, there are also around 22% of tweets posted on Web Apps.

5. Create a line chart to show the time trend of tweet counts (number of tweets in a day (or an hour or a minute) depending on the collected data).

In this question, we created a line chart to show the time trend of tweet counts number of tweets **in a minute**, because the range of post time of tweets is within 1 day.

We took the following steps to obtain the line chart:

- Using for loop to select the sources in the `words1` list.
- Define function to convert the format of the time.

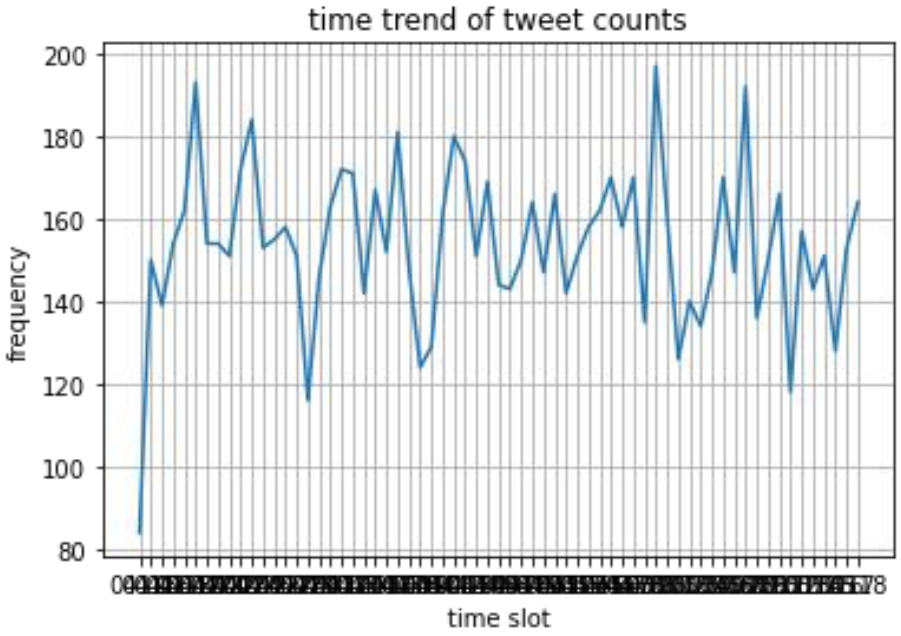
```
def time_handler(target_time):
    _date = datetime.strptime(target_time,"%Y-%m-%dT%H:%M:%S.%fZ")
    end_time = _date.strftime("%Y-%m-%d %H:%M:%S")
    return end_time
```

- Count the frequency of each time slot.

```
# To count the frequancy of each time
freq_dic=dict()
for i in sorted(hm):
    if i not in freq_dic.keys():
        freq_dic[i]=1
    else:
        freq_dic[i]+=1
print(freq_dic)
time_slot=list(freq_dic.keys())
frequency=[]
for k,v in freq_dic.items():
    frequency.append(freq_dic[k])
print(time_slot)
print(frequency)
```

- Get the frequency words using `Counter` function, and plot the results with the functions in `nltk` package.

The line chart is as follows:



Note: As there are so many different time units in 1 day, therefore the scale label of horizontal axis could not be shown clearly.

6. Which are the three most influential tweets?

We took the following steps to obtain the 3 most influential tweets:

- Create lists for “quote_count”, “reply_count”, “retweet_count” and “like_count”.
- Create a dataframe for “quote_count”, “reply_count”, “retweet_count” and “like_count”.

```
dic={"authors_id":authors_id,"quote_count":quote_count,"reply_count":reply_count,"retweet_count":retweet_count,"like_count":like_count}
lst1=["quote_count","reply_count","retweet_count","like_count"]
df1=pd.DataFrame(dic)
print(df1)
```

- Calculate influence score and create a new column named"influence_score" in dataframe.

```
df1["influence_score"]=df1[lst1].sum(axis = 1)
print(df1)
```

- Sort the data frame by influence score and print the results.

```
most_influential_tweets=df1.sort_values("influence_score",ascending=False)[lst1+["influence_score","authors_id"]][:3]
print(most_influential_tweets)
```

The 3 most influential tweets are:

Tweets index	Influence scores	Text content	Author IDs
6180	30643	RT @Strandjunker: Just for those who	554565840

		may be a little confused: Elon "Musk did not generously donate his Starlink satellite internet to "Ukrai…'	
4112	30643	RT @Strandjunker: Just for those who may be a little confused: Elon "Musk did not generously donate his Starlink satellite internet to "Ukrai…'	4776549743
2332	30643	RT @Strandjunker: Just for those who may be a little confused: Elon "Musk did not generously donate his Starlink satellite internet to "Ukrai…'	1339433486

Analysis: It is amazing to find that the 3 most popular tweets are the same, but posted by different users. Actually, the three most popular tweets are all the re-tweets of the original tweet. The content of the original tweet is a satire or sarcasm on Elon Musk by referring to the other topic (Starlink satellite internet) that Elon Musk involved in. The original tweet won favors from many twitter users and was widely re-tweeted, thus gaining high popularity in this topic.

7. Who are the three most vocal authors on the keyword?

We took the following steps to obtain the 3 most vocal authors:

- Use method `json_normalization` to parse the streaming_result into dataframe with the lowest unit in the original dictionary. The code is as follows:

```
tweet = pd.json_normalize(streaming_result['tweets'])
tweet.head()
```

- Use `groupby` and `count` method to count the frequency of each author id in the tweets collected, and create a column named "count" to store the results.

```
tweet1 = tweet.groupby('author_id')['author_id'].count().reset_index(name='counts')
tweet1.head()
```

- Sorted the dataframe by the value of column "count" and use slicing method to select the top 3 vocal authors

```
tweet2 = tweet1.sort_values('counts',ascending = False)[:3]
tweet3 = tweet2['author_id'].tolist()
print(tweet3)
```

The most vocal authors are:

Authors IDs	Usernames	Tweets Count
34135329	EvaCaroMadrid	17
1069209446142877696	jcdaley	16
1213646996940017666	DemeKhadim1	16

8. Who are the three most influential authors?

We took the following steps to obtain the 3 most influential authors:

- Create lists for "followers_count", "following_count", "listed_count", "tweet_count" by using the for loop in author_information.
- Create a dataframe for "followers_count", "following_count", "listed_count". "tweet_count".

```
dic1={"username":username,
      "followers_count":followers_count,
      "following_count":following_count,
      "listed_count":listed_count,
      "tweet_count":tweet_count}
lst2=["followers_count", "following_count", "listed_count", "tweet_count"]
df2=pd.DataFrame(dic1)
print(df2)
```

- Calculate influence score and create a new column named"author_influence_score" in dataframe.

```
df2["author_influence_score"]=df2[lst2].sum(axis = 1)
print(df2)
```

- Sort the dataframe by influence score and print 3 most influential authors by slicing.

```
most_influential_authors=df2.sort_values("author_influence_score",ascending=False)[["username","author_influence_score"]][:3]
print(most_influential_authors)
```

The most influential authors are:

Usernames	Influence Scores
test5f1798	50725451
Reuters	26619337
WSJ	20743501

Part C. Word Cloud

We followed the steps below to create a word cloud from the collected tweets:

- Create a word list without stop words.

```
text2 = [] # our accumulator list
for i in words1:
    if i not in stopwords and len(i) > 1 and i not in ['de','RT','u,']:# to exclude stop words
        text2.append(i)
# exclude the elements with meaningless url elements
text3= []
for i in text2:
    if "https://t.co/" not in i:
        text3.append(i)
```

- Combine the words without stop words into a single long text.

```
text4=""
for i in text3:
    text4=text4+" "+i
```

- Feed the text into the `wordcloud` module and display the generated image

The worlds cloud is as follows:

Analysis:

- This cloud shows the most frequent words and phrases in all the texts. We can find that besides “Elon Musk”, the phrases such as “promoting peace”, “buy Twitter” , “need know” and “everything need” are frequently mentioned in the tweets, which reflects users’ attention and concerns on acquisition and war.

- We noticed that other person’s name such as “Mark Zuckerberg” and “Jeff Bezos” are also appears in the text frequently, this tell us that the recent discussions on Elon Musk **may have spillover effect** on other CEOs of the famous internet companies in North America, namely users **may have the habit or mindset to group people with similar characteristics together in their mind**, **and if a person has a lot of news on internet, the users will automatically think of other persons with similar characteristics, even though they have nothing to do with the news**.

Part D. Sentiment Analysis

1. What are the average polarity and subjectivity scores?

We followed the steps below to calculate the average polarity and subjectivity scores:

- Use for loop and TextBlob to construct the polarity and subjectivity scores list to contain the scores for all the sentences.

- Calculate the average score for each list. The codes are as follows:

```
#construct the polarity and subjectivity scores list
pol_list=[]
sub_list=[]
for i in text:
    pol_list.append(TextBlob(i).sentiment.polarity)
    sub_list.append(TextBlob(i).sentiment.subjectivity)
ave_pol_score=sum(pol_list)/len(pol_list)
ave_sub_score=sum(sub_list)/len(sub_list)
print("The average polarity score is " + str(ave_pol_score))
print("The average subjectivity score is " + str(ave_sub_score))
```

Then we get:

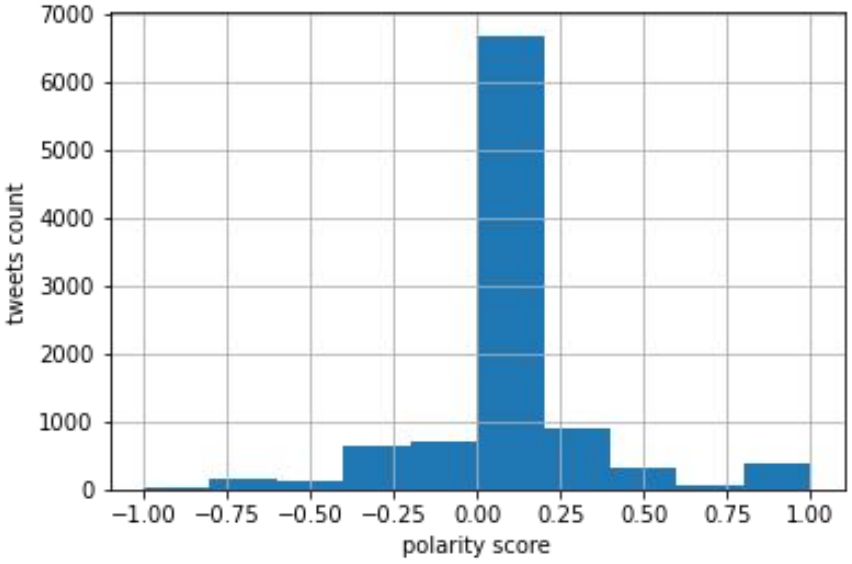
The average polarity score is 0.04 (rounded to 2 decimals).

The average subjectivity score is 0.25 (rounded to 2 decimals).

2. Visualize the polarity and subjectivity score distributions

(1) For the polarity score:

The distribution for polarity score is as follows:



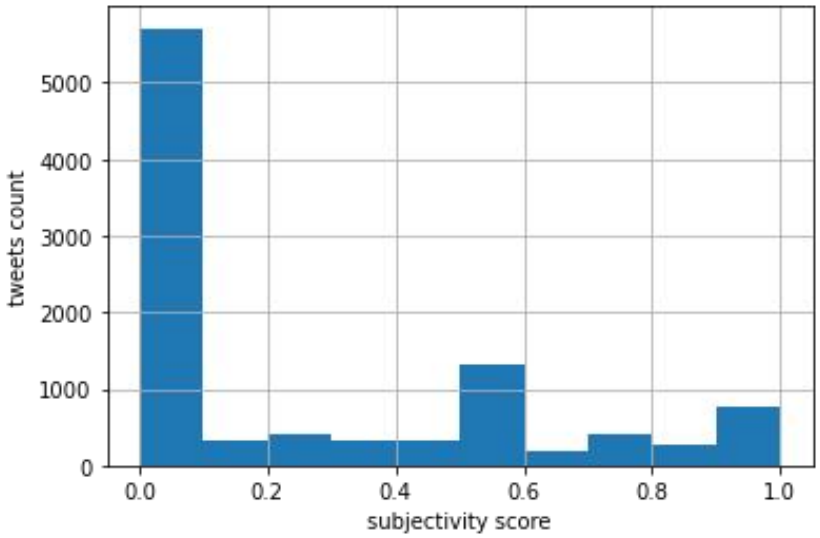
Analysis: the average polarity score is around 0, which means the overall sentiment of the keyword is neutral.And the polarity scores concentrate in range 0 to 0.25, this means that most tweets are mild positive under the keywords.

The codes are as follows:

```
plt.hist(pol_list, bins=10) #, normed=1, alpha=0.75)
plt.xlabel('polarity score')
plt.ylabel('tweets count')
plt.grid(True)
plt.savefig('polarity.pdf')
plt.show()
```

(2) For the subjectivity score:

The distribution for subjectivity score is as follows:



Analysis: the average subjectivity score is around 0.25, which means the overall sentiment of the keyword is mild subjective. And the subjectivity scores concentrate in range 0 to 0.1, this means that most tweets are objective.

The codes are as follows:

```
plt.hist(sub_list, bins=10) #, normed=1, alpha=0.75)
plt.xlabel('subjectivity score')
plt.ylabel('tweets count')
plt.grid(True)
plt.savefig('subjectivity.pdf')
plt.show()
```

3. What are the most positive and negative tweets on the keyword?

We counted the number of maximum and minimum values in the polarity list to detect whether there are multiple tweets with the same sentiment scores, and the results told us that there are 23 tweets with sentiment scores value at 1.0 and 9 tweets with sentiment scores value at -1.0. The codes are as follows:

```
print(max(pol_list))
print(min(pol_list))
print(pol_list.count(max(pol_list)))
print(pol_list.count(min(pol_list)))
print([i for i,j in enumerate(pol_list) if j==1.0])
print([i for i,j in enumerate(pol_list) if j==-1.0])
```

The indexes for the most positive tweets are: [595, 1000, 1188, 1612, 2308, 2323, 2746, 2977, 3056, 3867, 4378, 5024, 5639, 5962, 6267, 6288, 7325, 7746, 7999, 8029, 8463, 8511, 8558]; The indexes for the most negative tweets are: [159, 1522, 4377, 4869, 7148, 7461, 7650, 8174, 9087].

We pick 3 tweets among each of them, the most positive tweets are:

IDs	Text	Polarity score	Index
185115193	Ram Gopal Varma Extends Sexist Dussehra 2022 Greetings to Followers, That They Get Elon Musk's Wealth and Beautiful Women with Wife's Permission!	1.0	595
4075396512	RT @MariaSchenetzke: @oneunderscore__ @jaredlholt Make Putin loaned him some money Putin was very happy about Elon Musk's tweet yesterday	1.0	5639
1512746965272047622	RT @chicago_glenn: Elon Musk is the best thing that's ever happened to Twitter!	1.0	8558

The possible reasons are:

- For author ID185115193: even though the sentiment score of the tweets is high, the content signifies a protest for Ram Gopal Varma's sexist on women followers.
- For author ID4075396512: the author expressed his opinion about Musk's recent tweets towards Ukraine War, he believes there may be internal transactions between Putin's government and Musk's companies, he views Musk's speech as a purported way to gain benefit for his company.
- For author ID1512746965272047622: the author expressed his firm support for Musk's acquisition plan for Twitter.He believe Musk will rebuilt this platform and generate lots of value for the users.

The most negative tweets are:

IDs	Text	Polarity score	Index
1479295916830081028	RT @supermarETH: ELON MUSK WITH THIS INSANE UPDATE Hit that like and retweet button to see	-1.0	159
536008303	Need Elon Musk to kill this Twitter swipe up video update, it's terrible	-1.0	8174
29993717	@KavalAuthorActs Purging accounts before	-1.0	9087

	the buyout from Boring company (Elon Musk)		

The possible reasons are:

- For author ID1479295916830081028: the author is apparently feeling terrible for Elon Musk's proposal to end the Ukraine War with Russia.
- For author ID536008303: even though the sentiment score of this tweet is negative, the author actually expressed his compliant about Twitter's swipe up video update function, and showed his expectation for Musk to mitigate the video update.
- For author ID29993717: the author expressed his negative opinion on Musk's acquisition plan for Twitter by mentioning a author he likes named Pamela Kaval.

Part E. Insights

1. Please describe the insights you gained from the analyses.

The preliminary analysis tells us that:

Recently, Elon Musk has been involved in several controversial or popular topics: Twitter Acquisition, Ukraine War, LUNC and Bitcoin. He offered to buy Twitter for original price, weeks before trial following his moves in July to terminate his planned $44 billion purchase; He has gotten into a Twitter tussle with Ukrainian President Volodymyr Zelenskyy after the tech billionaire floated a divisive proposal to end Russia's invasion. Also, as one of the most influential opinion leaders of Bitcoin, he is often tied with the discussion of Bitcoin, especially debate about the LUNC. People are asking his opinion since every comment he made on Bitcoin will probably significantly affect the price of one certain Bitcoin.

The sentiment analysis tells us that:

The average polarity score is 0, and the distribution of the polarity score is concentrated near 0, which means the twitter users on average hold neutral attitudes towards Elon Musk's recent speeches.

The average subjectivity score is 0.25, which means that people have mild subjectivity on the controversial topics.

The words cloud tells us that:

The discussion on Elon Musk **may have spillover effect** on other CEOs of the famous internet companies in North America, namely users may have the **habit or mindset** to group people with similar characteristics together in their mind, and if a person has a lot of news on internet, the users will automatically think of other persons with similar characteristics, even though they have nothing to do with the news directly. This is actually a common phenomenon on social media platform.

2.Think about a broader social media project you may conduct using more datasets such as intermetal corporate datasets (e.g., customer transactions, firm financials, HR data) as well as other unstructured data (e.g., business/legal documents, social media pictures, YouTube/TikTok videos, etc.). I look forward to seeing your unique and creative perspectives.

The details of the project are as follows:

Topic: How will stakeholder's sentiment affect the performance of companies.

Datasets: internal corporate datasets (financials, customer transaction, HR data); structured and unstructured data on social media platform about the companies.

Content:

Part A: Data Collection

Using python to collect structured and unstructured data concerning about stakeholder's sentiment towards companies from social media platform.

Part B: Data Analysis

Parse the structured and unstructured data to quantify the sentiment of stakeholders on companies' operation.

Part C: Hypothesis testing

- Propose meaningful hypothesis to be tested, for example:
  - Positive sentiment from stakeholders will improve the financial records of company.
  - Positive sentiment from stakeholder will increase customer transaction.
  - Positive sentiment from stakeholder will increase employee retention ration.
- Build econometrical model for the hypothesis
- Use python to run the regression models to test hypothesis.

Insight and Conclusions