**A. [Data Collection; 20 points] Pick an English keyword that interests your team.**

We choose the keyword ‘Cannabis’ because the incoming marijuana legalization in July 2018 in Canada. We believe this topic/keyword is worth researching on twitter to see how people feel about cannabis/marijuana. We run the following code throughout a week, in average 13 times per day, to get freshly streaming tweets daily instead of getting all 10k tweets on the same day because we believe this is a better way to collect different ideas and have us more exposure to this topic.

The main source of this streaming code is from Professor Gene Lee, with our slightly modification:

***def store\_json(self):***

***with open('tweet\_stream\_{}\_{}\_{}.json'.format(keyword, len(tweets),times), 'w') as f:***

***json.dump(tweets, f, indent=4)***

***self.disconnect()***

As we want to the program to stream and catch each 100 tweets into one json file, and then go one and start on a new json file doing the same for the next 100 tweets, we modified the code so that the json file name will reflect the keyword, the amount of tweets in each file and the sequence of the file itself.

And then we create a for loop to control what keyword we use, how many times we run the streaming and from which sequence we start the streaming:

***keyword = 'Cannabis'***

***for times in range(0,100):***

***stream.statuses.filter(track=keyword)***

***tweets = []***

As we use the append method in the tweets list, every time we finish storing 100 tweets into 1 json file, we will want the tweets list to be cleared and ready for the next run, instead of creating another new list to store the next 100 tweets.

After running approximately 7 days (from November 22 to 29), we have our 10K tweets for the keyword “Cannabis”.

**B. [Preliminary Analysis; 20 points] Using the collect tweets, please answer the following questions:**

With our observation of json file, each json file contains a list of dictionaries. Each piece of tweet streamed from twitter API were stored in a dictionary. Within the dictionary, there are keys and values. Some of the values are dictionaries which are in subordinate levels in the contents.

**- Question a: Ten most popular keywords with and without stop words**

To extract the top ten popular keywords appearing in tweets, we need to separate everything about the content of tweets from the whole dictionary of a single tweet streamed from the twitter API. We observed that there exist item ‘text’ in each dictionary of tweets. Sometimes ‘extended\_tweet’ appears in the top level of a certain tweet and ‘full-text’ is a key inside that subordinate dictionary. Therefore, we used if and for loop to open the json file and extract ‘text’ and possibly ‘full text’ from it:

***for runs in range(0,100):***

***with open('tweet\_stream\_Cannabis\_100\_{}.json'.format(runs), 'r') as g:***

***tweets = json.load(g)***

***for item in tweets:***

***if 'extended\_tweet' in item:***

***text = item['extended\_tweet']['full\_text'].lower()***

***else:***

***text = item['text'].lower()***

In each text of tweet, there is a url of which we want to get rid. Due to the fact that every url starts with ‘http’, we can eliminate the web links through the following process:

***for s in t:***

***if 'http' not in s:***

***text += s+' '***

After punctuation and tokenizing, we obtain two lists of words split from all texts in tweets, of which one is with stop words, the other one is not:

***text = text.translate(table\_p)***

***text = nltk.word\_tokenize(text)***

***for word in text:***

***withstop\_wlst.append(word)***

***if word not in stopwords and len(word) > 1:***

***nostop\_wlst.append(word)***

We can count the frequency of words appearing in the two lists and return the results of counting:

***nonstopc = Counter(nostop\_wlst).most\_common(10)***

***withstopc = Counter(withstop\_wlst).most\_common(10)***

With printing the results in certain format, the results are as shown below:

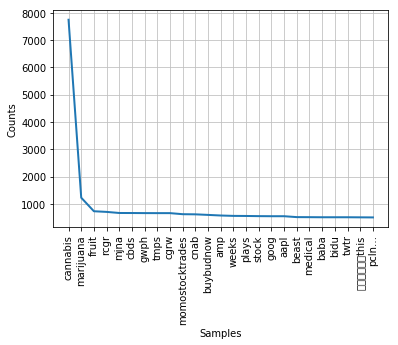
The 10 most popular keywords with stop words are: [('cannabis', 7743), ('rt', 5474), ('the', 3547), ('to', 3130), ('in', 2127), ('and', 2092), ('of', 2081), ('a', 1962), ('for', 1869), ('is', 1390)]

The 10 most popular words with stop words are: cannabis, rt, the, to, in, and, of, a, for, is.

The 10 most popular keywords without stop words are: [('cannabis', 7743), ('marijuana', 1226), ('fruit', 730), ('rcgr', 706), ('mjna', 666), ('cbds', 665), ('gwph', 662), ('tmps', 661), ('cgrw', 661), ('momostocktrades', 622)]

The 10 most popular words without stop words are: cannabis, marijuana, fruit, rcgr, mjna, cbds, gwph, tmps, cgrw, momostocktrades.

We’ve also required a frequency table of words appearing in the text of tweets from python (stopwords eliminated):



**Question b: Ten most popular hashtags (#hashtag)**

In the same ‘for’ loop which aims to read each item in tweets, we conducted similar commands to extract all the hashtags inside tweets in each json file:

***hashtag = item['entities']['hashtags']***

***for hash in hashtag:***

***hash\_list.append(hash['text'].lower())***

Similarly, We can count the frequency of hashtags appearing in the hash\_list and return the results of counting:

***hashclist = Counter(hash\_list).most\_common(10)***

With printing the results in certain format, the results are as shown below:

Question b: The 10 most common Hashtags are: [('cannabis', 2386), ('marijuana', 535), ('stock', 510), ('stocks', 297), ('bcbud', 215), ('weed', 180), ('mustweed', 179), ('cbd', 165), ('buybudnow', 155), ('mmj', 136)]

The 10 most common Hashtags are: #cannabis, #marijuana, #stock, #stocks, #bcbud, #weed, #mustweed, #cbd, #buybudnow, #mmj.

**Question c: Ten most frequently appearing usernames (@username)?**

In the same for loop which aims to read each item in tweets, we conducted similar commands to extract all the usernames mentioned in the content of tweets in each json file:

***mentions = item['entities']['user\_mentions']***

***for ment in mentions:***

***mention\_list.append(ment['screen\_name'])***

Similarly, We can count the frequency of usernames appearing in the mention\_list and return the results of counting:

***mentc = Counter(mention\_list).most\_common(10)***

With printing the results in certain format, the results are as shown below:

The 10 most frequently appearing usernames are: [('Mamene\_Lorenzo', 726), ('MOMOSTOCKTRADES', 622), ('buybudnow', 503), ('Trekles', 154), ('JodieEmery', 120), ('elparaguayo123', 100), ('prayforme79', 83), ('Leafly', 76), ('CannabisBizNews', 72), ('nowthisnews', 53)]

The 10 most frequently appearing usernames are: 'Mamene\_Lorenzo', 'MOMOSTOCKTRADES', 'buybudnow', 'Trekles', 'JodieEmery', 'elparaguayo123', 'prayforme79', 'Leafly', 'CannabisBizNews', 'nowthisnews'.

**Question d: The most frequently tweeting person about the keyword?**

In the same ‘for’ loop which aims to read each item in tweets, we conducted similar commands to extract all the names of tweeting person in each json file:

***username=item['user']['screen\_name']***

***usern\_list.append(username)***

Similarly, We can count the frequency of usernames appearing in the usern\_list and return the results of counting:

***tweetp = Counter(usern\_list).most\_common(1)***

With printing the results in certain format, the results are as shown below:

The most frequently tweeting person about the keyword is: collins\_wilbert

**Question e: The most influential tweet? (Define influence as the sum of retweet count, reply count, and quote count.)**

The influence of tweet is defined as the sum of retweet count, reply count, and quote count. To find the most influential tweet, we decided to define a new function called ‘findinfluence’ which aims to firstly count the summation of retweet count, quote count, and reply count inside the contents of tweets and then return the content of the ‘champion text’ together with the biggest summation number using if statement to iterate values inside the function:

***def findinfluence(RQT):***

***global sumcount***

***global championtext***

***quotec = RQT['quote\_count']***

***retweetc = RQT['retweet\_count']***

***replyc = RQT['reply\_count']***

***sumc = quotec + retweetc + replyc***

***if sumc > sumcount:***

***championtext = RQT['text']***

***sumcount = sumc***

***return championtext***

***return sumcount***

Through observation of the json file, we found that the labels of ‘quote\_count’, ‘reply\_count’, and ‘retweet\_count’ appear under the top level of each tweet in some cases. These counts numbers are always 0 because we are streaming the tweets real time, and theoretically there is no time for anyone to retweet this streaming tweets immediately. Sometimes ‘retweeted\_status’ and ‘quoted\_status’ appeared in the top level of the contents of tweets, which means the user was retweeting or quoting others’ tweets and during which time we need to count the sum of retweet count, reply count, and quote count under those two levels. In the same ‘for loop’ which aims to read each item in tweets, we used two ‘if statements’ to distinguish the three different levels and aggregate all the items needed to conduct the findinfluence function:

***if 'retweeted\_status' in item:***

***i = item['retweeted\_status']***

***findinfluence(i)***

***if 'quoted\_status' in item:***

***j = item['quoted\_status']***

***findinfluence(j)***

***findinfluence(item)***

With printing the results in certain format, the results are as shown below:

The most influential tweet is: "1st step: mind your gotdamn business . https://t.co/qAn3qXII25", with total 81848 in rt, quotes and replies.

The content of the tweet is indeed interesting. We tried to find the source of this tweet in twitter and found out that this quote is a reply to the question: ‘What are the steps to take when you find your neighbor smoking weeds?’

**C. [Word Cloud; 20 points] Create a word cloud from the collected 10K tweets. Depending on the needs, you may want to remove stop words and do stemming before feeding into the word cloud module.**

The focus is still on the content of text of each tweet:

To generate a word cloud, we need to separate everything about the content of tweets from the whole dictionary of a single tweet streamed from the twitter API. The previous steps will be the same as in Question B because we want to get the word list from the 10K tweets, without any hyperlinks and punctuations. And then, we take two further step needs to be taken before generating the wordcloud:

1) Stemming the words in nostop\_wlst

2) Gathering the words inside the final list consist of the 1000 most common words after stemming and then return a string of the aggregation of the words so as to generate a word cloud with it:

***lemma = nltk.wordnet.WordNetLemmatizer()***

***for word in nostop\_wlst:***

***stemlst.append(lemma.lemmatize(word))***

***st = list(Counter(stemlst).most\_common(1000))***

***for word in st:***

***cloudtext1 += ' {}'.format(word[0])***

Referring users’ comments about how to use background image and generate color from it when executing a word cloud: we were able to generate the word cloud by setting some parameters about the word cloud which was about to export. (reference #https://amueller.github.io/word\_cloud/auto\_examples/colored.html ) We went another try with generating the color from background image that we picked:

***backgroud\_Image=plt.imread('cannabis\_leaf.jpg')***

***wc=WordCloud(***

***background\_color='white',***

***mask=backgroud\_Image, max\_words=500, max\_font\_size=40, width=1000, height=1000, random\_state=42)***

***wc.generate\_from\_text(cloudtext1)***

***img\_colors=ImageColorGenerator(backgroud\_Image)***

***wc.recolor(color\_func=img\_colors)***

***plt.imshow(wc)***

***plt.axis('off')***

***plt.show()***

**Cannabis flag (word cloud)**



We used a picture to symbol the cannabis leaf as the background image of the word cloud and then put it inside the Canada national flag to represent the topic we are focusing on. The image of word cloud is shown in the next page.

Inside the word cloud, we could catch some frequently appearing words in tweets related to the key word ‘cannabis’ that we picked. For instance, ‘cannabis’, ‘weed’ and ‘marijuana’ represent the same meaning of our topic; ‘regulation’, ‘legal’, and ‘supporter’ represent the main discussion nowadays: legalization of cannabis; ‘medicinal’, ‘patient’, and ‘cancer’ represent the application of cannabis in pharmaceutical, nutraceutical, and cosmeceutical industries, medical related areas. The content in the word cloud shows that when talking about cannabis, people are concentrating on the process and comments about the legalization of cannabis, the medical use of marijuana, and also the companies engaged in selling tobacco, researching and developing medical application of cannabis.

**D. [Sentiment Analysis; 20 points] Using TextBlob, calculate the polarity and subjectivity scores for each tweet in the 10K tweet corpus. Summarize the calculated scores with histograms using Matplotlib, where X-axis is the score and Y-axis is the tweet count in the score bin. Also, provide the average of the polarity and subjectivity scores.**

In order to increase the efficiency of the code, we insert the following code into the for loop where we used in question B and C to get text from the tweets. Considering the fact that if we use the text after stemming, some important words which reflecting the polarity and subjectivity will be removed. Thus, the following code is placed after we remove the hyperlink from each text.

***tb = ''***

***for ws in text:***

***tb += ws***

***tbb = TextBlob(tb)***

***sub\_list.append(tbb.sentiment.subjectivity)***

***pol\_list.append(tbb.sentiment.polarity)***

By doing so, for each tweet, its subjectivity score and polarity score is stored in the two lists. (sub\_list and pol\_list)

The next step is to visualize the scores of all 10K tweets, with x-axis being the score and y-axis being the tweets count. The two histograms will show the frequency of each score range. To be able to make the histogram more statistically intuitive, we set the alpha level to be 0.95, with 20 bins in the histogram. The code shown below demonstrates how we set up the histogram using Matplotlib. Before doing so, matplotlib packages should be imported into the python environment.

***import matplotlib.mlab as mlab***

***import matplotlib.pyplot as plt***

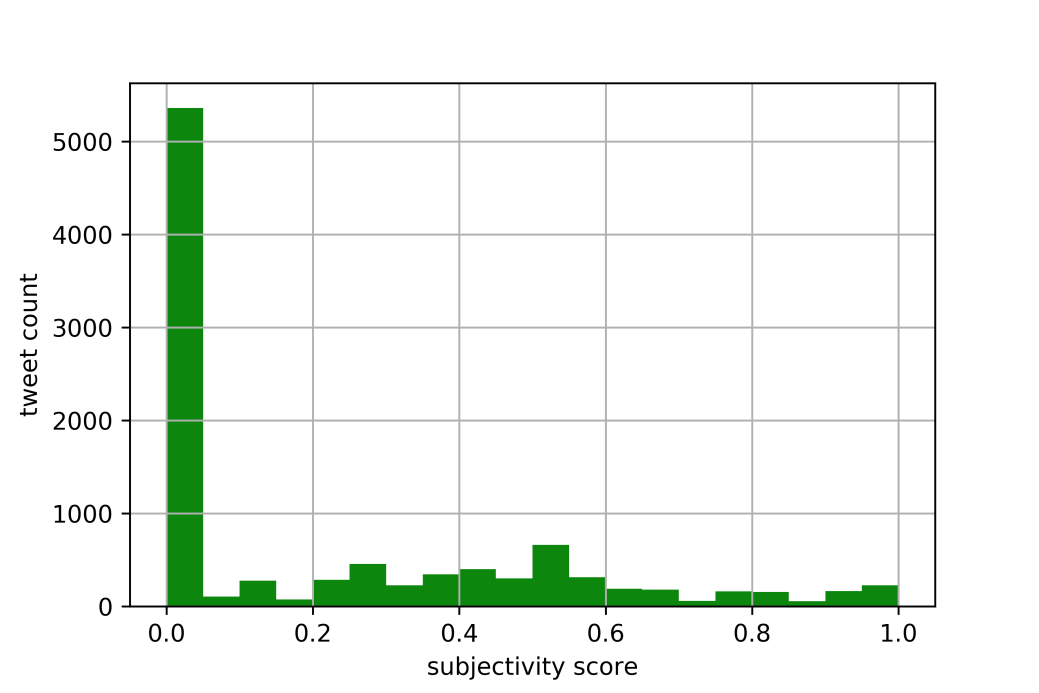
***n, bins, patches = plt.hist(sub\_list, bins=20, facecolor='green',alpha=0.95,)***

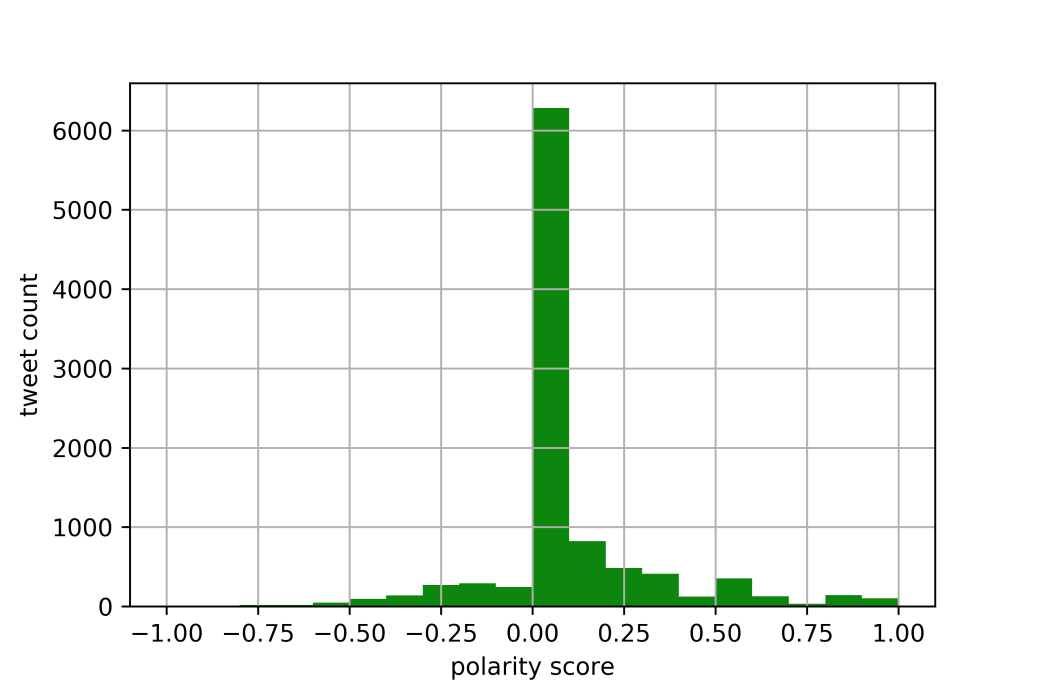
***(mu, sigma) = norm.fit(sub\_list)***

***y = mlab.normpdf( bins, mu, sigma)***

#(to get the polarity histogram, simply change all ‘sub\_list’ into ‘pol\_list’)

Our prediction to the histogram before running the visualization code is that: the subjectivity graph should be left skew because people tend to have very subjective opinions towards cannabis, either positive and subjective. For polarity graph, it should be somehow normally distributed because not many people are aggressively supporting or opposing cannabis. Below two graph shows the histogram for these 10K tweets’ subjectivity score distribution and polarity score distribution.





The subjectivity score histogram has shown us a very right skew distribution in regard to the 10K tweets’ subjectivity. With over 5000 tweets having a subjectivity score in the range of [0, 0.25], this means most tweets are objective, or at least stating facts which are neutral rather than subjective.

The polarity score histogram has shown us a very light tailed distribution, with over 6000 tweets having a polarity score in the rage of [0, 0.1]. This result somehow matches our prediction, that most people hold a neither positive nor negative attitudes towards ‘cannabis’.

With the following code, we also calculated the average subjectivity score and polarity score.

***print(sum(sub\_list)/len(sub\_list))***

***print(sum(pol\_list)/len(pol\_list))***

By running the code, we get 0.22230731969128828 as the average subjectivity score, and 0.07512991697691335 as the average polarity score.

**E. [Insights; 20 points] At the end of the day, what we want from all these analyses is the insights. Please describe the insights you gained from the analyses. I look forward to seeing your unique perspectives.**

As there are 10 K tweets, we would like to know the tweets are about indeed. By running the analysis of most frequent words and most frequent hashtags, we may be able to get a snippet of it but we would like to know what are the general topics that people are talking about related to the keyword ‘cannabis’. Therefore, we used the following code to generate vector score, based on the similarity of the tweets.

Instead of running the tweets one by one, we make a list to store per 100 tweets. The reason behind this is, we want to have a proper cluster and factor analysis. It is not reasonable to plot 10K points on the graph, neither will it give us some intuitive insights. The assumption we adopt using the text per hundred tweets is that, as we collect tweets by streaming, each 100 tweets reflect the most heated topics at that same time period. Thus, the variance of each 100 tweets(stored in the same json file) will not cause too many troubles in a statistical way. The following code shows how we collect the ‘clean text’ of 100 tweets and store it into the list where we will store all 10K tweets in 100 items.

***for item in tweets:***

***…***

***for word in text:***

***if word not in stopwords and len(word) > 1:***

***cleandoc += word + ' '***

***twtall.append(cleandoc)***

***cleandoc = ''***

***tweetcount += 1***

***twtname.append('{}-{}'.format(tweetcount\*100-99,tweetcount\*100))***

Then, with all the clean text from the 10K tweets, we want to generate vector score, which then can be used to generate the Multidimensional Scaling(MDS) plot. By running the following codes, we get our vectorizer score, MDS plot and dendrogram.

***vectorizer = TfidfVectorizer()***

***twt\_matrix = vectorizer.fit\_transform(twtall)***

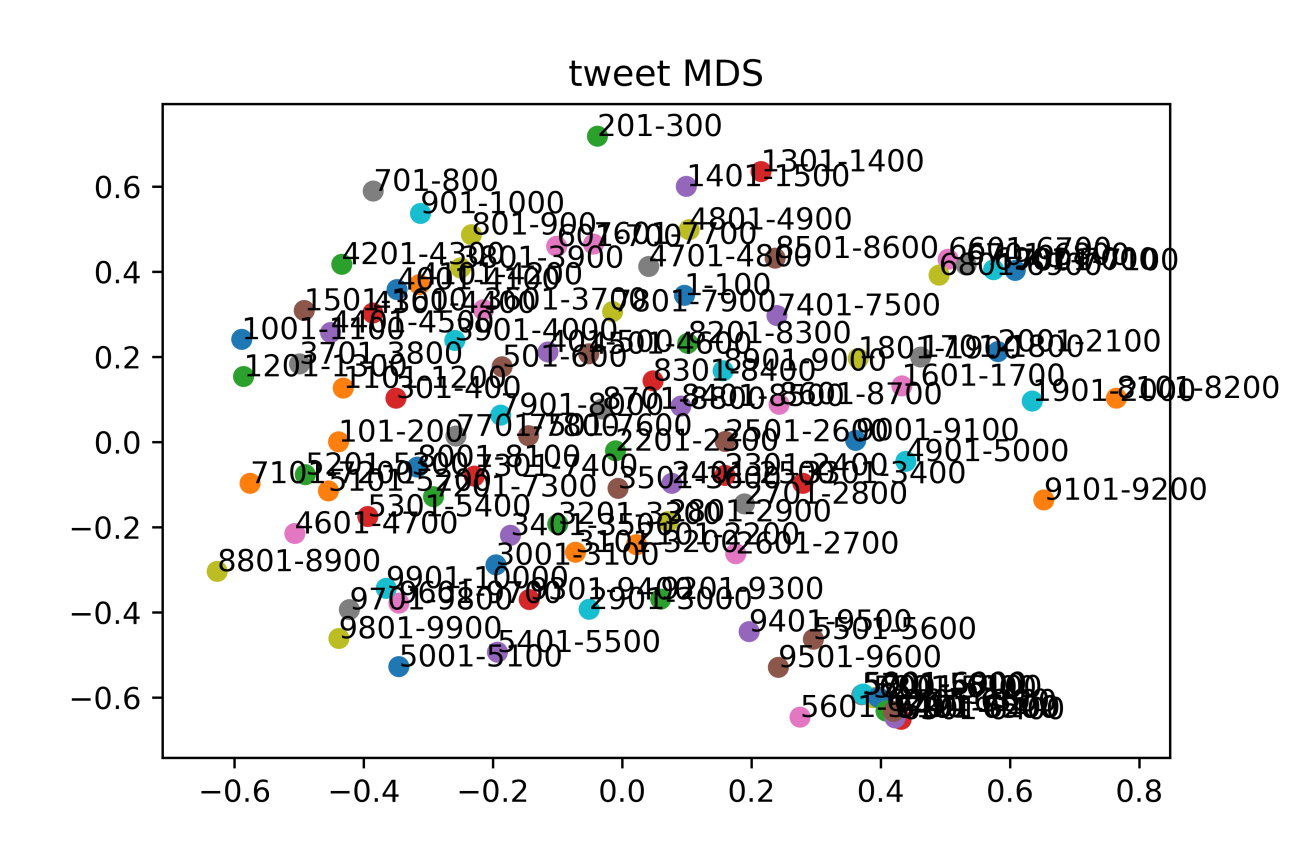
***cos\_dist = cosine\_distances(twt\_matrix)***

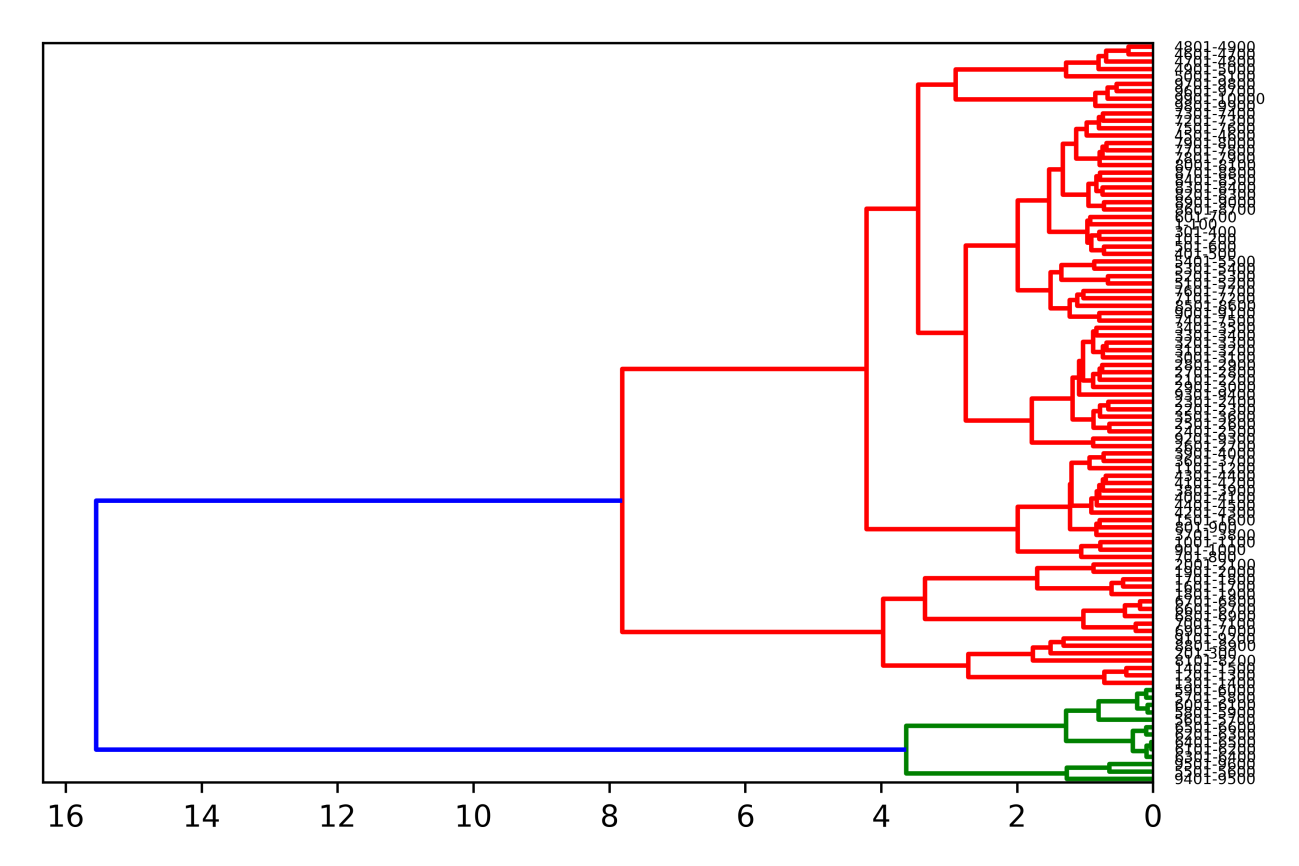
***mds = MDS(n\_components=2, dissimilarity='precomputed', random\_state=1)***

***pos = mds.fit\_transform(cos\_dist)***

***xs, ys = pos[:,0], pos[:,1]***

The following shows the two graph: MDS plot and dendrogram.





Based on the Dendrogram, we determine that all 10K tweets can be approximately divided into three clusters. Therefore, we run the following topic modeling with setting only 3 main topics. As we want to get more insights from each topic, we selected the most important (with the highest vector score) 30 words from each topics, by using the decomposition tool NMF package.

***num\_topic = 3***

***clf = decomposition.NMF(n\_components = num\_topic, random\_state=1)***

***doctopic = clf.fit\_transform(twt\_matrix)***

***print(num\_topic, clf.reconstruction\_err\_)***

***topic\_words = []***

***num\_top\_words = 30***

We then get the following three topics, with each 30 important words. When we use more than 30 topic words, there will be duplicate words in across topics. After we have the result, we then summarize each topic as: A. Cannabis industry legalization news; B. Cannabis and stock market; C. How people buy cannabis and how cannabis affects people.

***Topic A: Cannabis industry legalization news***

***{cannabis fruit marijuana medical amp weed business cbd step via crypto mustweed leafly mind legal stocks trekles news use medicinal new seeds industry like hemp high legalization california thc could}***

***Topic B: Cannabis and stock market***

***{momostocktrades twtr bidu pcln cgrw tmps mjna gwph beast this cnab weeks cbds rcgr plays aapl goog baba stock tradeideas insy zyne cara digaf motion cannabis atpt stocks gs fb }***

***Topic C: How people buy cannabis and how cannabis affects people***

***{buybudnow bcbud cannabis 20th opinion underage connectivity brain canadian alters attention supporters major january impact dramatically memory loss slow good finds protecting heard cells businesses small lots rules news retail}***

For topic A, we can briefly see that the industry has lots of related news in legalization in California, how cannabis recently become a heated news in the stock market, and how cannabis can be used as a medical treatment. People are also talking about planting cannabis legally (mainly in Canada). When we dig into the topic further, we can see people tweeting in US mainly talk about the marijuana stocks and the legalization process in California while people tweeting from Canada mostly talking about the upcoming July when the whole country will be open to cannabis. Such news are usually general and have some crossover with other topics.

Topic B is mainly about stock market. The short phrases shown in the topic words are mainly company stock code. After we run some research into these stock code, we realize that, except for some major tech firms from US, such as Apple, Google, Facebook and etc, most companies being mentioned are either companies making cannabis products, tobacco companies or companies related to the cannabis industry. More importantly, these companies’ stock price has recently pumped up at least 100% or more. This to some extent reflects how much the capital market favors in the uprising cannabis industry. All the further details of the company stock code are shown in the appendix.

We research on some topic words in topic C, and realize that ‘buybudnow’ and ‘bcbud’ are indeed website where people can buy cannabis product. This apparently shows people are talking about how to buy cannabis on twitter. Other key words such as ‘memory loss’, ‘slow’, ‘cells’, ‘impact’ and ‘underage’ reflect people’s discussion on how cannabis can affect cannabis users’ health. Generally speaking, people who oppose the legalization of cannabis will state that marijuana/cannabis is usually the entry level drug as most people who do drugs start with marijuana. Other who hold the opposite opinion will consider cannabis as a medical miracle as it can slow down the cell senescence process.

**Summary**

In general, by searching the keyword ‘cannabis’, we get a pretty good insight on how people think of marijuana, and how much it is recently being related to the stock market.

However, we are not very satisfied with the topic modeling as we don’t really get an intuitive insight by simply reading through the topic key words. I consider this as a natural flaw of streaming tweets because most of the tweets we stream, are retweets or quotes, which does not contain the full text of the original post. When the posted tweet is too long on the other hand, it will usually get shrink into shorter sentences. We try to deal with this problem by spanning the time length of streaming tweets into a week, and run extra streaming to get some useful information.

The three topics we drawn from the analysis are all related to cannabis, but in different area, which matching our cluster analysis. The subjectivity and polarity scores also reflect that these topics are more fact based rather than personal opinion, which have strong emotional expressions.

**Appendix**

twtr: Twitter, Inc.

Twitter, Inc. operates as a global platform for public self-expression and conversation in real time. The company offers various products and services, including Twitter that allows users to create, distribute, and discover content; and Periscope, a mobile application that enables user to broadcast and watch video live with others. It also provides promoted products and services, such as promoted tweets, promoted accounts, and promoted trends, which enable its advertisers to promote their brands, products, and services.

bidu: Baidu, Inc.

Baidu, Inc. provides Internet search services in China and internationally. It operates through three segments: Search Services, Transaction Services, and IQiyi. It offers Chinese language search platform on its Baidu.com Website that enables users to find relevant information online, including Web pages, news, images, documents, and multimedia files through links provided on its Website; and transaction platform, including Nuomi.com to connect online and offline services provided by third-parties.

pclr: The Priceline Group Inc.

The Priceline Group Inc. provides online travel and restaurant reservation, and related services. The company operates Booking.com, which offers online accommodation reservations services; and priceline.com that provides hotel, rental car, airline ticket, and vacation package reservation services. It also operates KAYAK, an online price comparison service that allows consumers to search and compare travel itineraries and prices, including airline ticket, accommodation reservation, and rental car reservation information from various travel Websites at once; and agoda.com, an online accommodation reservation service primarily for consumers in the Asia-Pacific region.

cgrw: CannaGrow Holdings, Inc.

CannaGrow Holdings, Inc. develops, designs, and builds grow facilities for legal cannabis industry in the State of Colorado. It offers design, permitting, development and construction, site management, staffing, research, and other professional services.

tmps: Tempus Applied Solutions Holdings, Inc.

Tempus Applied Solutions Holdings, Inc., through its subsidiaries, provides customized design, engineering, modification, integration, training, and operations solutions to support aircraft mission requirements. It serves the United States Department of Defense, the United States intelligence agencies, foreign governments, heads of state, and high net worth individuals worldwide.

mjna: Medical Marijuana, Inc.

Medical Marijuana, Inc., an investment holding company, operates in the medical marijuana and industrial hemp markets. Its products range from patented and proprietary based cannabinoid products to seed and stalk or isolated high value extracts manufactured and formulated for the pharmaceutical, nutraceutical, and cosmeceutical industries.

gwph: GW Pharmaceuticals plc

GW Pharmaceuticals plc, a biopharmaceutical company, engages in discovering, developing, and commercializing cannabinoid prescription medicines using botanical extracts derived from the Cannabis plant.

cnab: United Cannabis Corporation

United Cannabis Corporation owns intellectual properties related to growth, production, manufacture, marketing, management, utilization, and distribution of medical marijuana and marijuana infused products in the United States.

cbds: Cannabis Sativa, Inc.

Cannabis Sativa, Inc., through its subsidiaries, develops, manufactures, and sells herbal based skin care products in the United States. The company engages in the research, development, and licensing of natural cannabis products comprising cannabis formulations, edibles, topicals, strains, recipes, and delivery systems. In addition, it operates iBudtender, an online portal that offers information and patient reviews on marijuana dispensaries, cannabis businesses, marijuana strains, edibles, and concentrates and products.

rcgr: Rich Cigars, Inc.

Rich Cigars, Inc. manufactures and distributes cigars under the Rich Cigars brand name in the New Orleans, Louisiana area. The company intends to sell its cigars through the Internet, as well as through retail locations.

aapl: Apple Inc.

Apple Inc. designs, manufactures, and markets mobile communication and media devices, personal computers, and portable digital music players to consumers, small and mid-sized businesses, and education, enterprise, and government customers worldwide. The company also sells related software, services, accessories, networking solutions, and third-party digital content and applications.

goog: Alphabet Inc.

Alphabet Inc., through its subsidiaries, provides online advertising services in the United States, the United Kingdom, and rest of the world. The company offers performance and brand advertising services. It operates through Google and Other Bets segments. The Google segment includes principal Internet products, such as Search, Ads, Commerce, Maps, YouTube, Google Cloud, Android, Chrome, and Google Play, as well as technical infrastructure and newer efforts, including Virtual Reality.

baba: Alibaba Group Holding Limited

Alibaba Group Holding Limited, through its subsidiaries, operates as an online and mobile commerce company in the People's Republic of China and internationally. The company operates in four segments: Core Commerce, Cloud Computing, Digital Media and Entertainment, and Innovation Initiatives and Others. It operates Taobao Marketplace, a mobile commerce destination; Tmall, a third-party platform for brands and retailers; Rural Taobao program that enables rural residents and businesses to sell agricultural products to urban consumers

insy: Insys Therapeutics, Inc.

Insys Therapeutics, Inc., a specialty pharmaceutical company, develops and commercializes supportive care products. Its lead product candidate is SYNDROS, an orally administered liquid formulation of dronabinol for treating CINV and anorexia associated with weight loss in patients with AIDS. The company is also developing Cannabidiol Oral Solution, a synthetic cannabidiol for childhood catastrophic epilepsy syndromes; and other product candidates, including other dronabinol line extensions and sublingual spray product candidates.

zyne: Zynerba Pharmaceuticals, Inc.

Zynerba Pharmaceuticals, Inc., a clinical stage specialty pharmaceutical company, focuses on developing and commercializing proprietary synthetic cannabinoid therapeutics formulated for transdermal delivery. Its product candidates include ZYN002, which is in Phase II clinical trial for adult patients with refractory epileptic focal seizures and osteoarthritis, as well as pediatric patients with fragile X syndrome

cara: Cara Therapeutics, Inc.

Cara Therapeutics, Inc., a clinical-stage biopharmaceutical company, focuses on developing and commercializing chemical entities designed to alleviate pain and pruritus by selectively targeting kappa opioid receptors in the United States. It is developing product candidates that target the body's peripheral nervous system.

digaf: Digatrade Financial Corp.

Digatrade Financial Corp. operates as a digital asset exchange platform in Canada. The company offers blockchain development services and distributed ledger technology. It manages a trading and matching engine that offers multi-currency settlement, and real time FX pricing and risk management.

atpt: All State Properties Holdings, Inc.

All State Properties Holdings, Inc. does not have significant operations. Previously, it engaged in the land development, and construction and sale of residential housing businesses in the eastern United States and Argentina. The company is based in Lexington, Kentucky.

gs: The Goldman Sachs Group, Inc.

The Goldman Sachs Group, Inc. operates as an investment banking, securities, and investment management company worldwide. It operates through four segments: Investment Banking, Institutional Client Services, Investing & Lending, and Investment Management. The Investment Banking segment provides financial advisory services, including strategic advisory assignments related to mergers and acquisitions, divestitures, corporate defense activities, restructurings, spin-offs, and risk management; and underwriting services, such as debt and equity underwriting of public offerings and private placements of various securities and other financial instruments, as well as derivative transactions with public and private sector clients.

fb: Facebook, Inc.

Facebook, Inc. provides various products to connect and share through mobile devices, personal computers, and other surfaces worldwide. Its solutions include Facebook Website and mobile application that enables people to connect, share, discover, and communicate each other on mobile devices and personal computers; Instagram, a mobile application that enables people to take photos or videos, customize them with filter effects, and share them with friends and followers in a photo feed or send them directly to friends; Messenger, a messaging application to communicate with people and businesses across platforms and devices; and WhatsApp Messenger, a mobile messaging application.

\*Source: Yahoo Finance