covid19_sentiment

August 3, 2021

- Data Collection- Import data/ Data extraction: Web scrapping using Tweepy (Twitter API)
  - Setting up Tweepy Authorization
  - Scrapping tweets from a text search query
  - Renaming the file and saving it as CSV
- Data exploration
  - Importing libraries
  - Analyzing the dimentionality of the dataframe and having a look at the dataset
- Performing Sentiment Analysis, Visualization and NLP on the dataset
  - 1. Lemmatization: Remove all irrelevant characters such as any non alphanumeric characters, irrelevant words such as "@" twitter mentions or urls
  - 2. Tokenization: Tokenize your text by separating it into individual words
  - Convert all characters to lowercase, in order to treat words such as "hello", "Hello", and "HELLO" the same
  - 3. Creating Training and Testing models
  - 4. Creating Random Forest Classifier and verifying
- Visualizing / Word Cloud

1 Web scrapping using Tweepy (Twitter API)

Tweepy is a python library for using Twitter API that can be used for scrapping tweets. Once we obtain the developer credentials, dev environment has to be set up in developer dashboards & we will be able to use the Standard Twitter API The Twitter API can be used to programmatically retrieve and analyze data, as well as engage with the conversation on Twitter. This API provides access to a variety of different resources including the following: Tweets, Users, Direct Messages, Lists, Trends, Media,Places

Step 1: Apply and receive approval for a developer account

Step two: Save your App's key and tokens and keep them secure

Once you've been approved for developer access and have created a Project and App, you will be able to find or generate the following credentials within your developer App: - API Key - This is essentially a username, and allows you to make a request on behalf of your App. - API Key Secret - This is a password, and allows you to make a request on behalf of your App. - Access Token - This token represents the Twitter account that owns the App, and allows you to make a request

on behalf of that Twitter account. - Access Token Secret - This token also represents the Twitter account that owns the App, and allows you to make a request on behalf of that Twitter account. - Bearer Token - This token represents your App and enables you to authenticate requests that require OAuth 2.0 Bearer Token authentication.

You will use your API Key, API Key Secret, Access Token, and Access Token Secret to make requests that require OAuth 1.0a User Context authentication. If you would like to make requests on behalf of another user, you will need to use the 3-legged OAuth flow for them to authorize you. Since these keys and tokens do not expire unless regenerated, we suggest creating environment variables or using a secure password manager once you've received your credentials.

2 Data Collection Steps:

2.1 S1: Setting up Tweepy Authorization

[26]:
```
# Importing libraries

from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import tweepy
import json
import pandas as pd
import csv
import re
from textblob import TextBlob
import string
import os
import time
import wordcloud
```

[13]:
```
pip install preprocessor
```

```
Collecting preprocessor
  Downloading preprocessor-1.1.3.tar.gz (4.2 kB)
Building wheels for collected packages: preprocessor
  Building wheel for preprocessor (setup.py) … done
  Created wheel for preprocessor: filename=preprocessor-1.1.3-py3-none-
any.whl size=4475
sha256=9b5e6a63d379a5d08f254d7ac3446f609ca9224bd79639a57ccd4480225a5b96
  Stored in directory: /Users/Gaya/Library/Caches/pip/wheels/e4/4e/bf/0ecf68aa10
ee89d684d90437bd9f89ac19d5dc2921988bb59d
Successfully built preprocessor
Installing collected packages: preprocessor
Successfully installed preprocessor-1.1.3
Note: you may need to restart the kernel to use updated packages.
```

[51]:
```
pip install wordcloud
```

```
Collecting wordcloud
```

```
  Downloading wordcloud-1.8.1.tar.gz (220 kB)
     |                  | 220 kB 3.3 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.6.1 in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from wordcloud) (1.19.5)
Requirement already satisfied: pillow in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from wordcloud) (8.2.0)
Requirement already satisfied: matplotlib in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from wordcloud) (3.3.4)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: six in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from
cycler>=0.10->matplotlib->wordcloud) (1.15.0)
Building wheels for collected packages: wordcloud
  Building wheel for wordcloud (setup.py) … done
  Created wheel for wordcloud:
filename=wordcloud-1.8.1-cp38-cp38-macosx_10_9_x86_64.whl size=158421
sha256=f5880d4efc368df4ebeb58c0f31e86ee2a459a8c462b9400cb58f2da15a48fa1
  Stored in directory: /Users/Gaya/Library/Caches/pip/wheels/4d/3f/0d/a2ba9b7895
c9f1be89018b3141c3df3d4f9c786c882ccfbc3b
Successfully built wordcloud
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.1
Note: you may need to restart the kernel to use updated packages.
```

[4]:
```
# Obtaining twitter credentials from account

consumer_api_key= "sycrD6YSnRpyaw3qrXNOEUGh3"
consumer_api_secretkey= "Ii7bc4VRaoA7z7WQt5WEE5QAnn530dN7n886VjuHRTyox6AJE8"
access_token= "1409277512216109065-UDIzmZ3WBDJ0l1I5pfb7VctzQfu4WA"
access_token_secret= "YXKCrzG7m1sj3dZpM8LNXHLnXGFN9TZRg2vDM6UXiQoWs"

# Passing twitter credentials to Tweepy via OAuthHandler

auth= tweepy.OAuthHandler(consumer_api_key,consumer_api_secretkey)
auth.set_access_token(access_token,access_token_secret)
api= tweepy.API(auth)
```

2.2 S2: Scrapping tweets from a text search query

[25]:
```
text_query= 'COVID19Vaccine'
text_query= 'vaccineSideEffects'
count=500

# Creation of query method using Parameters  (using Try-Catch exception)

try:
    tweets= tweepy.Cursor(api.search, q=text_query).items(count)

    #     Pulling information from tweets iterable object
    tweets_list= [[tweet.created_at, tweet.id, tweet.text, tweet.user.
 ↪screen_name, tweet.user.location,
                   tweet.retweet_count, tweet.favorite_count] for tweet in␣
 ↪tweets]

    #     Creation of dataframe from Twertslist and we can add/remove tweet␣
 ↪info accordingly
    tweets_df= pd.DataFrame(tweets_list)

except BaseException as e:
    print('Failed', str(e))
    time.sleep(3)
```

2.3 S3: Renaming the file and saving it as CSV

[26]:
```
tweets_df.columns= ['Date','ID','Description','Username','User_Location',␣
 ↪'Retweet_Count', 'User_favorite_Count']
```

[27]:
```
tweets_df.to_csv('Covid_Vaccination_Tweets_Gaya.csv')
```

[28]:
```
tweets_df.to_csv('Covid_Vaccination_Tweets.csv')
```

3 Data exploration

3.0.1 Now lets use and explore the data....The above method has been used to scape a larger amount of data, so using that csv file for exploration

TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. Let us install Textblob...

[7]:
```
!pip install textblob
```

```
Requirement already satisfied: textblob in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (0.15.3)
Requirement already satisfied: nltk>=3.1 in
```

```
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from textblob) (3.6.1)
Requirement already satisfied: tqdm in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.1->textblob)
(4.59.0)
Requirement already satisfied: joblib in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.1->textblob)
(1.0.1)
Requirement already satisfied: click in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.1->textblob)
(7.1.2)
Requirement already satisfied: regex in
/Users/Gaya/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.1->textblob)
(2021.4.4)
```

Importing libraries, analyzing the dimentionality of the dataframe and having a look at the dataset...

[8]:
```
import pandas as pd
import matplotlib.pyplot as plot
from nltk.sentiment import SentimentIntensityAnalyzer
import re as re
```

[9]:
```
vaccine= pd.read_csv('/Users/Gaya/Dropbox/My Mac (Gayas-MacBook-Pro.local)/
 ↪Downloads/vaccination_all_tweets.csv')
```

[10]:
```
vaccine.shape
```

[10]:
```
(145025, 16)
```

[11]:
```
vaccine.describe()
```

[11]:

	id	user_followers	user_friends	user_favourites	\
count	1.450250e+05	1.450250e+05	145025.000000	1.450250e+05	
mean	1.389827e+18	1.038012e+05	1027.086971	1.270699e+04	
std	1.924603e+16	8.602818e+05	5098.714330	4.042375e+04	
min	1.337728e+18	0.000000e+00	0.000000	0.000000e+00	
25%	1.374312e+18	1.040000e+02	48.000000	6.500000e+01	
50%	1.393784e+18	5.100000e+02	269.000000	1.248000e+03	
75%	1.407239e+18	1.804000e+03	861.000000	7.919000e+03	
max	1.418861e+18	1.590058e+07	516578.000000	1.221784e+06	

	retweets	favorites
count	145025.000000	145025.000000
mean	2.722524	12.044275
std	51.634509	181.788373
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	1.000000
75%	1.000000	3.000000

```
max      11288.000000   25724.000000
```

3.0.2 1. Let us extract the vaccines mostly used and spoken about worldwide

[13]:
```
# def search_words(text):
#     result = re.findall(r'\b['sick']+\b', text)
#     return " ".join(result)

# df['only_words']=df['address'].apply(lambda x : search_words(x))
# print("\nOnly words:")
# print(df)
```

A lambda function is a small function containing a single expression. Lambda functions can also act as anonymous functions where they don't require any name. These are very helpful when we have to perform small tasks with less code. Weuse lambda functions when we have to pass a small function to another function

Lambda with Apply: We can use the apply() function to apply the lambda function to both rows and columns of a dataframe. If the axis argument in the apply() function is 0, then the lambda function gets applied to each column, and if 1, then the function gets applied to each row.

[14]:
```
vaccines = ['Pfizer','BioNTech','Sinopharm','Sinovac','Moderna','Oxford',

               'AstraZeneca','Covaxin', 'Sputnik V']
vaccine['vaccine_name'] = vaccine.text.apply(lambda text:
                              [v_name for v_name in vaccines if v_name.lower()␣
 ↪in text.lower()]
                             )
# apply- Apply a function along an axis of the DataFrame
```

2. Now that we have the vaccine names for each location, let us look into the location used and calculate the percentage for User location. It can help us gain knowledge about what vaccines where used more in which location

[15]:
```
vaccine.user_location.value_counts()

# Calculating share of voice
```

[15]:
```
India                              5694
Bengaluru, India                   2858
Toronto, Canada  and Worldwide     2338
New Delhi, India                   2238
Mumbai, India                      1393
                                   …
Golden State of mind                  1
city of angles                        1
Kuopio, Suomi                         1
TPE-YKA-BRO-MFE-SFO-SJC               1
Manchester, Europe.                   1
```

```
Name: user_location, Length: 19418, dtype: int64
```

[16]:
```
# Percentages for each User Location

vaccine.user_location.value_counts()/vaccine.shape[0]*100
```

[16]:
```
India                              3.926220
Bengaluru, India                   1.970695
Toronto, Canada  and Worldwide     1.612136
New Delhi, India                   1.543182
Mumbai, India                      0.960524
                                     …
Golden State of mind               0.000690
city of angles                     0.000690
Kuopio, Suomi                      0.000690
TPE-YKA-BRO-MFE-SFO-SJC            0.000690
Manchester, Europe.                0.000690
Name: user_location, Length: 19418, dtype: float64
```

We already know the countries that used the respective vaccines: - Canada: Pfizer, Moderna and AstraZeneca (eventually stopped) - USA: Pfizer, Moderna - India: Covaxin, CoviShield (AstraZeneca), Pfizer - UK: Oxford AstraZeneca, Pfizer, Moderna - Russia: Spuitnik- V - China:Sinopharm, Sinovac

3. Let us see what the users have to say regarding getting vaccinated...

[17]:
```
vaccine['is_vaccinated']=vaccine['text'].str.count('^[Vaccinated].*')>0
```

[18]:
```
vaccine[vaccine.is_vaccinated==True]
```

[18]:

	id	user_name	user_location	\
6	1337851215875608579	Gunther Fehlinger	Austria, Ukraine and Kosovo	
34	1337742528108519424	Rajat Kotra	London, England	
166	1338335155849752580	Shelley Uppal	NaN	
413	1338839502001987586	kmfm News	Kent, UK	
421	1338820828067205120	Zoheb Ahmad	Manchester, England	
…	…	…	…	
142215	1418777147175362561	Zeno Health	Mumbai, India	
143012	1418582396090290179	Ananth Rupanagudi	NaN	
143179	1418550691610521602	Shaji	Planet Earth	
143546	1418481233957244928	Ponmurugan P	Chennai, India	
144807	1417995438163066880	Rajkumar	India	

	user_description	\
6	End North Stream 2 now - the pipeline of corru…	
34	Ever-curious polymath,#global leader#startups#…	
166	Aspiring dermatologist •Albany Medical College…	
413	The latest Kent headlines from the kmfm news t…	
421	Northern, Man United fan, left-footed, lover…	

```
…                                                     …
142215  Mumbai's fastest growing pharmacy retail chain…
143012  Earlier @rananth. Railway bureaucrat. Rishi Va…
143179             Global Warning against Global Weirding
143546  Proud father, Family man, fan of Arsenal FC an…
144807                                                NaN

              user_created  user_followers  user_friends  user_favourites  \
6       2013-06-10 17:49:22            2731          5001            69344
34      2009-12-03 12:03:28             754           917              220
166     2020-03-16 12:21:00             160           255              727
413     2011-04-27 20:37:09           19486           854              301
421     2008-03-03 00:13:59            1029           595            48488
…                       …              …             …               …
142215  2017-09-01 14:37:00             196             1                1
143012  2021-06-04 09:46:35            1334           130             7127
143179  2012-03-29 20:06:12             177          1103             2019
143546  2009-02-09 09:21:08              40            78              179
144807  2017-07-22 08:29:21              68           198            12765

        user_verified                 date  \
6               False  2020-12-12 20:06:00
34              False  2020-12-12 12:54:07
166             False  2020-12-14 04:09:00
413              True  2020-12-15 13:33:06
421             False  2020-12-15 12:18:54
…                   …                   …
142215          False  2021-07-24 03:36:46
143012          False  2021-07-23 14:42:53
143179          False  2021-07-23 12:36:54
143546          False  2021-07-23 08:00:54
144807          False  2021-07-21 23:50:32

                                                    text  \
6       it is a bit sad to claim the fame for success …
34      Vaccine!! Anyone?? #covid #Pfizervaccine #Pfiz…
166     Vaccines save lives  Hopefully HCW can assuag…
413     VIDEO: Elderly patients on the Isle of #Sheppe…
421               Vaccinated  \n#PfizerBioNTech #COVID19
…                                                     …
142215  Vaccination today means a better tomorrow for …
143012  Vaccines at play!   #COVIDVaccines #Pfizer #…
143179  Vaccines Throw a Party \nhttps://t.co/CTFTQ45N…
143546  Vaccine status not yet updated even after 7 da…
144807  Vacinnated first dose #SputnikV https://t.co/v…

                                                hashtags  \
```

```
6                                          ['vaccination']
34           ['covid', 'Pfizervaccine', 'PfizerBioNTech']
166                                                   NaN
413                            ['Sheppey', 'coronavirus']
421                          ['PfizerBioNTech', 'COVID19']
…                                                      …
142215                                                NaN
143012  ['COVIDVaccines', 'Pfizer', 'Astrazenaca', 'Mo…
143179  ['vaccine', 'AstraZeneca', 'COVID19', 'Pfizer'…
143546                                                NaN
144807                                       ['SputnikV']

                    source  retweets  favorites  is_retweet  \
6          Twitter Web App         0          4       False
34      Twitter for Android         0          0       False
166      Twitter for iPhone         0          0       False
413              TweetDeck         0          0       False
421      Twitter for iPhone         0          1       False
…                       …       …          …           …
142215  Twitter for Android         0          1       False
143012  Twitter for Android         2          6       False
143179   Twitter for iPhone         0          0       False
143546  Twitter for Android         0          0       False
144807  Twitter for Android         0          0       False

                          vaccine_name  is_vaccinated
6                                   []           True
34                  [Pfizer, BioNTech]           True
166                                 []           True
413                                 []           True
421                 [Pfizer, BioNTech]           True
…                                  …            …
142215                              []           True
143012     [Pfizer, Moderna, Covaxin]           True
143179  [Pfizer, Moderna, AstraZeneca]           True
143546                      [Covaxin]           True
144807                              []           True

[2212 rows x 18 columns]
```

Let us look into what people say regarding getting SICK about vaccines and analyze the sentiment later too...

[19]:
```
vaccine['is_sick']=vaccine['text'].str.count('^[sick].*')>0
vaccine[vaccine.is_sick==True].head()
```

[19]:
```
                     id                   user_name  \
6     1337851215875608579            Gunther Fehlinger
```

```
581   1339239111643852801          RWinstanley-Chesters
626   1342065115714224128                       Alex777
1690  1343549914169683969  drosophila gene name (pablo)
2525  1346643634133856262                        meagan

                  user_location  \
6     Austria, Ukraine and Kosovo
581                         NaN
626                         NaN
1690            Santiago, Chile
2525                  arkansas

                                    user_description         user_created  \
6     End North Stream 2 now - the pipeline of corru…  2013-06-10 17:49:22
581   Dr Winstanley-Chesters, University of Leeds, B…  2011-12-14 14:52:55
626   Love Animals, Art, Books & Soccer. I hate evil…  2017-04-18 14:00:07
1690  developmental biology enthusiast, florence wel…  2010-04-03 15:33:44
2525            excuse me, which level of hell is this?  2014-04-01 00:50:35

      user_followers  user_friends  user_favourites  user_verified  \
6               2731          5001            69344          False
581              628            52               33          False
626             2117          2327            50706          False
1690            1035          2173            85848          False
2525             659           479             1320          False

                    date                                                text  \
6     2020-12-12 20:06:00  it is a bit sad to claim the fame for success …
581   2020-12-16 16:01:00  cried a little when my 83 year old auntie deci…
626   2020-12-24 11:10:32  so Ugur the #Turkish Chief &amp;amp; owner of #Bio…
1690  2020-12-28 13:30:36  cool things I didn't know about the #PfizerBio…
2525  2021-01-06 02:23:56  i did it y'all!  the silver lining to a shit-t…

                                          hashtags              source  \
6                                   ['vaccination']     Twitter Web App
581                               ['PfizerBioNTech']     Twitter Web App
626   ['Turkish', 'BioNtech', 'PfizerBioNTech', 'Vac…     Twitter Web App
1690                              ['PfizerBioNTech']    Twitter for iPad
2525  ['vaccinated', 'covid', 'PfizerBioNTech', 'Thi…  Twitter for iPhone

      retweets  favorites  is_retweet        vaccine_name  is_vaccinated  \
6            0          4       False                  []           True
581          0          2       False  [Pfizer, BioNTech]           True
626          0          1       False  [Pfizer, BioNTech]          False
1690         0          0       False  [Pfizer, BioNTech]           True
2525         0          5       False  [Pfizer, BioNTech]           True
```

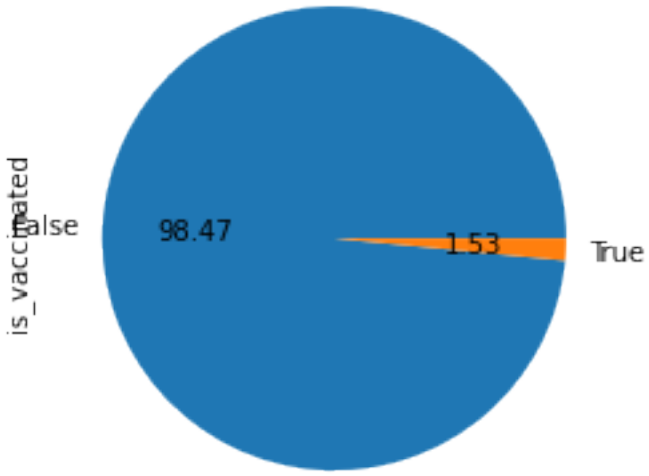
```
      is_sick
6        True
581      True
626      True
1690     True
2525     True
```

[20]:
```
# print(vaccine['vaccine_name'], vaccine['user_location'], vaccine[vaccine.
↪is_sick]==True).distinct()

vaccine[vaccine.is_sick==True][['vaccine_name','user_location']]
```

[20]:
```
              vaccine_name                user_location
6                       []  Austria, Ukraine and Kosovo
581     [Pfizer, BioNTech]                          NaN
626     [Pfizer, BioNTech]                          NaN
1690    [Pfizer, BioNTech]             Santiago, Chile
2525    [Pfizer, BioNTech]                    arkansas
…                      …                           …
139544           [Sinovac]   Gampola(H) Nawalapitiya(A)
140319            [Pfizer]                  Miami, Fl
140372                  []             Minneapolis, MN
140892           [Moderna]          Santa Rosa, Laguna
141453                  []                   she/her

[287 rows x 2 columns]
```

[21]:
```
vaccine.is_vaccinated.value_counts().plot(kind='pie', autopct='%0.02f')
```

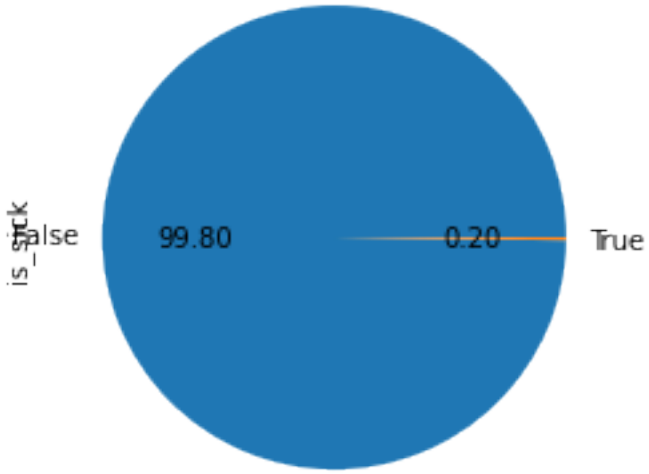
[21]:
```
<AxesSubplot:ylabel='is_vaccinated'>
```



We can find that a lot of people talk about it after being vaccinated or talk about being vaccinated..only 1.5% (around 2%) of the people have spoken about something other than the vaccination process. We can ananlyse this better by conding a sentiment analysis which we will be doing later

[22]:
```
vaccine.is_sick.value_counts().plot(kind='pie', autopct='%0.02f')
```

[22]:
```
<AxesSubplot:ylabel='is_sick'>
```

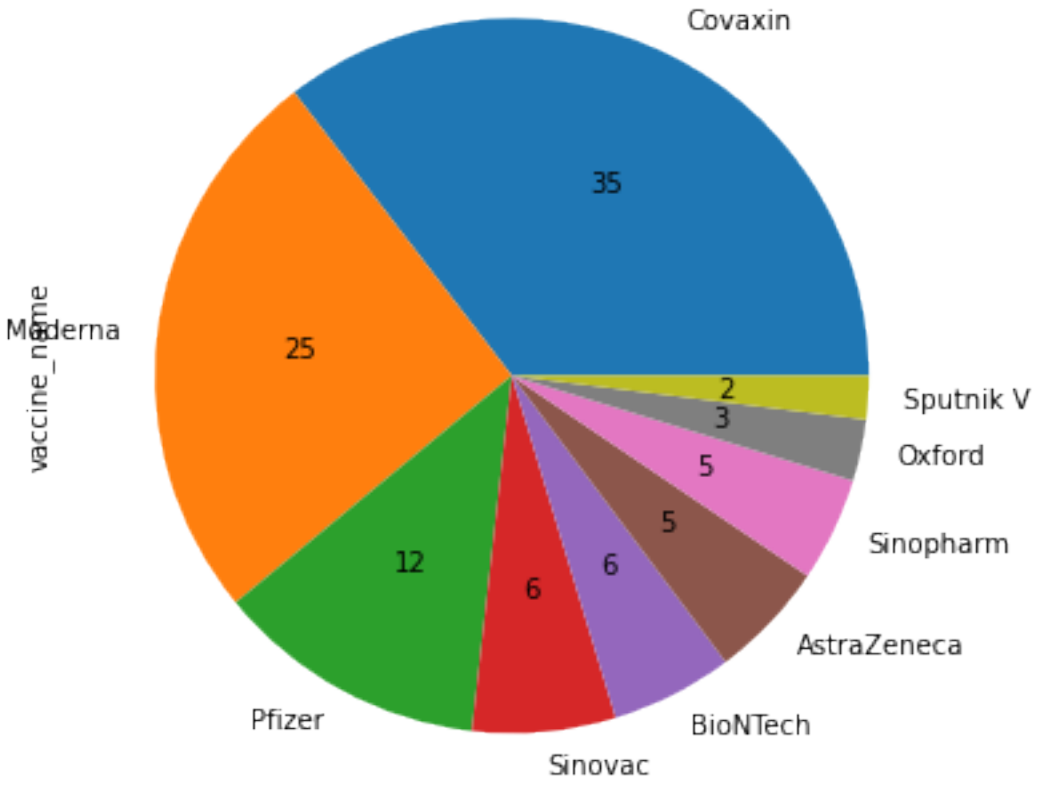


From the above, we can decipher that the number of people who are extremely sick due to the vaccine is very less. The flout is more around the vaccination process or which vaccine to use

[23]:
```
# Lets look into the percentage of vaccines spoken about

# Here we try to data explode from the list in order to find percentage of␣
 ↪vaccienes spoken about
# pandas explode: Transform each element of a list-like to a row, replicating␣
 ↪index values.

#Setting Plot Size
plot_size = plot.rcParams["figure.figsize"]

plot_size[0] = 6
plot_size[1] = 10
plot.rcParams["figure.figsize"] = plot_size
```

```
vaccine.vaccine_name.explode().value_counts().plot(kind='pie', autopct='%1.0f')
```

[23]: <AxesSubplot:ylabel='vaccine_name'>



[24]:
```
vaccine.vaccine_name.explode().value_counts()/vaccine.shape[0]*100
```

[24]:
```
Covaxin        28.881917
Moderna        20.779176
Pfizer         10.009309
Sinovac         5.290122
BioNTech        4.499914
AstraZeneca     4.336494
Sinopharm       3.840717
Oxford          2.272022
Sputnik V       1.594208
Name: vaccine_name, dtype: float64
```

From the pie chart and percentage of vaccines used, it is evident that the most talked about vaccines are Covaxin, Moderna and Pfizer. There is very less data on the vaccines used in China and Russia and people in those countries havent given much opinions nor disclosed much info rmation about them.

Lets look into what subjectivity and polarity is before conding SA... Sentiment polarity for an element defines the orientation of the expressed sentiment, i.e., it determines if the text expresses the positive, negative or neutral sentiment of the user about the entity in consideration. Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. This indicates emotional charge of a statement or passage

Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information.

The identification of subjective statements from the data is known as subjectivity detection. The aim is to find the opinionative data and classify it according to its polarity, i.e. positive, negative or neutral feedback, known as sentiment classification and then analysing it which is known as sentiment analysis

Subjectivity is float within the range [0,1] where 0 is very objective and 1 is very subjective

3.1 Performing Sentiment Analysis & Visualization

3.1.1 S1: Calculating Polarity and Subjectivity

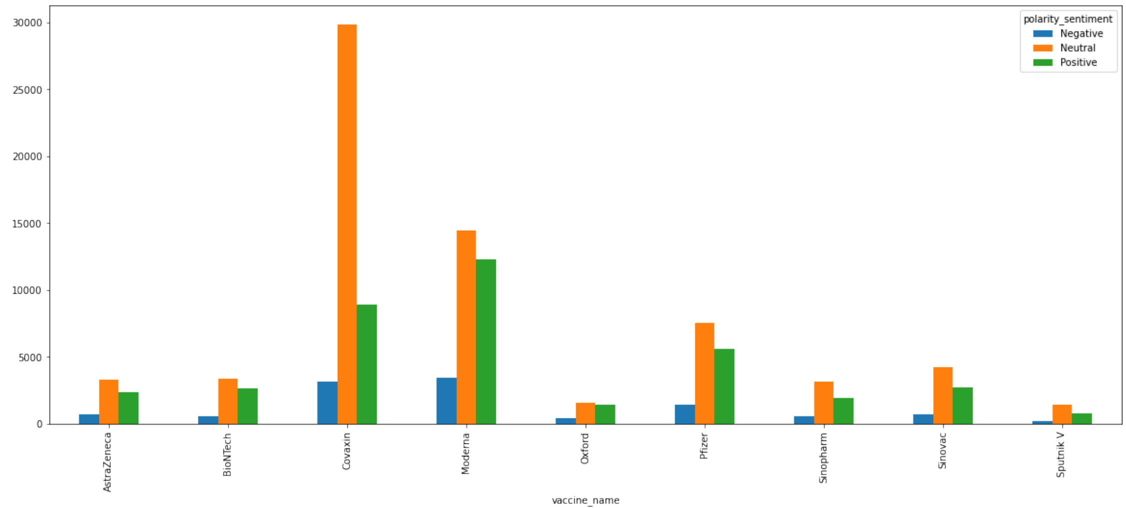
[27]:
```
vaccine['polarity']=vaccine['text'].apply(lambda x: TextBlob(x).sentiment.
 ↪polarity)
vaccine['subjectivity']=vaccine['text'].apply(lambda x: TextBlob(x).sentiment.
 ↪subjectivity)
```

[29]:
```
def polarity_check(polarity_val):
    if polarity_val == 0:
        return "Neutral"
    elif polarity_val > 0:
        return "Positive"
    elif polarity_val < 0:
        return "Negative"

vaccine['polarity_sentiment']=vaccine.polarity.apply(polarity_check)
```

[30]:
```
vaccine_exploded = vaccine.explode('vaccine_name')
```

[39]:
```
# ax = vaccine.plot.bar(x='vaccine_name', y='polarity_sentiment', rot=0)]

vaccine_group = vaccine_exploded.groupby(['vaccine_name','polarity_sentiment']).
 ↪size().unstack(fill_value=0)
ax = vaccine_group.plot.bar()

plot_size[0] = 20
```

```
plot_size[1] = 10
plot.rcParams["figure.figsize"] = plot_size
```



From the bar chart visualization, we can find that Covaxin, Moderna and Pfizer are the vaccines most used . There are a lot of neutral sentiments regarding the vaccines. But all the vaccines have more postitive and neutral sentiments than negative ones. In such cases, the subjectivity can also be considered, and more data can be collected on Sputnik V and Sinovac

[42]:
```
vaccine.head(3)
```

[42]:
```
                    id     user_name               user_location  \
0  1340539111971516416    Rachel Roh  La Crescenta-Montrose, CA
1  1338158543359250433  Albert Fong           San Francisco, CA
2  1337858199140118533          eli                    Your Bed

                                    user_description         user_created  \
0  Aggregator of Asian American news; scanning di…  2009-04-08 17:52:46
1  Marketing dude, tech geek, heavy metal & '80s …  2009-09-21 15:27:30
2                                        heil, hydra  2020-06-25 23:30:28

   user_followers  user_friends  user_favourites  user_verified  \
0             405          1692             3247          False
1             834           666              178          False
2              10            88              155          False

                  date  …               source retweets favorites  \
0  2020-12-20 06:06:44  …  Twitter for Android        0         0
1  2020-12-13 16:27:13  …      Twitter Web App        1         1
2  2020-12-12 20:33:45  …  Twitter for Android        0         0
```

```
   is_retweet                                vaccine_name  is_vaccinated  \
0       False                          [Pfizer, BioNTech]          False
1       False                                          []          False
2       False  [Pfizer, BioNTech, Moderna, AstraZeneca]          False

  is_sick  polarity  subjectivity  polarity_sentiment
0   False       0.0      0.125000             Neutral
1   False      -0.5      0.900000            Negative
2   False       0.0      0.033333             Neutral

[3 rows x 22 columns]
```

[63]:
```
# Calculate Sentiment using Vader in NLTK

# from nltk.sentiment.vader import SentimentIntensityAnalyzer

# sia=SentimentIntensityAnalyzer()
# sentiment= [None]*vaccine['text'].shape[0]
# for i, vac in vaccine.iterrows():
#     sentiment[i]=sia.polarity_scores(vac)
# vaccine.insert(vaccine.shape[1],'sentiment', sentiment)
```

Let us try to use text blob..TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks.

S1: Stemming and Featuring- Creating Classifiers

[47]:
```
#Create Feature and Labels Set
features = vaccine['text'].values
labels = vaccine['polarity_sentiment'].values
```

[48]:
```
#Remove Special Characters

processed_features = []

for sentence in range(0, len(features)):
    # Remove all the special characters
    processed_feature = re.sub(r'\W', ' ', str(features[sentence]))

    # remove all single characters
    processed_feature= re.sub(r'\s+[a-zA-Z]\s+', ' ', processed_feature)

    # Remove single characters from the start
    processed_feature = re.sub(r'\^[a-zA-Z]\s+', ' ', processed_feature)

    # Substituting multiple spaces with single space
    processed_feature = re.sub(r'\s+', ' ', processed_feature, flags=re.I)
```

```
    # Removing prefixed 'b'
    processed_feature = re.sub(r'^b\s+', '', processed_feature)

    # Converting to Lowercase
    processed_feature = processed_feature.lower()

    processed_features.append(processed_feature)
```

[49]:
```
import nltk
from random import shuffle
```

[ ]:
```
nltk.download()
```

S2: Tokenizing

[50]:
```
#Vectorize- Tokenizing

from nltk.corpus import stopwords
nltk.download('stopwords')
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(max_features=2500, min_df=7, max_df=0.8,␣
 ↪stop_words=stopwords.words('english'))
processed_features = vectorizer.fit_transform(processed_features).toarray()
```

```
[nltk_data] Downloading package stopwords to /Users/Gaya/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

S3: Creating Testing and Training

[51]:
```
#Create Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(processed_features,
                                                    labels, test_size=0.2,
                                                    random_state=100)
```

S4: Creating classifier algorithms and Evaluating model accuracy

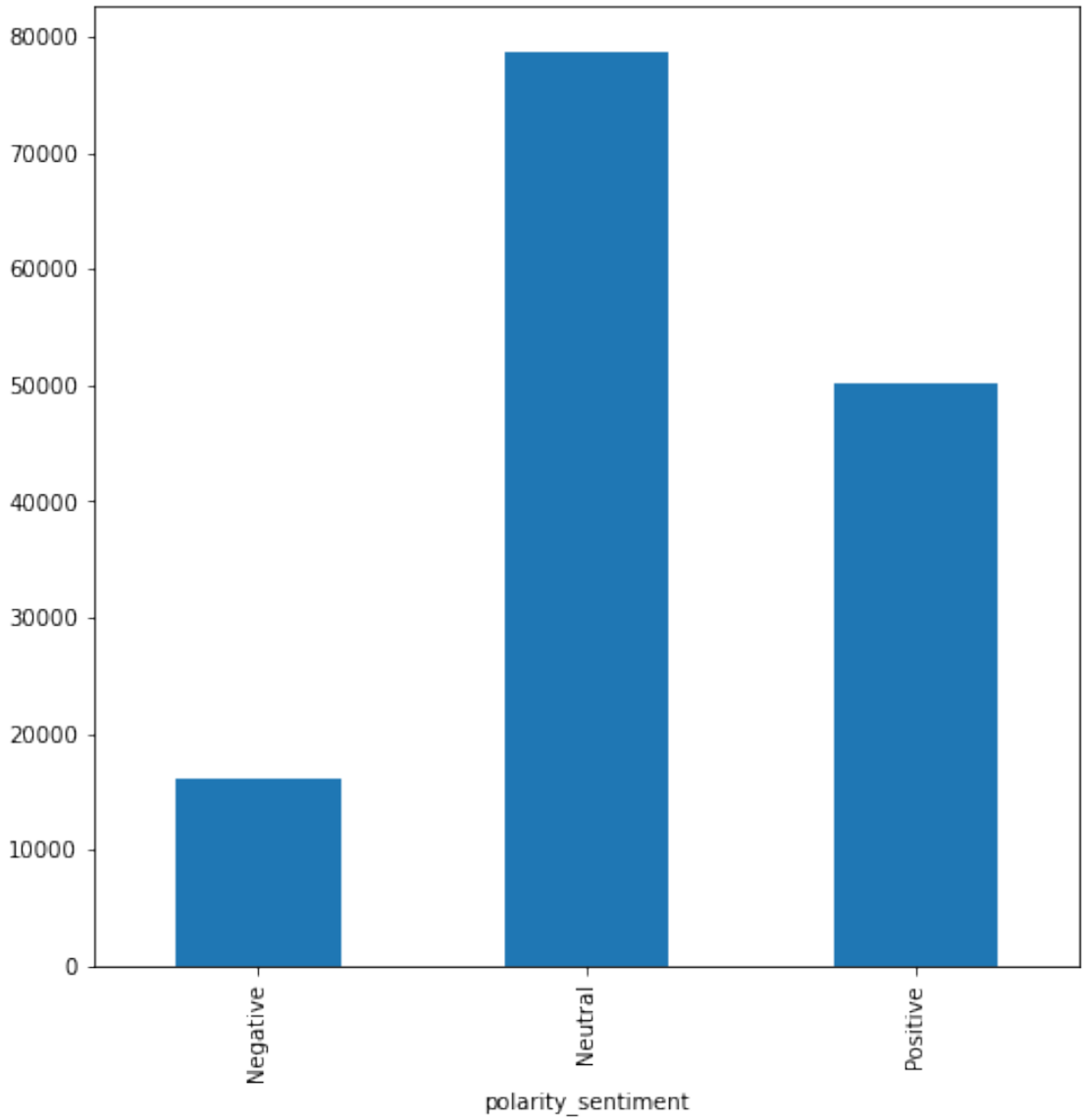
[52]:
```
#Create Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
text_classifier = RandomForestClassifier(n_estimators=200, random_state=100)
text_classifier.fit(X_train, y_train)

#Evaluation Report and Matrix
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
predictions = text_classifier.predict(X_test)
print(confusion_matrix(y_test,predictions))
print('\n',classification_report(y_test,predictions))
print('\nAccuracy Score: {:.2f}'.format(accuracy_score(y_test, predictions)))
```

```
[[ 1971   730   451]
 [  107 15306   270]
 [  188  1079  8903]]

              precision    recall  f1-score   support

    Negative       0.87      0.63      0.73      3152
     Neutral       0.89      0.98      0.93     15683
    Positive       0.93      0.88      0.90     10170

    accuracy                           0.90     29005
   macro avg       0.90      0.83      0.85     29005
weighted avg       0.90      0.90      0.90     29005


Accuracy Score: 0.90
```

The model is great as we got an accutracy of 90%

3.1.2 S5: Lets visualize the total sentiment... whether it is positive, negative or neutral..

[62]:
```
vaccine_sentiment = vaccine.groupby(['polarity_sentiment']).size()
ax = vaccine_sentiment.plot.bar()

plot_size[0] = 5
plot_size[1] = 10
plot.rcParams["figure.figsize"] = plot_size
```

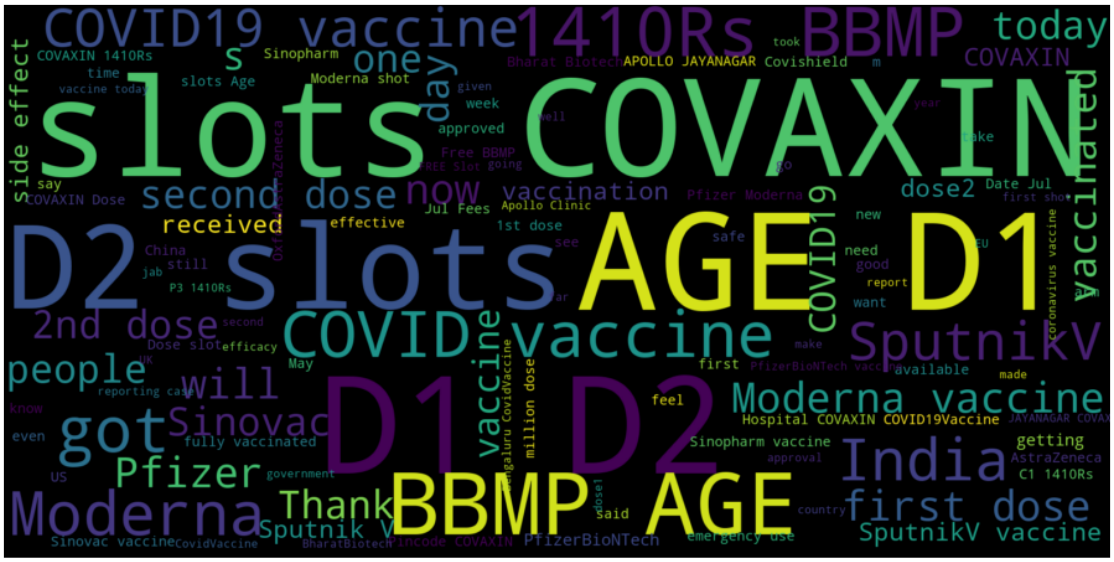


The overall negative sentiment is very less, so people are more focused on getting the second doses and having very less side effects. We can cross check the information from. thee word cloud too

WordCloud:

[56]:
```
# Wordcloud
from wordcloud import WordCloud, STOPWORDS
def show_wordcloud(data, title=""):
    text = " ".join(t for t in data.dropna())
    stopwords = set(STOPWORDS)
    stopwords.update(["t", "co", "https", "amp", "U"])
    wordcloud = WordCloud(stopwords=stopwords, scale=4, max_font_size=50,␣
 ↪max_words=500,background_color="black").generate(text)
```

```
    fig = plot.figure(1, figsize=(16,16))
    plot.axis('off')
    fig.suptitle(title, fontsize=20)
    fig.subplots_adjust(top=2.3)
    plot.imshow(wordcloud, interpolation='bilinear')
    plot.show()

show_wordcloud(vaccine['text'], title = 'Prevalent words in tweets')
```



Prevalent words in tweets

The most prevalent words in the word cloud are: Age, slots, first dose, second dose, COVID vaccine, got, D1 and D2. From the word cloud, we can get to understand that slots for the vaccine , age groups and the second dose of vaccine taken, divisions (D1 and D2) categorized by the government are what is most spoken about. The side effects/ getting sick is least spoken about, so that can be interpreted that the vacination process/ rollout is going great around the worls and side effects are insignificant

3.2 Conclusion

We can find that the COVID-19 vaccination roll out process is effective and going great. People are having the least side effects on their heath. The overall sentiment is positive.

[23]:
```
# References
```