Midterm

June 28, 2022

A Natural Language Processing pipeline for Twitter

Lucas Viola

University of London

0.1 Summary

0.2 1. Aims, objectives and background

0.2.1 1.1 Introduction

This project presents a Data Pipeline built in Python leveraging the power of tools such as NLTK for Natural Language Processing, Pandas for dealing with different file types, Numpy for statistics and HTTP for data retrieval through the Requests Tool. It aims to be a tool for processing Tweets from different users, analyze them using NLP techniques such as Sentiment Analysis and Frequency Distribution and generate insights and visualizations on top of the results.

0.2.2 1.2 Purpose of this project

The purpose of this project is to understand the behavior of Twitter's users and how they handle themselves exposing their opinions online, the analysis is made on top of parameters such as frequency and classification of Tweets and Words, and the resulting sentiment of positiveness or negativeness of such bodies. It also aims to help visualize the most common words and expressions used by said user.

0.2.3 1.3 Data Gathering methodology

Data here is generated through an HTTP GET request to Twitter's Open API using a OAuth Authentication. It then returns a list of the latest 100 tweets of the chosen user. All processing is made on top of these 100 tweets.

0.3 2. Data

0.3.1 2.1 Data Source

The single source of data is Twitter's Recent-Search REST API, using as parameter an ID which is generated by querying Twitter's User API by the user's handle. It is then transformed into a list of tweets.

User Handle Input: Type the user handle input below

[71]:
```
# Add user handle in the variable below. If left blank it will use the default␣
↪json file containing Elon Musk's Tweets
# as an example
user_handle= 'elonmusk'

# Insert your Twitter API OAuth2 Bearer Token here.
# Learn how to generate yours here: https://developer.twitter.com/en/docs/
↪twitter-api/tweets/lookup/introduction
authToken = ''
```

[72]:
```
# Request to retrieve the user id based on an user's handle
import requests

def get_user_id(handle):

    if handle == '':
        return -1

    url = 'https://api.twitter.com'
    resource = '/2/users/by/username/'+ handle
    endpoint = url+resource

    headers = {'Authorization': 'Bearer '+authToken}

    response = requests.get(endpoint, headers=headers)

    print('GET Request to: ', response.url)
    print('Response code: ', response.status_code)
    if 'errors' in response.json():
        print('Error making the request')
        return -1

    return response.json()['data']['id']

user_id = get_user_id(user_handle)
```

```
GET Request to:  https://api.twitter.com/2/users/by/username/elonmusk
Response code:  200
```

[73]:
```
# Request to retrieve the tweets based on an user ID
import requests
import json

# Loading the default dataset from the system
# If an error occurs, will default to this file to analyse
file = open('tweets.json')
default_json = json.loads(file.read())

def get_tweets(user):
    url = 'https://api.twitter.com'
    resource = '/2/users/'+str(user_id)+'/tweets/'
    endpoint = url+resource

    params = {'max_results':100}

    headers = {'Authorization': 'Bearer'+authToken}

    response = requests.get(endpoint, params, headers=headers)

    print('GET Request to: ', response.url)
    print('Response code: ', response.status_code)

    if response.status_code != 200:
        print('Error making the request. Returning default dataset')
        return default_json

    return response.json()

json_dataset = get_tweets(user_id)

json_dataset
```

```
GET Request to:
https://api.twitter.com/2/users/44196397/tweets/?max_results=100
Response code:  401
```

```
Error making the request. Returning default dataset
```

```
[73]: {'data': [{'id': '1539375908800368641',
         'text': '@thesheetztweetz Their attempt to bait and switch satellite spectrum
       for cellular spectrum is super shady and unethical. \n\nIf they are successful,
       it would hurt the least served and completely unserved of the world. Very messed
       up.'},
        {'id': '1539325996897292289',
         'text': '@SawyerMerritt Hardly anyone knows this'},
        {'id': '1539293206042578944',
         'text': '@alex_avoigt @WholeMarsBlog Probably only a few months'},
        {'id': '1539292625433501702',
         'text': '@business Twitter me &amp; real-life me are quite different haha!
       https://t.co/zedimZrthW'},
        {'id': '1539291794436915200',
         'text': '@BloombergLive @Twitter The vote of confidence is much
       appreciated'},
        {'id': '1539275446625476614', 'text': 'https://t.co/YhpHKcCYXz'},
        {'id': '1539033430876356609',
         'text': '@Degentraland Artificial Insemination?'},
        {'id': '1539031248638812160',
         'text': 'AI gets better every day https://t.co/Lz5XfXRJjh'},
        {'id': '1538998865495564291',
         'text': 'Some great suggestions in the comments!'},
        {'id': '1538992942983127040',
         'text': 'But sometimes they’re out of stock\nhttps://t.co/ybRiBp1kkP'},
        {'id': '1538992456083161097',
         'text': '@marenkahnert That was the largest wheel of cheese in the Beverly
       Hills Cheese Shop!'},
        {'id': '1538992101739855877',
         'text': '@teslaownersSV I love many cheeses, so hard to say that a particular
       one is best, but maybe Stilton'},
        {'id': '1538991118603411461',
         'text': 'The sheer variety of cheese is amazing'},
        {'id': '1538986794703781888', 'text': 'What … is your favorite cheese?'},
        {'id': '1538985909395279872', 'text': '@teslaownersSV cgi irl'},
        {'id': '1538951248187011072', 'text': '@Andst7 Trending to emptiness'},
        {'id': '1538946250409660418', 'text': '@waitbutwhy Could be a contributor'},
        {'id': '1538943339776684034',
         'text': '@teslaownersSV @SpaceX Super talented team at SpaceX'},
        {'id': '1538793660585857029', 'text': '@jmhorp @paulg Interesting'},
        {'id': '1538681719779360768', 'text': '@ilyasut Maybe we’re in a computer'},
        {'id': '1538605529332953089', 'text': '@waitbutwhy @BillyM2k  '},
        {'id': '1538406703150014466', 'text': '@AltcoinGordon I am'},
        {'id': '1538406040374595585', 'text': 'I will keep supporting Dogecoin'},
        {'id': '1538403887635390465', 'text': '  shadow crew  '},
        {'id': '1538401108271452161',
   'text': '@BillyM2k I feel swindled every time I drink one'},
  {'id': '1538400836669296640', 'text': '@EvaFoxU Gwynne is the best'},
  {'id': '1538400096865472512',
   'text': '@VladimirVargasM He did teach me a lot of engineering &amp; physics
while growing up (in an environment that was austere &amp; often bleak)'},
  {'id': '1538399223317676035', 'text': 'I love all my kids so much'},
  {'id': '1538398534499721218', 'text': '@zebulgar  '},
  {'id': '1538398098766151680',
   'text': '@Liv_Boeree Eventually, everything runs out of
time.\nhttps://t.co/j4ZHlahNXC'},
  {'id': '1538397546510426112', 'text': 'Happy Father’s Day'},
  {'id': '1538396007167365120',
   'text': 'If you can’t smell your wifi, how do you know it’s real?'},
  {'id': '1538394661483687936',
   'text': 'Congrats to SpaceX Falcon team for executing 3 flawless launches in
2 days! https://t.co/2MFmlkXmVz'},
  {'id': '1538388295570411520',
   'text': '@BillyM2k @CryptoWhale Good question'},
  {'id': '1538208757905297409',
   'text': 'To answer the question: Why Twitter?'},
  {'id': '1538205372892303360',
   'text': 'This will encourage people to change it haha'},
  {'id': '1538203601234976769', 'text': 'I’m pretty sure that unique'},
  {'id': '1538202890258591744',
   'text': 'We’re changing Starlink’s default wifi name to Stinky'},
  {'id': '1538201177824931848',
   'text': 'This is where \nthe writers are,\n\nOf past,\nPresent,\nAnd
Future.'},
  {'id': '1538200746054897664',
   'text': '@BillyM2k The only thing keeping the other orbital rocket programs
alive is government protection or they’d be deader than a doornail and everyone
knows it. \n\nBut oh well … comme ci, comme ça.'},
  {'id': '1538199752822738944',
   'text': '@BillyM2k The super weird thing is that Falcon 9 is still the only
orbital booster to land or refly after all these years!'},
  {'id': '1538197346093260802', 'text': '@Adam_4T @Google  '},
  {'id': '1538196236896960512',
   'text': 'Feels like déjà vu all over again haha https://t.co/ZokV7kPBV1'},
  {'id': '1538195371943403521',
   'text': '@Erdayastronaut @IzanRamos2002 @Caspar_Stanley Yes, about 20% more
thrust &amp; 20% less mass, but focus has been heavily on production rate &amp;
reliability.\n\nMass, thrust &amp; Isp will all improve, as will production
rate, reliability &amp; cost.\n\nThis is the only way to make life multi-
planetary and extend consciousness into the void.'},
  {'id': '1538146173332103169', 'text': '@WholeMarsBlog  '},
  {'id': '1538103995281055744', 'text': '@BillyM2k More currency-like'},
  {'id': '1538103933201260544',
  'text': '@BillyM2k Tesla and SpaceX merch, maybe more down the road'},
 {'id': '1538098006473555970', 'text': '@dennis_wilborn Rock on!'},
 {'id': '1538097755041763328',
  'text': '@TeslaAIBot @__SeriousGemini But we should have humans too!'},
 {'id': '1538097610980048896', 'text': '@nichegamer  '},
 {'id': '1537997699793833985',
  'text': 'Congratulations to Giga Berlin team on making over 1000 cars in a
week! https://t.co/TX8S4ozuxJ'},
 {'id': '1537997091800092679', 'text': '@EvaFoxU   '},
 {'id': '1537992573024755716',
  'text': '@blueskykites @Tesla @SpaceX @mayemusk @WholeMarsBlog @28delayslater
@JohnnaCrider1 @Kristennetten @SirineAti @GailAlfarATX @DimaZeniuk @bevedoni
@RationalEtienne @ashleevance @adamhoov @klwtts @RenataKonkoly For a couple of
months, but, yeah, that looks like the place. Does Mark still live there?'},
 {'id': '1537978581346709505',
  'text': '@EvaFoxU @BillyM2k I'm a dirty rocket'},
 {'id': '1537975779379662852',
  'text': 'It may as be a documentary, since it's coming true'},
 {'id': '1537970070307020808',
  'text': 'Watch the opening scene of Idiocracy. \n\nWhen I ask my friends why
they're not yet having kids (very few are), it sounds exactly like the
movie.\n\nhttps://t.co/528L1mhHi1'},
 {'id': '1537966358394073090',
  'text': '@BigImpactHumans My son (SJM) wanted to know why we couldn't have
20,000 cats. Perhaps this will make him reconsider.'},
 {'id': '1537964882074562561', 'text': 'Or perhaps social media in general'},
 {'id': '1537964094958948352', 'text': '@rabois Hmm …'},
 {'id': '1537962566638415872',
  'text': 'Is TikTok destroying civilization? Some people think so.'},
 {'id': '1537961702116847618', 'text': '@EvaFoxU Interesting'},
 {'id': '1537938252035870722', 'text': '@WholeMarsBlog Lame'},
 {'id': '1537906813797945347', 'text': 'https://t.co/oD5D5CVe5A'},
 {'id': '1537883083461959680', 'text': '@BillyM2k @Austen  '},
 {'id': '1537879670506389504', 'text': '@PPathole Yeah'},
 {'id': '1537873373711196161',
  'text': '@RenataKonkoly @SawyerMerritt So much free advertising!  '},
 {'id': '1537870974703607810',
  'text': '@SawyerMerritt Hyundai is doing pretty well'},
 {'id': '1537845711861071873', 'text': '@PPathole Exactly'},
 {'id': '1537845131147190273',
  'text': 'And rocket landings are now triple digits'},
 {'id': '1537844931351531521',
  'text': 'Our best landing video to date, thanks to Starlink!
https://t.co/kAjA3mxBta'},
 {'id': '1537840313733107719', 'text': '@teslaownersSV @UAW @GM @klwtts Yup'},
 {'id': '1537832293485621248', 'text': '@westcoastbill Yeah'},
 {'id': '1537831770833289216',
   'text': 'RT @SpaceX: Watch Falcon 9 launch 53 Starlink satellites to low-
Earth orbit → https://t.co/FHNtCC2xqJ  https://t.co/2GBQjLMNTt'},
  {'id': '1537831745130663937',
   'text': 'RT @SpaceX: Liftoff! https://t.co/28eNKniMqe'},
  {'id': '1537826398361903104', 'text': '@Model3Owners Ok'},
  {'id': '1537679369925279744', 'text': '@jespow @krakenfx Good thread'},
  {'id': '1537675348145590273',
   'text': '@DegreaseNeil That’s why Dragon has shields'},
  {'id': '1537672692035375107',
   'text': '@teslaownersSV @ID_AA_Carmack There will probably be several launch
countdowns before we pass all the abort triggers, but hopefully first countdown
is next month'},
  {'id': '1537669351062511616',
   'text': '@ID_AA_Carmack Yes, but it is achievable'},
  {'id': '1537649038274723840', 'text': '@__SeriousGemini @BillyM2k  '},
  {'id': '1537612145193541632',
   'text': '@BillyM2k @__SeriousGemini @bennyjohnson Accurate'},
  {'id': '1537583743006920704', 'text': '@EvaFoxU Exactly'},
  {'id': '1537582520501293056', 'text': '@Cernovich Wow'},
  {'id': '1537582037951213569', 'text': '@Andst7 Haha'},
  {'id': '1537580171888304129', 'text': '@bennyjohnson Interesting'},
  {'id': '1537531331453886465', 'text': '@lexfridman True'},
  {'id': '1537528613414875137',
   'text': '@GregScott_photo @FarryFaz @SpaceX @MarcusHouse @LabPadre
@13ericralph31 @spaceflashnews @spacex360 @SpaceIntellige3 @DJSnM
@Erdayastronaut @ScalesNews SpaceX team is making great progress at the Cape
&amp; Starbase!'},
  {'id': '1537527642013966338', 'text': '@sourpatchlyds @Twitter Exactly'},
  {'id': '1537517629035798533', 'text': '@BillyM2k @Rainmaker1973  '},
  {'id': '1537516259801169920',
   'text': '@alex_avoigt @Kaih042018 @klwtts True (sigh)'},
  {'id': '1537515240480448518',
   'text': '@dogeofficialceo @BillyM2k Also, In Sink band should let in a tiny
sink'},
  {'id': '1537513077238792193', 'text': '@ashleevance  '},
  {'id': '1537492318059696134', 'text': '@MatchasmMatt @EvaFoxU So it goes'},
  {'id': '1537492081823961090', 'text': '@EvaFoxU Good thread'},
  {'id': '1537490831262855168',
   'text': '@PPathole @paulg Awe no, I’m busted!'},
  {'id': '1537484570257391618',
   'text': '@joeydillon The Austin airport needs to be upgraded as fast as
possible'},
  {'id': '1537446415256854528', 'text': '@PPathole @teslaownersSV Have kids!'},
  {'id': '1537446007314608129', 'text': '@BillyM2k @dogeofficialceo  '},
  {'id': '1537445786954264578', 'text': '@BillyM2k Haha  '},
  {'id': '1537442531008339968',
   'text': '@teslaownersSV It’s a bigger risk than AI, so I’d put it at #1. If
```

```
these trends continue, humanity will cease to exist.'}],
 'meta': {'result_count': 100,
  'newest_id': '1539375908800368641',
  'oldest_id': '1537442531008339968',
  'next_token': '7140dibdnow9c7btw421tf85mdumry7lnkj413f9jt8ag'}}
```

The resulting JSON is then transformed into a list of tweets in the string format which will then be cleaned, processed and analyzed.

[74]:
```python
# Creating a list of tweets based in the json

tweets_list = []

for element in json_dataset['data']:
    tweets_list.append(element['text'])

print('List of tweets: ', len(tweets_list))
```

```
List of tweets:  100
```

0.3.2 2.2 Data processing

After retrieving the list of tweets, the notebook takes a few steps in order to prepare the data to be analyzed. Being them:

0.3.3 2.2.1 Dataset Tokenization

The first step of the pipeline is to tokenize each tweet in its own list of words in order to generate the first of the two basic data structures we will deal with during the whole process.

[75]:
```python
# Using NLTK to transform the list of tweets into tokens that can be later␣
 ↪analyzed and classified

from nltk.tokenize import word_tokenize, sent_tokenize

# Defining a function to use word_tokenize to tokenize elements in a dataset
def tokenize(list_data):
    tokens = []

    for element in list_data:
        words = word_tokenize(element)
        tokens.append(words)

    return tokens

tokens = tokenize(tweets_list)
print('List of tokens: ', len(tweets_list))
print('\n Showing the first 2 tweets as tokens: ', tokens[:2])
```

```
List of tokens:  100

 Showing the first 2 tweets as tokens:  [['@', 'thesheetztweetz', 'Their',
'attempt', 'to', 'bait', 'and', 'switch', 'satellite', 'spectrum', 'for',
'cellular', 'spectrum', 'is', 'super', 'shady', 'and', 'unethical', '.', 'If',
'they', 'are', 'successful', ',', 'it', 'would', 'hurt', 'the', 'least',
'served', 'and', 'completely', 'unserved', 'of', 'the', 'world', '.', 'Very',
'messed', 'up', '.'], ['@', 'SawyerMerritt', 'Hardly', 'anyone', 'knows',
'this']]
```

0.3.4 2.2.2 Removing unwanted data

After we have the list of tokens, we then use the stopwords corpora from NLTK in order to remove stopwords (such as 'and', 'or', 'it') as we do not want these common connection words to interfere in our analysis. We will also remove the punctuation so the only thing we have left is a clean list of words.

Removing the punctuation

[76]:
```python
# Removing the punctuation in order to clean the data

# Defining a function to remove punctuation from the wdataset
def remove_punctuation(data):
    tokens_with_no_punctuation = []
    for t in data:
        clean = [word for word in t if word.isalpha()]
        tokens_with_no_punctuation.append(clean)

    return tokens_with_no_punctuation

no_punctuation_tokens = remove_punctuation(tokens)
no_punctuation_tokens[:3][:3]
```

[76]:
```
[['thesheetztweetz',
  'Their',
  'attempt',
  'to',
  'bait',
  'and',
  'switch',
  'satellite',
  'spectrum',
  'for',
  'cellular',
  'spectrum',
  'is',
  'super',
  'shady',
  'and',
  'unethical',
  'If',
  'they',
  'are',
  'successful',
  'it',
  'would',
  'hurt',
  'the',
  'least',
  'served',
  'and',
  'completely',
  'unserved',
  'of',
  'the',
  'world',
  'Very',
  'messed',
  'up'],
 ['SawyerMerritt', 'Hardly', 'anyone', 'knows', 'this'],
 ['WholeMarsBlog', 'Probably', 'only', 'a', 'few', 'months']]
```

Removing the stopwords

[77]:
```python
# Removing the stopwords
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')

# Defining a function to remove stopwords based on a language
def remove_stopwords(data, language="portuguese"):
    stop_words = stopwords.words(language)

    tokens_without_stopwords = []
    for token in data:
        clean_token = [word for word in token if not word in stop_words]
        tokens_without_stopwords.append(clean_token)

    return tokens_without_stopwords

# Removing stopwords from the tokens
no_stopwords = remove_stopwords(no_punctuation_tokens, "portuguese")
no_stopwords_final = remove_stopwords(no_stopwords, "english")
```

```python
print('Tokens Length: ', len(no_stopwords_final))
print('\nTokens with no stopwords:', no_stopwords_final)
```

```
Tokens Length:  100

Tokens with no stopwords: [['thesheetztweetz', 'Their', 'attempt', 'bait',
'switch', 'satellite', 'spectrum', 'cellular', 'spectrum', 'super', 'shady',
'unethical', 'If', 'successful', 'would', 'hurt', 'least', 'served',
'completely', 'unserved', 'world', 'Very', 'messed'], ['SawyerMerritt',
'Hardly', 'anyone', 'knows'], ['WholeMarsBlog', 'Probably', 'months'],
['business', 'Twitter', 'amp', 'quite', 'different', 'haha', 'https'],
['BloombergLive', 'Twitter', 'The', 'vote', 'confidence', 'much',
'appreciated'], ['https'], ['Degentraland', 'Artificial', 'Insemination'],
['AI', 'gets', 'better', 'every', 'day', 'https'], ['Some', 'great',
'suggestions', 'comments'], ['But', 'sometimes', 'stock', 'https'],
['marenkahnert', 'That', 'largest', 'wheel', 'cheese', 'Beverly', 'Hills',
'Cheese', 'Shop'], ['teslaownersSV', 'I', 'love', 'many', 'cheeses', 'hard',
'say', 'particular', 'one', 'best', 'maybe', 'Stilton'], ['The', 'sheer',
'variety', 'cheese', 'amazing'], ['What', 'favorite', 'cheese'],
['teslaownersSV', 'cgi', 'irl'], ['Trending', 'emptiness'], ['waitbutwhy',
'Could', 'contributor'], ['teslaownersSV', 'SpaceX', 'Super', 'talented',
'team', 'SpaceX'], ['jmhorp', 'paulg', 'Interesting'], ['ilyasut', 'Maybe',
'computer'], ['waitbutwhy'], ['AltcoinGordon', 'I'], ['I', 'keep', 'supporting',
'Dogecoin'], ['shadow', 'crew'], ['I', 'feel', 'swindled', 'every', 'time', 'I',
'drink', 'one'], ['EvaFoxU', 'Gwynne', 'best'], ['VladimirVargasM', 'He',
'teach', 'lot', 'engineering', 'amp', 'physics', 'growing', 'environment',
'austere', 'amp', 'often', 'bleak'], ['I', 'love', 'kids', 'much'],
['zebulgar'], ['Eventually', 'everything', 'runs', 'time', 'https'], ['Happy',
'Father', 'Day'], ['If', 'smell', 'wifi', 'know', 'real'], ['Congrats',
'SpaceX', 'Falcon', 'team', 'executing', 'flawless', 'launches', 'days',
'https'], ['CryptoWhale', 'Good', 'question'], ['To', 'answer', 'question',
'Why', 'Twitter'], ['This', 'encourage', 'people', 'change', 'haha'], ['I',
'pretty', 'sure', 'unique'], ['We', 'changing', 'Starlink', 'default', 'wifi',
'name', 'Stinky'], ['This', 'writers', 'Of', 'past', 'Present', 'And',
'Future'], ['The', 'thing', 'keeping', 'orbital', 'rocket', 'programs', 'alive',
'government', 'protection', 'deader', 'doornail', 'everyone', 'knows', 'But',
'oh', 'well', 'comme', 'ci', 'comme', 'ça'], ['The', 'super', 'weird', 'thing',
'Falcon', 'still', 'orbital', 'booster', 'land', 'refly', 'years'], ['Google'],
['Feels', 'like', 'déjà', 'vu', 'haha', 'https'], ['Erdayastronaut', 'Yes',
'thrust', 'amp', 'less', 'mass', 'focus', 'heavily', 'production', 'rate',
'amp', 'reliability', 'Mass', 'thrust', 'amp', 'Isp', 'improve', 'production',
'rate', 'reliability', 'amp', 'cost', 'This', 'way', 'make', 'life', 'extend',
'consciousness', 'void'], ['WholeMarsBlog'], ['More'], ['Tesla', 'SpaceX',
'merch', 'maybe', 'road'], ['Rock'], ['TeslaAIBot', 'But', 'humans'],
['nichegamer'], ['Congratulations', 'Giga', 'Berlin', 'team', 'making', 'cars',
'week', 'https'], ['EvaFoxU'], ['blueskykites', 'Tesla', 'SpaceX', 'mayemusk',
'WholeMarsBlog', 'Kristennetten', 'SirineAti', 'GailAlfarATX', 'DimaZeniuk',
'bevedoni', 'RationalEtienne', 'ashleevance', 'adamhoov', 'klwtts',
'RenataKonkoly', 'For', 'couple', 'months', 'yeah', 'looks', 'like', 'place',
'Does', 'Mark', 'still', 'live'], ['EvaFoxU', 'I', 'dirty', 'rocket'], ['It',
'may', 'documentary', 'since', 'coming', 'true'], ['Watch', 'opening', 'scene',
'Idiocracy', 'When', 'I', 'ask', 'friends', 'yet', 'kids', 'sounds', 'exactly',
'like', 'movie', 'https'], ['BigImpactHumans', 'My', 'son', 'SJM', 'wanted',
'know', 'cats', 'Perhaps', 'make', 'reconsider'], ['Or', 'perhaps', 'social',
'media', 'general'], ['rabois', 'Hmm'], ['Is', 'TikTok', 'destroying',
'civilization', 'Some', 'people', 'think'], ['EvaFoxU', 'Interesting'],
['WholeMarsBlog', 'Lame'], ['https'], ['Austen'], ['PPathole', 'Yeah'],
['RenataKonkoly', 'SawyerMerritt', 'So', 'much', 'free', 'advertising'],
['SawyerMerritt', 'Hyundai', 'pretty', 'well'], ['PPathole', 'Exactly'], ['And',
'rocket', 'landings', 'triple', 'digits'], ['Our', 'best', 'landing', 'video',
'date', 'thanks', 'Starlink', 'https'], ['teslaownersSV', 'UAW', 'GM', 'klwtts',
'Yup'], ['westcoastbill', 'Yeah'], ['RT', 'SpaceX', 'Watch', 'Falcon', 'launch',
'Starlink', 'satellites', 'orbit', 'https', 'https'], ['RT', 'SpaceX',
'Liftoff', 'https'], ['Ok'], ['jespow', 'krakenfx', 'Good', 'thread'],
['DegreaseNeil', 'That', 'Dragon', 'shields'], ['teslaownersSV', 'There',
'probably', 'several', 'launch', 'countdowns', 'pass', 'abort', 'triggers',
'hopefully', 'first', 'countdown', 'next', 'month'], ['Yes', 'achievable'], [],
['bennyjohnson', 'Accurate'], ['EvaFoxU', 'Exactly'], ['Cernovich', 'Wow'],
['Haha'], ['bennyjohnson', 'Interesting'], ['lexfridman', 'True'], ['FarryFaz',
'SpaceX', 'MarcusHouse', 'LabPadre', 'spaceflashnews', 'DJSnM',
'Erdayastronaut', 'ScalesNews', 'SpaceX', 'team', 'making', 'great', 'progress',
'Cape', 'amp', 'Starbase'], ['sourpatchlyds', 'Twitter', 'Exactly'], [],
['klwtts', 'True', 'sigh'], ['dogeofficialceo', 'Also', 'In', 'Sink', 'band',
'let', 'tiny', 'sink'], ['ashleevance'], ['MatchasmMatt', 'EvaFoxU', 'So',
'goes'], ['EvaFoxU', 'Good', 'thread'], ['PPathole', 'paulg', 'Awe', 'I',
'busted'], ['joeydillon', 'The', 'Austin', 'airport', 'needs', 'upgraded',
'fast', 'possible'], ['PPathole', 'teslaownersSV', 'Have', 'kids'],
['dogeofficialceo'], ['Haha'], ['teslaownersSV', 'It', 'bigger', 'risk', 'AI',
'I', 'put', 'If', 'trends', 'continue', 'humanity', 'cease', 'exist']]
```

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

0.3.5 2.2.3 Stemming and Lemmatization

In order to have a better result for the Frequency Distribution graphs, NLTK's Stem library was applied using the Porter Stemmer algorithm to reduce the words to their roots. Porter Stemmer was chosen for this task due to the fact it is the less intrusive of the Stemming Algorithms. After stemming the list, WordNetLemmatizer was also used to lemmatize it as a second approach.

Stemming

[78]:
```python
# Stemming the tokens using PorterStemmer
from nltk.stem import PorterStemmer, LancasterStemmer

def stem_tokens(tokens):
    stemmed_tokens = []

    ps = PorterStemmer()

    for t in tokens:
        cleaned = [ps.stem(word) for word in t]
        stemmed_tokens.append(cleaned)

    return stemmed_tokens;

stemmed_tokens = stem_tokens(no_stopwords_final)

print('Tokens Length: ', len(stemmed_tokens))
print('\nStemmed Tokens:', stemmed_tokens)
```

```
Tokens Length:  100

Stemmed Tokens: [['thesheetztweetz', 'their', 'attempt', 'bait', 'switch',
'satellit', 'spectrum', 'cellular', 'spectrum', 'super', 'shadi', 'uneth', 'if',
'success', 'would', 'hurt', 'least', 'serv', 'complet', 'unserv', 'world',
'veri', 'mess'], ['sawyermerritt', 'hardli', 'anyon', 'know'], ['wholemarsblog',
'probabl', 'month'], ['busi', 'twitter', 'amp', 'quit', 'differ', 'haha',
'http'], ['bloombergl', 'twitter', 'the', 'vote', 'confid', 'much', 'appreci'],
['http'], ['degentraland', 'artifici', 'insemin'], ['ai', 'get', 'better',
'everi', 'day', 'http'], ['some', 'great', 'suggest', 'comment'], ['but',
'sometim', 'stock', 'http'], ['marenkahnert', 'that', 'largest', 'wheel',
'chees', 'beverli', 'hill', 'chees', 'shop'], ['teslaownerssv', 'i', 'love',
'mani', 'chees', 'hard', 'say', 'particular', 'one', 'best', 'mayb', 'stilton'],
['the', 'sheer', 'varieti', 'chees', 'amaz'], ['what', 'favorit', 'chees'],
['teslaownerssv', 'cgi', 'irl'], ['trend', 'empti'], ['waitbutwhi', 'could',
'contributor'], ['teslaownerssv', 'spacex', 'super', 'talent', 'team',
'spacex'], ['jmhorp', 'paulg', 'interest'], ['ilyasut', 'mayb', 'comput'],
['waitbutwhi'], ['altcoingordon', 'i'], ['i', 'keep', 'support', 'dogecoin'],
['shadow', 'crew'], ['i', 'feel', 'swindl', 'everi', 'time', 'i', 'drink',
'one'], ['evafoxu', 'gwynn', 'best'], ['vladimirvargasm', 'he', 'teach', 'lot',
'engin', 'amp', 'physic', 'grow', 'environ', 'auster', 'amp', 'often', 'bleak'],
['i', 'love', 'kid', 'much'], ['zebulgar'], ['eventu', 'everyth', 'run', 'time',
'http'], ['happi', 'father', 'day'], ['if', 'smell', 'wifi', 'know', 'real'],
['congrat', 'spacex', 'falcon', 'team', 'execut', 'flawless', 'launch', 'day',
'http'], ['cryptowhal', 'good', 'question'], ['to', 'answer', 'question', 'whi',
'twitter'], ['thi', 'encourag', 'peopl', 'chang', 'haha'], ['i', 'pretti',
'sure', 'uniqu'], ['we', 'chang', 'starlink', 'default', 'wifi', 'name',
'stinki'], ['thi', 'writer', 'of', 'past', 'present', 'and', 'futur'], ['the',
'thing', 'keep', 'orbit', 'rocket', 'program', 'aliv', 'govern', 'protect',
'deader', 'doornail', 'everyon', 'know', 'but', 'oh', 'well', 'comm', 'ci',
'comm', 'ça'], ['the', 'super', 'weird', 'thing', 'falcon', 'still', 'orbit',
'booster', 'land', 'refli', 'year'], ['googl'], ['feel', 'like', 'déjà', 'vu',
'haha', 'http'], ['erdayastronaut', 'ye', 'thrust', 'amp', 'less', 'mass',
'focu', 'heavili', 'product', 'rate', 'amp', 'reliabl', 'mass', 'thrust', 'amp',
'isp', 'improv', 'product', 'rate', 'reliabl', 'amp', 'cost', 'thi', 'way',
'make', 'life', 'extend', 'conscious', 'void'], ['wholemarsblog'], ['more'],
['tesla', 'spacex', 'merch', 'mayb', 'road'], ['rock'], ['teslaaibot', 'but',
'human'], ['nichegam'], ['congratul', 'giga', 'berlin', 'team', 'make', 'car',
'week', 'http'], ['evafoxu'], ['blueskykit', 'tesla', 'spacex', 'mayemusk',
'wholemarsblog', 'kristennetten', 'sirineati', 'gailalfaratx', 'dimazeniuk',
'bevedoni', 'rationaletienn', 'ashleev', 'adamhoov', 'klwtt', 'renatakonkoli',
'for', 'coupl', 'month', 'yeah', 'look', 'like', 'place', 'doe', 'mark',
'still', 'live'], ['evafoxu', 'i', 'dirti', 'rocket'], ['it', 'may',
'documentari', 'sinc', 'come', 'true'], ['watch', 'open', 'scene', 'idiocraci',
'when', 'i', 'ask', 'friend', 'yet', 'kid', 'sound', 'exactli', 'like', 'movi',
'http'], ['bigimpacthuman', 'my', 'son', 'sjm', 'want', 'know', 'cat', 'perhap',
'make', 'reconsid'], ['or', 'perhap', 'social', 'media', 'gener'], ['raboi',
'hmm'], ['is', 'tiktok', 'destroy', 'civil', 'some', 'peopl', 'think'],
['evafoxu', 'interest'], ['wholemarsblog', 'lame'], ['http'], ['austen'],
['ppathol', 'yeah'], ['renatakonkoli', 'sawyermerritt', 'so', 'much', 'free',
'advertis'], ['sawyermerritt', 'hyundai', 'pretti', 'well'], ['ppathol',
'exactli'], ['and', 'rocket', 'land', 'tripl', 'digit'], ['our', 'best', 'land',
'video', 'date', 'thank', 'starlink', 'http'], ['teslaownerssv', 'uaw', 'gm',
'klwtt', 'yup'], ['westcoastbil', 'yeah'], ['rt', 'spacex', 'watch', 'falcon',
'launch', 'starlink', 'satellit', 'orbit', 'http', 'http'], ['rt', 'spacex',
'liftoff', 'http'], ['ok'], ['jespow', 'krakenfx', 'good', 'thread'],
['degreaseneil', 'that', 'dragon', 'shield'], ['teslaownerssv', 'there',
'probabl', 'sever', 'launch', 'countdown', 'pass', 'abort', 'trigger', 'hope',
'first', 'countdown', 'next', 'month'], ['ye', 'achiev'], [], ['bennyjohnson',
'accur'], ['evafoxu', 'exactli'], ['cernovich', 'wow'], ['haha'],
['bennyjohnson', 'interest'], ['lexfridman', 'true'], ['farryfaz', 'spacex',
'marcushous', 'labpadr', 'spaceflashnew', 'djsnm', 'erdayastronaut',
'scalesnew', 'spacex', 'team', 'make', 'great', 'progress', 'cape', 'amp',
'starbas'], ['sourpatchlyd', 'twitter', 'exactli'], [], ['klwtt', 'true',
'sigh'], ['dogeofficialceo', 'also', 'in', 'sink', 'band', 'let', 'tini',
'sink'], ['ashleev'], ['matchasmmatt', 'evafoxu', 'so', 'goe'], ['evafoxu',
'good', 'thread'], ['ppathol', 'paulg', 'awe', 'i', 'bust'], ['joeydillon',
'the', 'austin', 'airport', 'need', 'upgrad', 'fast', 'possibl'], ['ppathol',
'teslaownerssv', 'have', 'kid'], ['dogeofficialceo'], ['haha'],
['teslaownerssv', 'it', 'bigger', 'risk', 'ai', 'i', 'put', 'if', 'trend',
'continu', 'human', 'ceas', 'exist']]
```

Lemmatization

[79]:
```python
# Lemmatization on top of the list of tweets
from nltk.stem import WordNetLemmatizer

def lemmatize_tokens(tokens):

    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = []

    for t in tokens:
        cleaned = [lemmatizer.lemmatize(word) for word in t]
        lemmatized_tokens.append(cleaned)

    return lemmatized_tokens

lemmatized_tokens = lemmatize_tokens(stemmed_tokens)
print('The first 10 tweets after lemmatization: ', lemmatized_tokens[:10])
```

```
The first 10 tweets after lemmatization:  [['thesheetztweetz', 'their',
'attempt', 'bait', 'switch', 'satellit', 'spectrum', 'cellular', 'spectrum',
'super', 'shadi', 'uneth', 'if', 'success', 'would', 'hurt', 'least', 'serv',
'complet', 'unserv', 'world', 'veri', 'mess'], ['sawyermerritt', 'hardli',
'anyon', 'know'], ['wholemarsblog', 'probabl', 'month'], ['busi', 'twitter',
'amp', 'quit', 'differ', 'haha', 'http'], ['bloombergl', 'twitter', 'the',
'vote', 'confid', 'much', 'appreci'], ['http'], ['degentraland', 'artifici',
'insemin'], ['ai', 'get', 'better', 'everi', 'day', 'http'], ['some', 'great',
'suggest', 'comment'], ['but', 'sometim', 'stock', 'http']]
```

0.3.6 2.2.4 Flattening

After our data is clean and concise, we then flatten the list of tweets in order to generate the second data structure we will deal with: a list containing all the words in the dataset.

[80]:
```
# Flattening the clean list

tokens_with_lem_stem = []
for word in lemmatized_tokens:
    tokens_with_lem_stem += word

print(tokens_with_lem_stem[:50])
print("\nNumber of tokens:", len(tokens_with_lem_stem))
```

```
['thesheetztweetz', 'their', 'attempt', 'bait', 'switch', 'satellit',
'spectrum', 'cellular', 'spectrum', 'super', 'shadi', 'uneth', 'if', 'success',
'would', 'hurt', 'least', 'serv', 'complet', 'unserv', 'world', 'veri', 'mess',
'sawyermerritt', 'hardli', 'anyon', 'know', 'wholemarsblog', 'probabl', 'month',
'busi', 'twitter', 'amp', 'quit', 'differ', 'haha', 'http', 'bloombergl',
'twitter', 'the', 'vote', 'confid', 'much', 'appreci', 'http', 'degentraland',
'artifici', 'insemin', 'ai', 'get']

Number of tokens: 525
```

0.3.7 2.3 Data limitations

Due to Twitter's API limitations, the maximum number of Tweets we can collect is 100. This of course limits the exploration we can do on this dataset as a higher number would generate better insights. There are a few workarounds on this limitations (such as exploring the API's Pagination tool) which will be considered for an extension on this research. As per now only 100 tweets can be processed, though.

0.4 3. Processing Output

0.4.1 3.1 Data Visualization

In order to better visualize the data we are dealing with, Wordcloud and Matplotlib were used to plot a word cloud view based on the words and the colors of Twitter. At this step it is already possible to gather a few insights on the content of the dataset, such as the words who appear the most in the user's twitters.

Two word clouds were plotted: * One based on the clean list of words * One based on the stemmed and lemmatized list of words

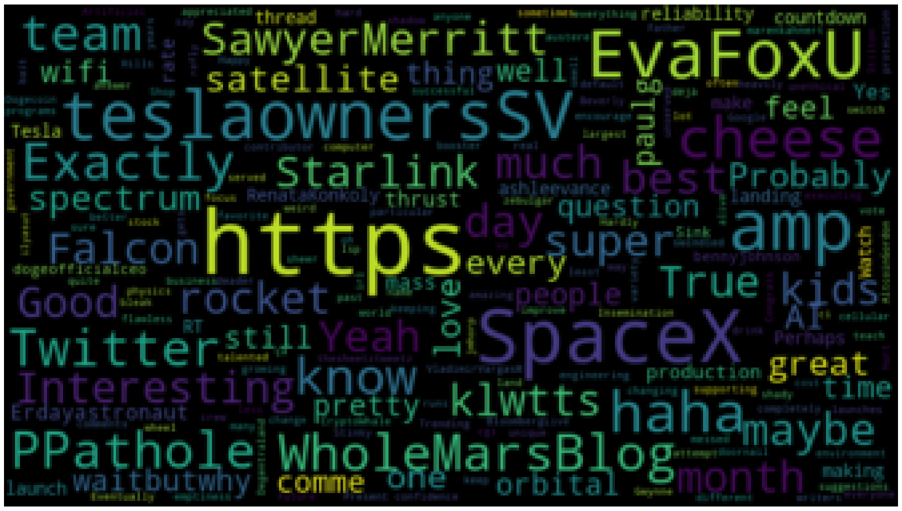
0.4.2 Creating the Word Cloud

[81]:
```
# Using WordCloud here to create a wordcloud for all the words in the dataset
# Using matplotlib to display the world cloud image
import matplotlib.pyplot as plt
import numpy as np
import os
import wordcloud
from os import path
from PIL import Image
from wordcloud import WordCloud, ImageColorGenerator

def create_wordcloud(dataset, wordcloud):
    # Get Image Path
    image_path = path.join(os.getcwd(), "twitter.jpeg")

    # Open the image using PIL.Image to generate the color mask
    color_mask = np.array(Image.open(image_path))

    wordcloud = WordCloud(background_color="black", max_font_size=70,␣
 ↪mask=color_mask).generate(dataset)
    plt.figure()
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.show()
```

[82]:
```
# Creating a wordcloud for the tokens before lemmatization and stemming so we␣
 ↪can see the complete words
tokens_without_lem_stem = []
```

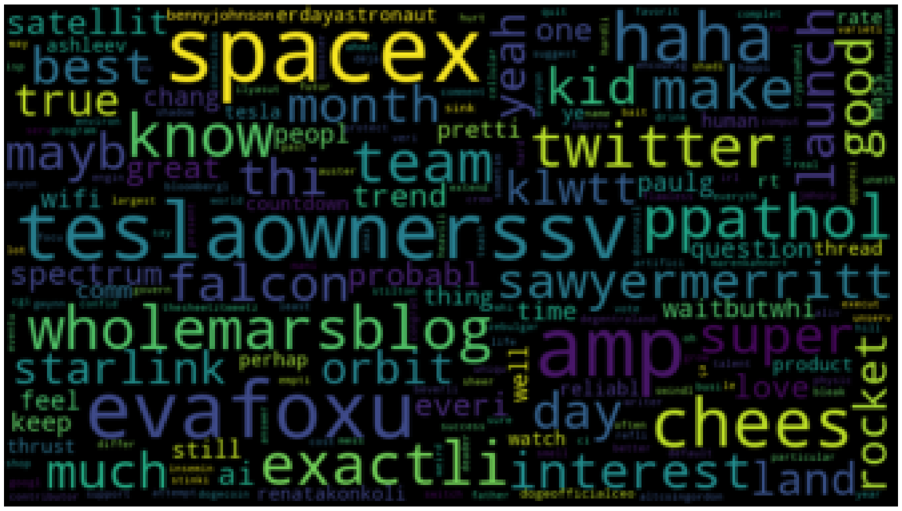
```
for word in no_stopwords_final:
    tokens_without_lem_stem += word

print('Word cloud for the tokens with no lemmatization and stemming:')
create_wordcloud(' '.join(tokens_without_lem_stem), wordcloud)
```

```
Word cloud for the tokens with no lemmatization and stemming:
```



```
[83]: # Creating a wordcloud for the tokens after lemmatization and stemming
print('Word cloud for the Stemmed & Lemmatized tokens:')
create_wordcloud(' '.join(tokens_with_lem_stem), wordcloud)
```

```
Word cloud for the Stemmed & Lemmatized tokens:
```



0.5 3.2 Frequency Distribution

To understand more about the words used most frequently by the User two Graphs were plotted: A distribution of Words over 100 tokens. The frequency of the top 10 most common words over the whole dataset.

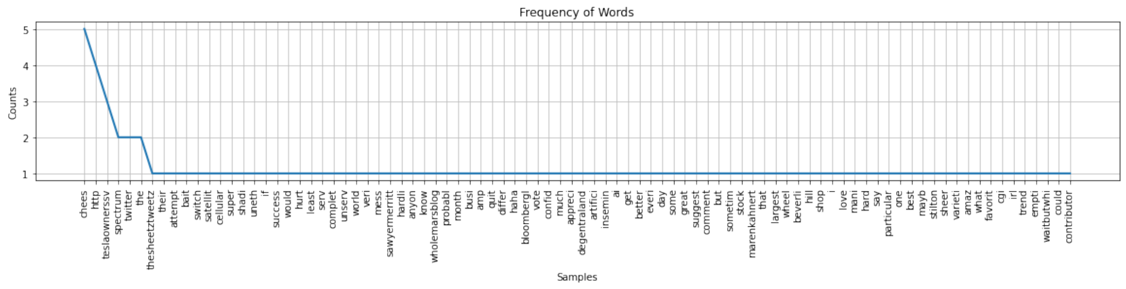
NLTK.freqDist was used in order to generate both these graphs.

0.5.1 Counting the frequency distribution

[84]:
```
# Defining a function to plot the frequency distribution on top of a list of␣
 ↪words
def plot_frequency_distribution(x, title=""):
    # Get freq distribution of tokens
    f = nltk.FreqDist(x)

    # Plot freq dist
    plt.figure(figsize=(20,3))

    plt.title(title)
    f.plot()
```

0.5.2 Distribution of words over 100 tokens

[85]:
```
# Frequency of words over 100 tokens
plot_frequency_distribution(tokens_with_lem_stem[:100], "Frequency of Words")
```



0.5.3 Frequency of the 10 most common words

[86]:
```
# Frequency of most common words over the whole token set
freq_dist_tokens = nltk.FreqDist(tokens_without_lem_stem)


freq_dist_tokens.most_common()
```

```
print('Frequency of 10 most common words in the whole token set:\n')
freq_dist_tokens.tabulate(10)
```

```
Frequency of 10 most common words in the whole token set:

        https               I         SpaceX            amp teslaownersSV
EvaFoxU             The WholeMarsBlog        Twitter            team
           14              11              9              8              7
7              5              4              4              4
```

0.6 3.2 Bigrams, Trigrams and Quadrigrams analysis

NLTK's collocations package was used in order to find the 20 most common words that were frequently used together for Bigrams, Trigrams and Quadrigrams. This analysis provides a more contextual analysis on the user's tweets. By looking at the results we can understand more about the subjects of interest of the User and a big picture view on the user's sentiment on these subjects.

0.6.1 20 Most Common Bigrams

```
[87]: bigrams = nltk.collocations.BigramCollocationFinder.
       ↪from_words(tokens_without_lem_stem)

      bigrams.ngram_fd.most_common(20)
```

```
[87]: [(('haha', 'https'), 2),
       (('I', 'love'), 2),
       (('thrust', 'amp'), 2),
       (('production', 'rate'), 2),
       (('Tesla', 'SpaceX'), 2),
       (('team', 'making'), 2),
       (('RT', 'SpaceX'), 2),
       (('Good', 'thread'), 2),
       (('thesheetztweetz', 'Their'), 1),
       (('Their', 'attempt'), 1),
       (('attempt', 'bait'), 1),
       (('bait', 'switch'), 1),
       (('switch', 'satellite'), 1),
       (('satellite', 'spectrum'), 1),
       (('spectrum', 'cellular'), 1),
       (('cellular', 'spectrum'), 1),
       (('spectrum', 'super'), 1),
       (('super', 'shady'), 1),
       (('shady', 'unethical'), 1),
       (('unethical', 'If'), 1)]
```

0.6.2 20 Most Common Trigrams

```
[88]: trigrams = nltk.collocations.TrigramCollocationFinder.
       ↪from_words(tokens_without_lem_stem)

      trigrams.ngram_fd.most_common(20)
```

```
[88]: [(('thesheetztweetz', 'Their', 'attempt'), 1),
       (('Their', 'attempt', 'bait'), 1),
       (('attempt', 'bait', 'switch'), 1),
       (('bait', 'switch', 'satellite'), 1),
       (('switch', 'satellite', 'spectrum'), 1),
       (('satellite', 'spectrum', 'cellular'), 1),
       (('spectrum', 'cellular', 'spectrum'), 1),
       (('cellular', 'spectrum', 'super'), 1),
       (('spectrum', 'super', 'shady'), 1),
       (('super', 'shady', 'unethical'), 1),
       (('shady', 'unethical', 'If'), 1),
       (('unethical', 'If', 'successful'), 1),
       (('If', 'successful', 'would'), 1),
       (('successful', 'would', 'hurt'), 1),
       (('would', 'hurt', 'least'), 1),
       (('hurt', 'least', 'served'), 1),
       (('least', 'served', 'completely'), 1),
       (('served', 'completely', 'unserved'), 1),
       (('completely', 'unserved', 'world'), 1),
       (('unserved', 'world', 'Very'), 1)]
```

0.6.3 20 Most Common Quadgrams

```
[89]: quadgrams = nltk.collocations.QuadgramCollocationFinder.
       ↪from_words(tokens_without_lem_stem)

      quadgrams.ngram_fd.most_common(20)
```

```
[89]: [(('thesheetztweetz', 'Their', 'attempt', 'bait'), 1),
       (('Their', 'attempt', 'bait', 'switch'), 1),
       (('attempt', 'bait', 'switch', 'satellite'), 1),
       (('bait', 'switch', 'satellite', 'spectrum'), 1),
       (('switch', 'satellite', 'spectrum', 'cellular'), 1),
       (('satellite', 'spectrum', 'cellular', 'spectrum'), 1),
       (('spectrum', 'cellular', 'spectrum', 'super'), 1),
       (('cellular', 'spectrum', 'super', 'shady'), 1),
       (('spectrum', 'super', 'shady', 'unethical'), 1),
       (('super', 'shady', 'unethical', 'If'), 1),
       (('shady', 'unethical', 'If', 'successful'), 1),
       (('unethical', 'If', 'successful', 'would'), 1),
       (('If', 'successful', 'would', 'hurt'), 1),
       (('successful', 'would', 'hurt', 'least'), 1),
       (('would', 'hurt', 'least', 'served'), 1),
       (('hurt', 'least', 'served', 'completely'), 1),
       (('least', 'served', 'completely', 'unserved'), 1),
       (('served', 'completely', 'unserved', 'world'), 1),
       (('completely', 'unserved', 'world', 'Very'), 1),
       (('unserved', 'world', 'Very', 'messed'), 1)]
```

0.7 3.3 Sentiment Analysis

To finish the analysis I used NLTK's Vader pre-trained lexicon to analyze the compound positiveness and negativeness index for each of the User's Tweets. A CSV file is then generated using containing three columns: The tweet, the compound score and the final result for the sentiment analysis.

0.8 Vader

[90]:
```
# Analyzing if tweet is positive or negative using VADER's Sentiment Analyzer
from random import shuffle
from nltk.sentiment import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')

sia = SentimentIntensityAnalyzer()

def get_polarity_score(tweet: str) -> bool:
    return sia.polarity_scores(tweet)


analysis_result_json = []
dict = {}
for tweet in tweets_list:

    sentiment = 'POSITIVE' if get_polarity_score(tweet)["compound"] > 0 else␣
 ↪'NEGATIVE'
    compound = get_polarity_score(tweet)["compound"]

    tweet_sentiment_dictionary = {
        "Tweet": tweet,
        "Compound Score": compound,
        "Sentiment": sentiment
    }
    analysis_result_json.append(tweet_sentiment_dictionary)

analysis_result_json
```

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

[90]:
```
[{'Tweet': '@thesheetztweetz Their attempt to bait and switch satellite spectrum
for cellular spectrum is super shady and unethical. \n\nIf they are successful,
it would hurt the least served and completely unserved of the world. Very messed
up.',
  'Compound Score': -0.1761,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@SawyerMerritt Hardly anyone knows this',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@alex_avoigt @WholeMarsBlog Probably only a few months',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@business Twitter me &amp; real-life me are quite different haha!
https://t.co/zedimZrthW',
  'Compound Score': 0.553,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@BloombergLive @Twitter The vote of confidence is much appreciated',
  'Compound Score': 0.765,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'https://t.co/YhpHKcCYXz',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@Degentraland Artificial Insemination?',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'AI gets better every day https://t.co/Lz5XfXRJjh',
  'Compound Score': 0.4404,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'Some great suggestions in the comments!',
  'Compound Score': 0.6588,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'But sometimes they’re out of stock\nhttps://t.co/ybRiBp1kkP',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@marenkahnert That was the largest wheel of cheese in the Beverly
Hills Cheese Shop!',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@teslaownersSV I love many cheeses, so hard to say that a particular
one is best, but maybe Stilton',
  'Compound Score': 0.5813,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'The sheer variety of cheese is amazing',
  'Compound Score': 0.5859,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'What … is your favorite cheese?',
  'Compound Score': 0.4588,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@teslaownersSV cgi irl',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@Andst7 Trending to emptiness',
  'Compound Score': -0.4404,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@waitbutwhy Could be a contributor',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@teslaownersSV @SpaceX Super talented team at SpaceX',
  'Compound Score': 0.802,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@jmhorp @paulg Interesting',
  'Compound Score': 0.4019,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@ilyasut Maybe we’re in a computer',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@waitbutwhy @BillyM2k  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@AltcoinGordon I am',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'I will keep supporting Dogecoin',
  'Compound Score': 0.4404,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '  shadow crew  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k I feel swindled every time I drink one',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@EvaFoxU Gwynne is the best',
  'Compound Score': 0.6369,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@VladimirVargasM He did teach me a lot of engineering &amp; physics
while growing up (in an environment that was austere &amp; often bleak)',
  'Compound Score': 0.1779,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'I love all my kids so much',
  'Compound Score': 0.6369,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@zebulgar  ', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': '@Liv_Boeree Eventually, everything runs out of
time.\nhttps://t.co/j4ZHlahNXC',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Happy Father’s Day',
  'Compound Score': 0.5719,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'If you can’t smell your wifi, how do you know it’s real?',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Congrats to SpaceX Falcon team for executing 3 flawless launches in
2 days! https://t.co/2MFmlkXmVz',
  'Compound Score': 0.7901,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@BillyM2k @CryptoWhale Good question',
  'Compound Score': 0.4404,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'To answer the question: Why Twitter?',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'This will encourage people to change it haha',
  'Compound Score': 0.743,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'I’m pretty sure that unique',
  'Compound Score': 0.6705,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'We’re changing Starlink’s default wifi name to Stinky',
  'Compound Score': -0.3612,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'This is where \nthe writers are,\n\nOf past,\nPresent,\nAnd
Future.',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k The only thing keeping the other orbital rocket programs
alive is government protection or they’d be deader than a doornail and everyone
knows it. \n\nBut oh well … comme ci, comme ça.',
  'Compound Score': 0.5346,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@BillyM2k The super weird thing is that Falcon 9 is still the only
orbital booster to land or refly after all these years!',
  'Compound Score': 0.5411,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@Adam_4T @Google  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Feels like déjà vu all over again haha https://t.co/ZokV7kPBV1',
  'Compound Score': 0.6705,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@Erdayastronaut @IzanRamos2002 @Caspar_Stanley Yes, about 20% more
thrust &amp; 20% less mass, but focus has been heavily on production rate &amp;
reliability.\n\nMass, thrust &amp; Isp will all improve, as will production
rate, reliability &amp; cost.\n\nThis is the only way to make life multi-
planetary and extend consciousness into the void.',
  'Compound Score': 0.775,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@WholeMarsBlog  ', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k More currency-like',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k Tesla and SpaceX merch, maybe more down the road',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@dennis_wilborn Rock on!',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@TeslaAIBot @__SeriousGemini But we should have humans too!',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@nichegamer  ', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Congratulations to Giga Berlin team on making over 1000 cars in a
week! https://t.co/TX8S4ozuxJ',
  'Compound Score': 0.636,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@EvaFoxU   ', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': '@blueskykites @Tesla @SpaceX @mayemusk @WholeMarsBlog @28delayslater
@JohnnaCrider1 @Kristennetten @SirineAti @GailAlfarATX @DimaZeniuk @bevedoni
@RationalEtienne @ashleevance @adamhoov @klwtts @RenataKonkoly For a couple of
months, but, yeah, that looks like the place. Does Mark still live there?',
  'Compound Score': 0.7227,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@EvaFoxU @BillyM2k I'm a dirty rocket',
  'Compound Score': -0.4404,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'It may as be a documentary, since it's coming true',
  'Compound Score': 0.4215,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'Watch the opening scene of Idiocracy. \n\nWhen I ask my friends why
they're not yet having kids (very few are), it sounds exactly like the
movie.\n\nhttps://t.co/528L1mhHi1',
  'Compound Score': 0.6808,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@BigImpactHumans My son (SJM) wanted to know why we couldn't have
```

```
20,000 cats. Perhaps this will make him reconsider.',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Or perhaps social media in general',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@rabois Hmm …', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Is TikTok destroying civilization? Some people think so.',
  'Compound Score': -0.5574,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@EvaFoxU Interesting',
  'Compound Score': 0.4019,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@WholeMarsBlog Lame',
  'Compound Score': -0.4215,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'https://t.co/oD5D5CVe5A',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k @Austen  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@PPathole Yeah', 'Compound Score': 0.296, 'Sentiment': 'POSITIVE'},
 {'Tweet': '@RenataKonkoly @SawyerMerritt So much free advertising!  ',
  'Compound Score': 0.5954,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@SawyerMerritt Hyundai is doing pretty well',
  'Compound Score': 0.6486,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@PPathole Exactly',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'And rocket landings are now triple digits',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'Our best landing video to date, thanks to Starlink!
https://t.co/kAjA3mxBta',
  'Compound Score': 0.8122,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@teslaownersSV @UAW @GM @klwtts Yup',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@westcoastbill Yeah',
  'Compound Score': 0.296,
  'Sentiment': 'POSITIVE'},
 {'Tweet': 'RT @SpaceX: Watch Falcon 9 launch 53 Starlink satellites to low-
Earth orbit → https://t.co/FHNtCC2xqJ  https://t.co/2GBQjLMNTt',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': 'RT @SpaceX: Liftoff! https://t.co/28eNKniMqe',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@Model3Owners Ok',
  'Compound Score': 0.296,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@jespow @krakenfx Good thread',
  'Compound Score': 0.4404,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@DegreaseNeil That's why Dragon has shields',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@teslaownersSV @ID_AA_Carmack There will probably be several launch
countdowns before we pass all the abort triggers, but hopefully first countdown
is next month',
  'Compound Score': 0.5499,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@ID_AA_Carmack Yes, but it is achievable',
  'Compound Score': 0.5859,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@__SeriousGemini @BillyM2k  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k @__SeriousGemini @bennyjohnson Accurate',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@EvaFoxU Exactly', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': '@Cernovich Wow',
  'Compound Score': 0.5859,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@Andst7 Haha', 'Compound Score': 0.4588, 'Sentiment': 'POSITIVE'},
 {'Tweet': '@bennyjohnson Interesting',
  'Compound Score': 0.4019,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@lexfridman True',
  'Compound Score': 0.4215,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@GregScott_photo @FarryFaz @SpaceX @MarcusHouse @LabPadre
@13ericralph31 @spaceflashnews @spacex360 @SpaceIntellige3 @DJSnM
@Erdayastronaut @ScalesNews SpaceX team is making great progress at the Cape
&amp; Starbase!',
  'Compound Score': 0.8016,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@sourpatchlyds @Twitter Exactly',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k @Rainmaker1973  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@alex_avoigt @Kaih042018 @klwtts True (sigh)',
  'Compound Score': 0.4215,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@dogeofficialceo @BillyM2k Also, In Sink band should let in a tiny
sink',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@ashleevance  ', 'Compound Score': 0.0, 'Sentiment': 'NEGATIVE'},
 {'Tweet': '@MatchasmMatt @EvaFoxU So it goes',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@EvaFoxU Good thread',
  'Compound Score': 0.4404,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@PPathole @paulg Awe no, I'm busted!',
  'Compound Score': -0.3595,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@joeydillon The Austin airport needs to be upgraded as fast as
possible',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@PPathole @teslaownersSV Have kids!',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k @dogeofficialceo  ',
  'Compound Score': 0.0,
  'Sentiment': 'NEGATIVE'},
 {'Tweet': '@BillyM2k Haha  ',
  'Compound Score': 0.4588,
  'Sentiment': 'POSITIVE'},
 {'Tweet': '@teslaownersSV It's a bigger risk than AI, so I'd put it at #1. If
these trends continue, humanity will cease to exist.',
  'Compound Score': -0.2732,
  'Sentiment': 'NEGATIVE'}]
```

[91]:
```
# Generating a csv file with the scores of the sentiment analysis

import numpy as np
import pandas as pd

# Generating a data frame table getting specific information
analysis_dataframe = pd.DataFrame(analysis_result_json, columns=["Tweet",␣
 ↪"Compound Score", "Sentiment"])
```

```
# Saving the CSV with all the content
analysis_dataframe.to_csv('sentiment_analysis_'+user_handle+'.csv')

analysis_dataframe
```

[91]:

```
                                                Tweet  Compound Score  \
0   @thesheetztweetz Their attempt to bait and swi…         -0.1761
1             @SawyerMerritt Hardly anyone knows this          0.0000
2   @alex_avoigt @WholeMarsBlog Probably only a fe…          0.0000
3   @business Twitter me &amp; real-life me are qu…          0.5530
4   @BloombergLive @Twitter The vote of confidence…          0.7650
..                                                …               …
95  @joeydillon The Austin airport needs to be upg…          0.0000
96                @PPathole @teslaownersSV Have kids!          0.0000
97                      @BillyM2k @dogeofficialceo          0.0000
98                                   @BillyM2k Haha          0.4588
99  @teslaownersSV It’s a bigger risk than AI, so …         -0.2732

   Sentiment
0   NEGATIVE
1   NEGATIVE
2   NEGATIVE
3   POSITIVE
4   POSITIVE
..       …
95  NEGATIVE
96  NEGATIVE
97  NEGATIVE
98  POSITIVE
99  NEGATIVE

[100 rows x 3 columns]
```

0.9 4. Next Steps

A few ideas are being considered here as next steps fo the this NLP Data Processing pipeline: Explore more than 100 tweets (break API limit) As mentioned previously, due to Twitter’s API Limitation we can only grab the latest 100 tweets. Historical searching is only allowed with a specific Researcher License available only for MSc. and PhD students. Twitter has, however, a Pagination tool which I will try to explore in order to improve the dataset.

1. Explore more than 100 tweets (break API limit) As mentioned previously, due to Twitter’s API Limitation we can only grab the latest 100 tweets. Historical searching is only allowed with a specific Researcher License available only for MSc. and PhD students. Twitter has, however, a Pagination tool which I will try to explore in order to improve the dataset.

2. Expand on NLP & Sentiment Analysis I only analyzed the positiveness and negativeness of tweets. I intend to do other possible analysis such as applying a Classification Algorithm to classify the tweets/words into categories. Or training my own Model instead of using a pre-trained one.

3. Add more information to the CSV with results During the analysis, a lot of insights were gathered, however only the Sentiment Analysis was contemplated in the final CSV file that was generated. I intend to put more columns with more insights in this file.

0.10 5. References and Resources

[1] Twitter's REST API Documentation: https://developer.twitter.com/en/docs/twitter-api/tweets/timelines/api-reference/get-users-id-tweets

[2] NLTK: https://www.nltk.org/

[3] Word Cloud: http://amueller.github.io/word_cloud

[4] MatplotLib: https://matplotlib.org/

[5] Pandas: https://pandas.pydata.org/

[6] Beri. A. Sentimental Analysis Using Vader: https://towardsdatascience.com/sentimental-analysis-using-vader. May 27, 2020

[7] Mogyorosi, Marius. Sentiment Analysis: First Steps With Python's NLTK Library: https://realpython.com/python-nltk-sentiment-analysis/

[8]Building a Classification Model with NLTK: https://www.nltk.org/api/nltk.classify.naivebayes.html https://www.nltk.org/api/nltk.classify.html

[9] Dadvar, M.J & Hauf, C. & De Jong. F. VADER: Scope of Negation Detection in Sentiment Analysis. University of Twente.

[10] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[11] NumPy: https://numpy.org/

```
[ ]:
```