For Text Mining assignment

ONE:

1) Perform sentimental analysis on the Elon-musk tweets (Exlon-musk.csv)

In [1]:
```
!pip install spacy
```

```
Requirement already satisfied: spacy in c:\users\admin\anaconda3\lib\site-packages (2.3.5)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (0.7.4)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (2.0.5)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (1.0.5)
Requirement already satisfied: numpy>=1.15.0 in c:\users\admin\anaconda3\lib\site-packages (from spacy) (1.
19.2)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (1.1.3)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\admin\anaconda3\lib\site-packages (fro
m spacy) (1.0.5)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in c:\users\admin\anaconda3\lib\site-packages (from
spacy) (1.0.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (4.50.2)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\admin\anaconda3\lib\site-packages (from sp
acy) (3.0.5)
Requirement already satisfied: thinc<7.5.0,>=7.4.1 in c:\users\admin\anaconda3\lib\site-packages (from spac
y) (7.4.5)
Requirement already satisfied: setuptools in c:\users\admin\anaconda3\lib\site-packages (from spacy) (50.3.
1.post20201107)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in c:\users\admin\anaconda3\lib\site-packages (from spa
cy) (0.8.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\admin\anaconda3\lib\site-packages (from
spacy) (2.24.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\admin\anaconda3\lib\site
-packages (from requests<3.0.0,>=2.13.0->spacy) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\admin\anaconda3\lib\site-packages (from reques
ts<3.0.0,>=2.13.0->spacy) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests<3.
0.0,>=2.13.0->spacy) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from reque
sts<3.0.0,>=2.13.0->spacy) (2020.6.20)
```

In [3]:
```
!pip install WordCloud
```

```
Collecting WordCloud
  Downloading wordcloud-1.8.1-cp38-cp38-win_amd64.whl (155 kB)
Requirement already satisfied: matplotlib in c:\users\admin\anaconda3\lib\site-packages (from WordCloud)
(3.3.2)
Requirement already satisfied: numpy>=1.6.1 in c:\users\admin\anaconda3\lib\site-packages (from WordCloud)
(1.19.2)
Requirement already satisfied: pillow in c:\users\admin\anaconda3\lib\site-packages (from WordCloud) (8.0.
1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplo
tlib->WordCloud) (1.3.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\admin\anaconda3\lib\sit
e-packages (from matplotlib->WordCloud) (2.4.7)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\admin\anaconda3\lib\site-packages (from matp
lotlib->WordCloud) (2020.6.20)
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib-
>WordCloud) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\admin\anaconda3\lib\site-packages (from mat
plotlib->WordCloud) (2.8.1)
Requirement already satisfied: six in c:\users\admin\anaconda3\lib\site-packages (from cycler>=0.10->matplo
tlib->WordCloud) (1.15.0)
Installing collected packages: WordCloud
Successfully installed WordCloud-1.8.1
```

In [32]:
```
!pip install google-colab
```

```
copying pandas\tests\sparse\test_indexing.py -> build\lib.win-amd64-3.8\pandas\tests\sparse
copying pandas\tests\sparse\test_pivot.py -> build\lib.win-amd64-3.8\pandas\tests\sparse
copying pandas\tests\sparse\test_reshape.py -> build\lib.win-amd64-3.8\pandas\tests\sparse
copying pandas\tests\sparse\__init__.py -> build\lib.win-amd64-3.8\pandas\tests\sparse
creating build\lib.win-amd64-3.8\pandas\tests\tools
copying pandas\tests\tools\test_numeric.py -> build\lib.win-amd64-3.8\pandas\tests\tools
copying pandas\tests\tools\__init__.py -> build\lib.win-amd64-3.8\pandas\tests\tools
creating build\lib.win-amd64-3.8\pandas\tests\tseries
copying pandas\tests\tseries\test_frequencies.py -> build\lib.win-amd64-3.8\pandas\tests\tseries
copying pandas\tests\tseries\test_holiday.py -> build\lib.win-amd64-3.8\pandas\tests\tseries
copying pandas\tests\tseries\__init__.py -> build\lib.win-amd64-3.8\pandas\tests\tseries
creating build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_api.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_array_to_datetime.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_ccalendar.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_conversion.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_libfrequencies.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_liboffsets.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_normalize_date.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
copying pandas\tests\tslibs\test_parse_iso8601.py -> build\lib.win-amd64-3.8\pandas\tests\tslibs
```

In [10]:
```
# importing libraries
import pandas as pd
import numpy as np
import string

import matplotlib.pyplot as plt
from matplotlib.pyplot import imread
from wordcloud import WordCloud

import spacy
import nltk

import re
from textblob import TextBlob

%matplotlib inline
```

In [3]:
```
tweets_data = pd.read_csv('Elon_musk.csv', encoding = 'latin1', error_bad_lines=False)
tweets_data.head()
```

Out[3]:

	Unnamed: 0	Text
0	1	@kunalb11 I m an alien
1	2	@ID_AA_Carmack Ray tracing on Cyberpunk with H...
2	3	@joerogan @Spotify Great interview!
3	4	@gtera27 Doge is underestimated
4	5	@teslacn Congratulations Tesla China for amazi...

In [4]:
```
tweets_data.shape
```

Out[4]: (1999, 2)

In [5]:
```
tweets_data = tweets_data.drop('Unnamed: 0', axis = 1)
tweets_data.head()
```

Out[5]:

	Text
0	@kunalb11 I m an alien
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...
2	@joerogan @Spotify Great interview!
3	@gtera27 Doge is underestimated
4	@teslacn Congratulations Tesla China for amazi...

In [8]:
```
# we need to remove a lot of junk(urls, tags, RT, Hashtags) from the tweets
# so we will use regular exlpression for that
def clean_tweets(tweets):
    tweets = re.sub('@[A-Za-z0-9]+', '', tweets) #Removing tag(@)
    tweets = re.sub('#', '', tweets) # Removing hashtag(#)
    tweets = re.sub('RT[\s]+', '', tweets) # Removing RT
    tweets = re.sub('https?:\/\/\S+', '', tweets) # Removing links
    return tweets

tweets_data.Text = tweets_data.Text.apply(clean_tweets)
tweets_data.head()
```

Out[8]:

	Text
0	11 I m an alien
1	_AA_Carmack Ray tracing on Cyberpunk with HDR ...
2	Great interview!
3	27 Doge is underestimated
4	Congratulations Tesla China for amazing execu...

Calculating Subjectivity and polarity

Subjectivity is nothing but a sentence that expresses some personal feelings, views, or beliefs. Its values range from 0 to 1 where 0 is very objective and 1 is very subjective,

while polarity simply means emotions expressed in a sentence. Its value ranges from -1 to 1, where -1 represents the most negative comment and 1 represent the most positive comment

In [11]:
```
# let's calculate subjectivity and Polarity
# function for subjectivity
def calc_subj(tweet):
    return TextBlob(tweet).sentiment.subjectivity

# function for Polarity
def calc_pola(tweet):
    return TextBlob(tweet).sentiment.polarity

tweets_data['Subjectivity'] = tweets_data.Text.apply(calc_subj)
tweets_data['Polarity'] = tweets_data.Text.apply(calc_pola)
tweets_data.head()
```

Out[11]:

	Text	Subjectivity	Polarity
0	11 I m an alien	0.750000	-0.250000
1	_AA_Carmack Ray tracing on Cyberpunk with HDR ...	0.000000	0.000000
2	Great interview!	0.750000	1.000000
3	27 Doge is underestimated	0.000000	0.000000
4	Congratulations Tesla China for amazing execu...	0.366667	0.345313

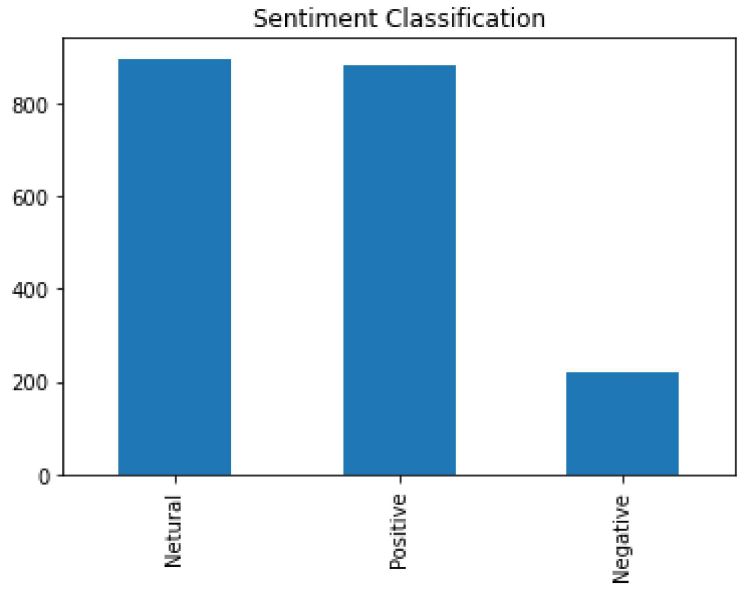
In [12]:
```
# now let's classify these tweets based on their sentiment(polarity)
def sentiment(polarity):
    result = ''
    if polarity > 0:
        result = 'Positive'
    elif polarity == 0:
        result = 'Netural'
    else:
        result = 'Negative'
    return result

tweets_data['Sentiment'] = tweets_data.Polarity.apply(sentiment)
tweets_data.head()
```

Out[12]:

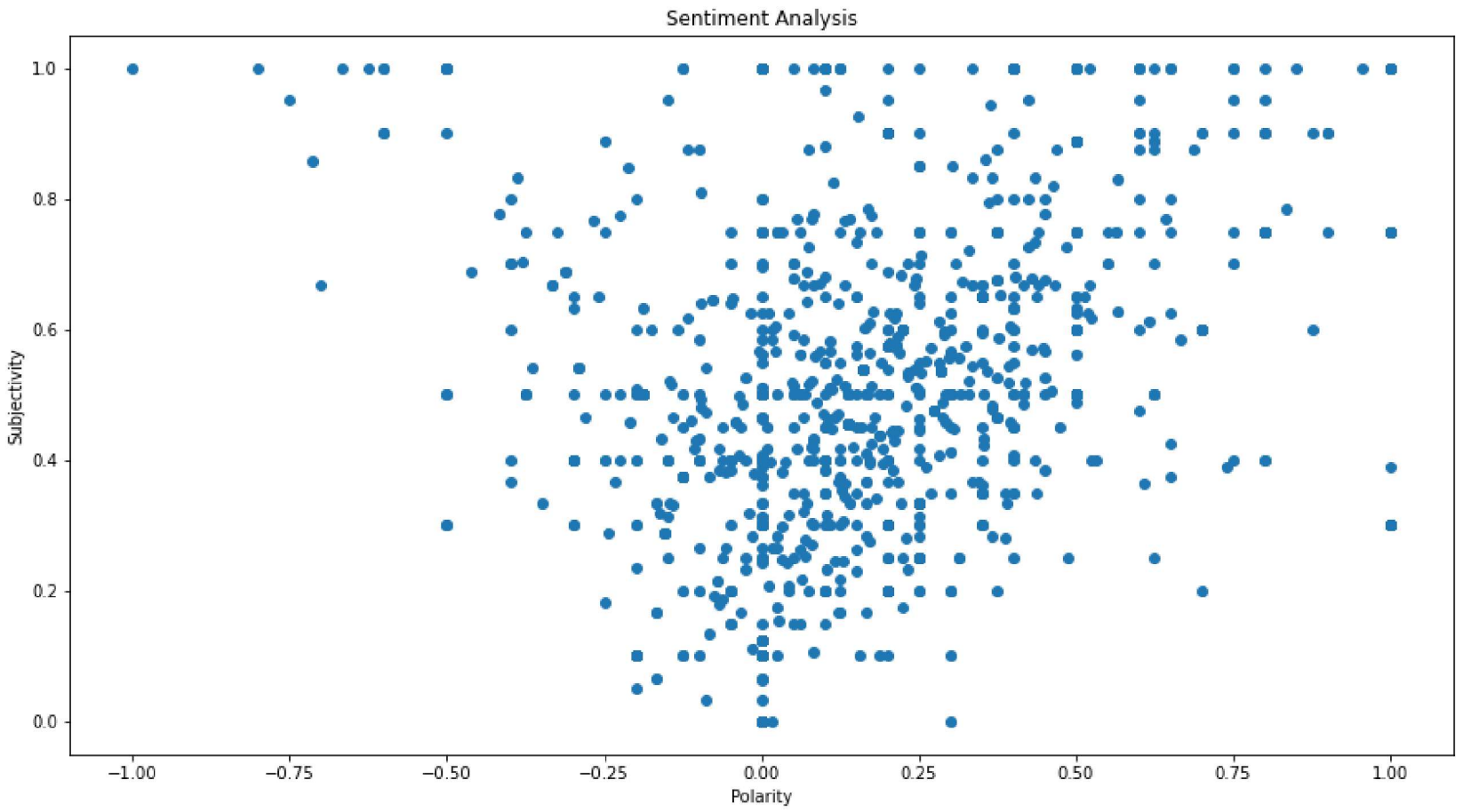
	Text	Subjectivity	Polarity	Sentiment
0	11 I m an alien	0.750000	-0.250000	Negative
1	_AA_Carmack Ray tracing on Cyberpunk with HDR ...	0.000000	0.000000	Netural
2	Great interview!	0.750000	1.000000	Positive
3	27 Doge is underestimated	0.000000	0.000000	Netural
4	Congratulations Tesla China for amazing execu...	0.366667	0.345313	Positive

In [16]:
```
tweets_data.Sentiment.value_counts().plot(kind='bar')
plt.title('Sentiment Classification')
plt.show()
```



In [18]:
```
f, axes = plt.subplots(figsize = (15,8))
plt.scatter(tweets_data.Polarity, tweets_data.Subjectivity)
plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
```

Out[18]: Text(0, 0.5, 'Subjectivity')



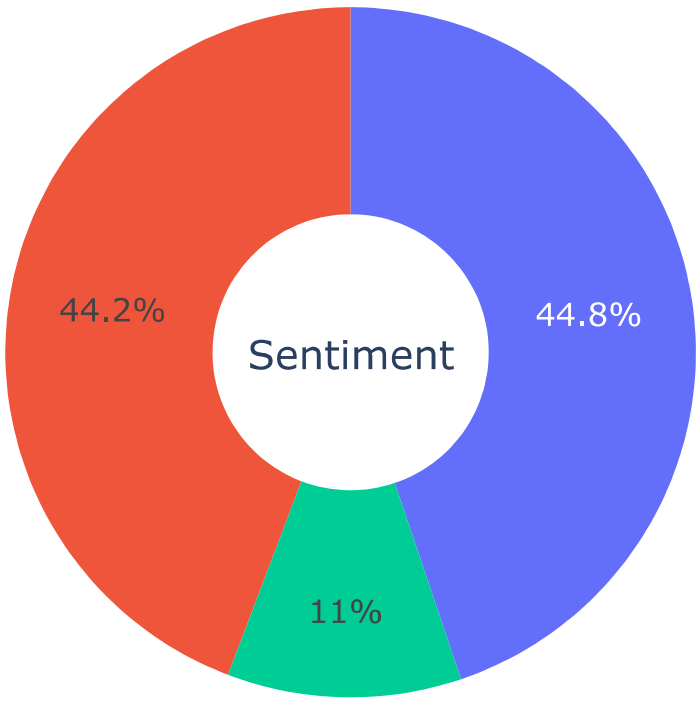
In [24]:
```
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings("ignore")


type_ = ["Positive", "Neutral", "Negative"]
fig = make_subplots(rows=1, cols=1)

fig.add_trace(go.Pie(labels=type_, values=tweets_data['Sentiment'].value_counts(), name="Sentiment"))

# Use `hole` to create a donut-like pie chart
fig.update_traces(hole=.4, hoverinfo="label+percent+name", textfont_size=16)

fig.update_layout(
    title_text="Sentiment Analysis",
    # Add annotations in the center of the donut pies.
    annotations=[dict(text='Sentiment', x=0.5, y=0.5, font_size=20, showarrow=False)])
fig.show()
```



Most used words by Elon musk

In [27]:
```
# setting up stop words
nltk.download('stopwords')
stpwrd = set(nltk.corpus.stopwords.words('english'))

# Combining all tweets text
allWords = ' '.join([twts for twts in tweets_data['Text']])
#allWords
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

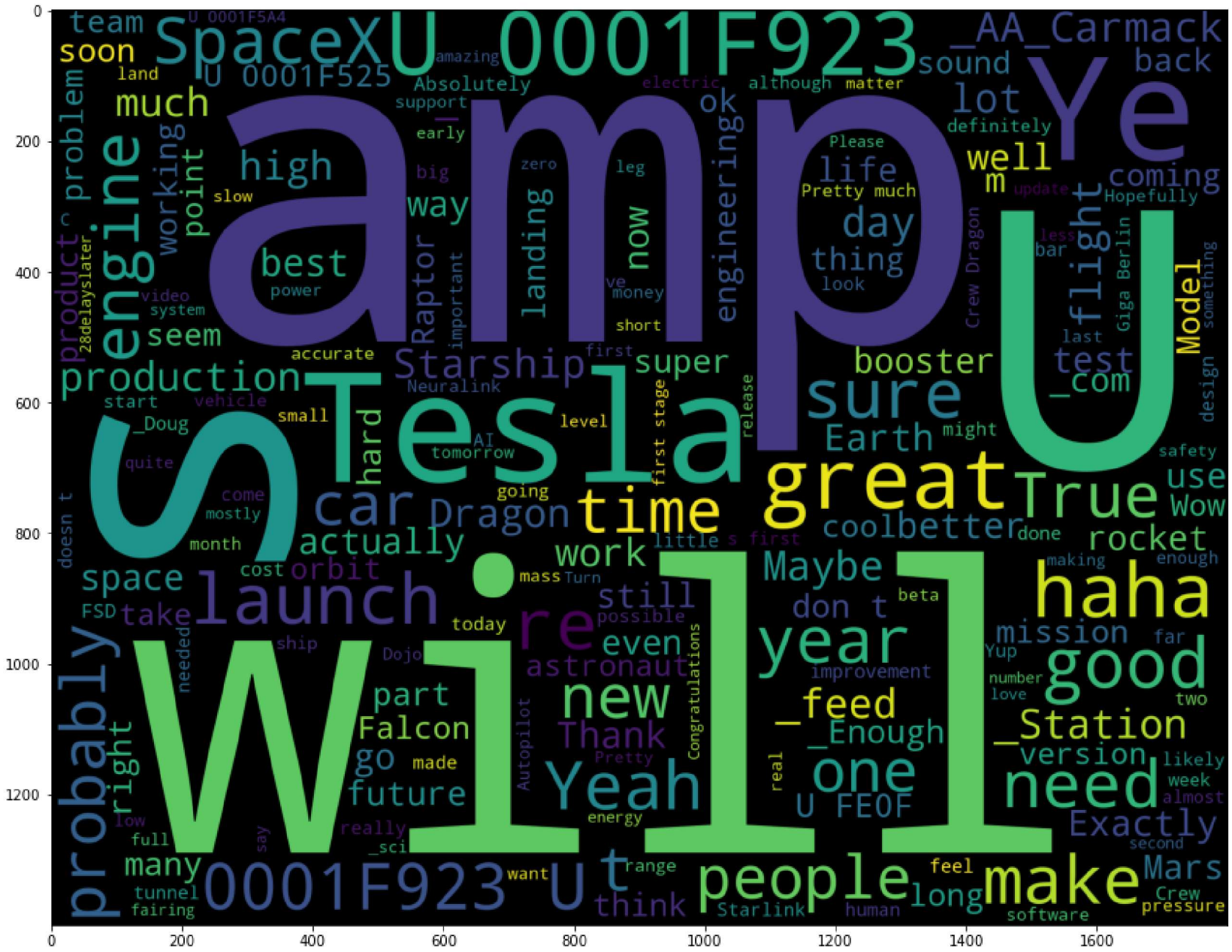
In [35]:
```
# setting up stop words
nltk.download('stopwords')  # run this if you get any error
stpwrd = set(nltk.corpus.stopwords.words('english'))

# Combining all tweets text
allWords = ' '.join([twts for twts in tweets_data['Text']])

f, axes = plt.subplots(figsize=(20,12))
wordcloud= WordCloud(
        background_color = 'black',
        width = 1800,
        height =1400).generate(allWords)
plt.imshow(wordcloud)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[35]: <matplotlib.figure.AxesImage at 0x26666c89760>



In [36]:
```
with open("positive-words.txt","r") as pos:
    poswords = pos.read().split("\n")

with open("negative-words.txt","r") as neg:
    negwords = neg.read().split("\n")


pos_words = poswords[35:]
pos_words
```

Out[36]:
```
['a+',
 'abound',
 'abounds',
 'abundance',
 'abundant',
 'accessable',
 'accessible',
 'acclaim',
 'acclaimed',
 'acclamation',
 'accolade',
 'accolades',
 'accommodative',
 'accomodative',
 'accomplish',
 'accomplished',
 'accomplishment',
 'accomplishments',
 'accurate',
```

In [37]:
```
neg_words = negwords[35:]
neg_words
```

Out[37]:
```
['2-faced',
 '2-faces',
 'abnormal',
 'abolish',
 'abominable',
 'abominably',
 'abominate',
 'abomination',
 'abort',
 'aborted',
 'aborts',
 'abrade',
 'abrasive',
 'abrupt',
 'abruptly',
 'abscond',
 'absence',
 'absent-minded',
 'absentee',
```

In [55]:
```
sentences = tweets_data["Text"].to_list()
```

In [56]:
```
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
wordnet = WordNetLemmatizer()
import re

filtered_sent=[]
for i in range(len(sentences)):
    review = re.sub("[^A-Za-z" "]+"," ",sentences[i])
    review = re.sub("[0-9" "]+"," ",sentences[i])

    review = review.lower()
    review =review.split()
    review = [wordnet.lemmatize(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    filtered_sent.append(review)
```

In [57]:
```
filtered_sent[0:5]
```

Out[57]:
```
['i\x92m alien',
 '_aa_carmack ray tracing cyberpunk hdr next-level. tried it?',
 'great interview!',
 'doge underestimated',
 'congratulation tesla china amazing execution last year. next even more!!']
```

In [58]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tf = TfidfVectorizer()
text_tf = tf.fit_transform(filtered_sent)
feature_names = tf.get_feature_names()
dense = text_tf.todense()
denselist = dense.tolist()
sentences_df =pd.DataFrame(denselist, columns=feature_names)
sentences_df.head()
```

Out[58]:

	__	__mania	__surfer	_aa_carmack	_addicted	_adri	_afshari	_ai	_alarms	_anne	...	yourself	yup	zealand	zenit	zero	ze
0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.259768	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	

5 rows × 3682 columns

In [59]:
```
f, axes = plt.subplots(figsize=(20,12))
pos_words = ' '.join([w for w in sentences_df if w in poswords])

cloud_pos = WordCloud(
        background_color = 'black',
        width =1800,
        height=1400).generate(pos_words)
plt.imshow(cloud_pos)
```

Out[59]: <matplotlib.image.AxesImage at 0x266674dc9a0>

In [61]:
```
f, axes = plt.subplots(figsize=(20,12))
neg_words = ' '.join([w for w in sentences_df if w in negwords])

cloud_neg = WordCloud(
        background_color='black',
        width =1800,
        height =1400).generate(neg_words)
plt.imshow(cloud_neg)
```

Out[61]: <matplotlib.image.AxesImage at 0x26666c20d00>