In [137]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import spacy
import string
import nltk
import re

from matplotlib import pyplot as plt
%matplotlib inline
from wordcloud import WordCloud,STOPWORDS
from nltk.tokenize import TweetTokenizer,word_tokenize
from nltk.corpus import stopwords
import warnings
warnings.filterwarnings('ignore')
```

In [138]:
```
data = pd.read_csv('Elon_musk.csv',encoding = 'Latin-1')
data
```

Out[138]:

	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...
...	...	...
1994	1995	@flcnhvy True, it sounds so surreal, but the n...
1995	1996	@PPathole Make sure to read ur terms & con...
1996	1997	@TeslaGong @PPathole Samwise Gamgee
1997	1998	@PPathole Altho Dumb and Dumber is <U+0001F525...
1998	1999	Progress update August 28

1999 rows × 2 columns

In [139]:
```
data.drop(['Unnamed: 0'],inplace = True,axis=1)
```

In [140]:
```
data = [Text.strip() for Text in data.Text]
data =[Text for Text in data if Text]
data[0:10]
```

Out[140]:
```
['@kunalb11 I\x92m an alien',
 '@ID_AA_Carmack Ray tracing on Cyberpunk with HDR is next-level. Have you trie
d it?',
 '@joerogan @Spotify Great interview!',
 '@gtera27 Doge is underestimated',
 '@teslacn Congratulations Tesla China for amazing execution last year. Now on
to the next for even more!!',
 'Happy New Year of the Ox! https://t.co/9WFKMYu2oj', (https://t.co/9WFKMYu2o
j',)
 'Frodo was the underdoge,\nAll thought he would fail,\nHimself most of all. ht
tps://t.co/zGxJFDzzrM', (https://t.co/zGxJFDzzrM',)
 '@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)',
 '@flcnhvy @anonyx10 Indeed! Tweets definitely do not represent real-world time
allocation.',
 'The most entertaining outcome is the most likely']
```

In [141]:
```
text_data = ' '.join(data)
text_data
```

```
s://t.co/3rWE9uHSTS) @geoffkeighley @UnrealEngine It\x92s getting real Bought
some Dogecoin for lil X, so he can be a toddler hodler @JoshManMode He defini
tely has issues, but the sentencing seems a bit high @freewalletorg Thanks fo
r fixing @freewalletorg Please unlock my account @AstroJordy <U+0001F923><U+0
001F923> This is true power haha https://t.co/Fc9uhQSd7O (https://t.co/Fc9uhQ
Sd7O) @freewalletorg Any crypto wallet that won\x92t give you your private ke
ys should be avoided at all costs @freewalletorg Your app sucks RT @SpaceX: N
ASA has selected Falcon Heavy to launch the first two elements of the lunar G
ateway together on one mission! https://t.co/3pWt (https://t.co/3pWt) @ajtour
ville Yes @BLKMDL3 @RationalEtienne @Adamklotz_ Once we can predict cash flow
reasonably well, Starlink will IPO @RationalEtienne @Adamklotz_ Starlink is a
staggeringly difficult technical &amp;amp; economic endeavor. However, if we don
\x85 https://t.co/9Z8Ac6skqx (https://t.co/9Z8Ac6skqx) @RationalEtienne @Adam
klotz_ SpaceX needs to pass through a deep chasm of negative cash flow over t
he next year or\x85 https://t.co/7J1c92hdjl (https://t.co/7J1c92hdjl) @ID_AA_
Carmack Lowest cost per ton of carbon sequestered, net of value of any produc
t made. \n\nMust be scalable to g\x85 https://t.co/XMyI7qWSgw (https://t.co/X
MyI7qWSgw) @Adamklotz_ It\x92s meant to be the same price in all countries. O
nly difference should be taxes &amp;amp; shipping. @tobyliiiiiiiiii This is inten
```

In [142]:
```
tknzr = TweetTokenizer(strip_handles=True)
data_tokens=tknzr.tokenize(text_data)
print(data_tokens)
```

```
o', 'launch', 'the', 'first', 'two', 'elements', 'of', 'the', 'lunar', 'Gatew
ay', 'together', 'on', 'one', 'mission', '!', 'https://t.co/3pWt', 'Yes', 'On
ce', 'we', 'can', 'predict', 'cash', 'flow', 'reasonably', 'well', ',', 'Star
link', 'will', 'IPO', 'Starlink', 'is', 'a', 'staggeringly', 'difficult', 'te
chnical', '&', 'economic', 'endeavor', '.', 'However', ',', 'if', 'we', 'do
n', 'https://t.co/9Z8Ac6skqx', 'SpaceX', 'needs', 'to', 'pass', 'through',
'a', 'deep', 'chasm', 'of', 'negative', 'cash', 'flow', 'over', 'the', 'nex
t', 'year', 'or', 'https://t.co/7J1c92hdjl', 'Lowest', 'cost', 'per', 'ton',
'of', 'carbon', 'sequestered', ',', 'net', 'of', 'value', 'of', 'any', 'produ
ct', 'made', '.', 'Must', 'be', 'scalable', 'to', 'g', 'https://t.co/XMyI7qWS
gw', 'It', '\x92', 's', 'meant', 'to', 'be', 'the', 'same', 'price', 'in', 'a
ll', 'countries', '.', 'Only', 'difference', 'should', 'be', 'taxes', '&', 's
hipping', '.', 'This', 'is', 'intended', 'for', 'Earth', ',', 'but', 'there',
'may', 'be', 'some', 'ideas', 'that', 'apply', 'to', 'Mars', 'too', '<U+0001F
923>', '<U+0001F923>', 'XPrize', 'team', 'will', 'manage', 'the', '$', '100
M', 'carbon', 'capture', 'prize', 'https://t.co/fSw5IanL0r', 'Everyone', 'a
t', 'Tesla', 'receives', 'stock', '.', 'My', 'comp', 'is', 'all', 'stock',
'/', 'options', ',', 'which', 'I', 'do', 'not', 'take', 'off', 'the', 'tabl
e', '.', 'That', '\x92', 's', 'what', 'you', '\x92', 're', 'missing', '.', 'B
```

In [143]:
```
tokens_text =' '.join(data_tokens)
tokens_text
```

```
& booster mass Back to work tonight ! Ð is for Dogecoin ! Instructional vide
o . https://t.co/UEEocOfcTb (https://t.co/UEEocOfcTb) The people have spoken
https://t.co/x41oVMzTGo (https://t.co/x41oVMzTGo) So cute <U+0001F495> Extrem
ely misleading image , as doesn \x92 t reflect true time cost to people or ra
in & pain <U+0001F3B6> Who let the Doge out <U+0001F3B6> Hodl the rainforests
! ! So it \x92 s finally come to this https://t.co/Gf0Rg2QOaF (https://t.co/G
f0Rg2QOaF) It \x92 s the most fun crypto ! Its simplicity is its genius Yup <
U+0001F923> <U+0001F923> True <U+0001F923> <U+0001F923> Not that easy . This
 is two decades of intense work . Have to look at old notes , emails , texts
 . Yes Lessons learned Of Earth and Mars Time to tell the story of Tesla & Sp
aceX Have you read ? It \x92 s great ! ! The Second Last Kingdom https://t.c
o/Je4EI88HmV (https://t.co/Je4EI88HmV) Haven \x92 t heard that name in years
 Dogecake YOLT https://t.co/cnOf9yjpF1 (https://t.co/cnOf9yjpF1) That \x92 s
 Damian Yeah Sure The great thing about restaurants is that you get to hang o
ut with strangers ! - SJM The future currency of Earth Just a scratch Much wo
w ! <U+0001F5A4> Destiny Franz was essential That said , the ship landing bur
n has a clear solution . My greate https://t.co/e5Wikiugkz (https://t.co/e5Wi
kiugkz) Will still use hot gas maneuvering ( RCS ) thrusters , https://t.co/v
sO9h4Ioed (https://t.co/vsO9h4Ioed) Higher Isp too Intuitively , it would see
m so , but turbopump-fed Raptors have mu https://t.co/lBTG1sIBuC (https://t.c
```

In [144]:
```
no_punc_text = tokens_text.translate(str.maketrans('','',string.punctuation))
no_punc_text
```

Out[144]:
```
'I \x92 m an alien Ray tracing on Cyberpunk with HDR is nextlevel  Have you t
ried it  Great interview  Doge is underestimated Congratulations Tesla China
for amazing execution last year  Now on to the next for even more  Happy New
Year of the Ox  httpstco9WFKMYu2oj Frodo was the underdoge  All thought he wo
uld fail  Himself most of all  httpstcozGxJFDzzrM Haha thanks  Indeed  Tweets
definitely do not represent realworld time allocation  The most entertaining
outcome is the most likely Just sent some Just agree to do Clubhouse with htt
pstco3rWE9uHSTS It \x92 s getting real Bought some Dogecoin for lil X  so he
can be a toddler hodler He definitely has issues  but the sentencing seems a
bit high Thanks for fixing Please unlock my account U0001F923 U0001F923 This
is true power haha httpstcoFc9uhQSd7O Any crypto wallet that won \x92 t give
you your private keys should be avoided at all costs Your app sucks RT  NASA
has selected Falcon Heavy to launch the first two elements of the lunar Gatew
ay together on one mission  httpstco3pWt Yes Once we can predict cash flow re
asonably well  Starlink will IPO Starlink is a staggeringly difficult technic
al  economic endeavor  However  if we don httpstco9Z8Ac6skqx SpaceX needs to
pass through a deep chasm of negative cash flow over the next year or httpstc
o7J1c92hdjl Lowest cost per ton of carbon sequestered  net of value of any pr
oduct made  Must be scalable to g httpstcoXMyI7qWSgw It \x92 s meant to be th
```

In [145]:
```
no_url_text = re.sub(r'http\S+', '',no_punc_text)
no_url_text
```

```
questered  net of value of any product made  Must be scalable to g  It \x92 s
meant to be the same price in all countries  Only difference should be taxes
shipping  This is intended for Earth  but there may be some ideas that apply
to Mars too U0001F923 U0001F923 XPrize team will manage the  100M carbon capt
ure prize  Everyone at Tesla receives stock  My comp is all stock  options  w
hich I do not take off the table  That \x92 s what you \x92 re missing  Back
to work I go Does seem a bit high Doge appears to be inflationary  but is not
meaningfully so  fixed  of coins per unit time  whereas  Wow 1  Orbital laun
ch tower that can stack 2  Enough Raptors for orbit booster 3  Improve ship
booster mass Back to work tonight  Ð is for Ðogecoin  Instructional video  T
he people have spoken  So cute U0001F495 Extremely misleading image  as doesn
\x92 t reflect true time cost to people or rain  pain U0001F3B6 Who let the D
oge out U0001F3B6 Hodl the rainforests  So it \x92 s finally come to this  I
t \x92 s the most fun crypto  Its simplicity is its genius Yup U0001F923 U000

1F923 True U0001F923 U0001F923 Not that easy  This is two decades of intense
work  Have to look at old notes  emails  texts  Yes Lessons learned Of Earth
and Mars Time to tell the story of Tesla  SpaceX Have you read  It \x92 s gre
at  The Second Last Kingdom  Haven \x92 t heard that name in years Dogecake
YOLT  That \x92 s Damian Yeah Sure The great thing about restaurants is that
```

In [146]:
```
text_tokens = word_tokenize(no_url_text)
print(text_tokens)
```

```
u', '\x92', 're', 'missing', 'Back', 'to', 'work', 'I', 'go', 'Does', 'seem',
'a', 'bit', 'high', 'Doge', 'appears', 'to', 'be', 'inflationary', 'but', 'i
s', 'not', 'meaningfully', 'so', 'fixed', 'of', 'coins', 'per', 'unit', 'tim
e', 'whereas', 'Wow', '1', 'Orbital', 'launch', 'tower', 'that', 'can', 'stac
k', '2', 'Enough', 'Raptors', 'for', 'orbit', 'booster', '3', 'Improve', 'shi
p', 'booster', 'mass', 'Back', 'to', 'work', 'tonight', 'Ð', 'is', 'for', 'Ðo
gecoin', 'Instructional', 'video', 'The', 'people', 'have', 'spoken', 'So',
'cute', 'U0001F495', 'Extremely', 'misleading', 'image', 'as', 'doesn', '\x9
2', 't', 'reflect', 'true', 'time', 'cost', 'to', 'people', 'or', 'rain', 'pa
in', 'U0001F3B6', 'Who', 'let', 'the', 'Doge', 'out', 'U0001F3B6', 'Hodl', 't
he', 'rainforests', 'So', 'it', '\x92', 's', 'finally', 'come', 'to', 'this',
'It', '\x92', 's', 'the', 'most', 'fun', 'crypto', 'Its', 'simplicity', 'is',
'its', 'genius', 'Yup', 'U0001F923', 'U0001F923', 'True', 'U0001F923', 'U0001
F923', 'Not', 'that', 'easy', 'This', 'is', 'two', 'decades', 'of', 'intens
e', 'work', 'Have', 'to', 'look', 'at', 'old', 'notes', 'emails', 'texts', 'Y
es', 'Lessons', 'learned', 'Of', 'Earth', 'and', 'Mars', 'Time', 'to', 'tel
l', 'the', 'story', 'of', 'Tesla', 'SpaceX', 'Have', 'you', 'read', 'It', '\x
92', 's', 'great', 'The', 'Second', 'Last', 'Kingdom', 'Haven', '\x92', 't',
'heard', 'that', 'name', 'in', 'years', 'Dogecake', 'YOLT', 'That', '\x92',
's', 'Damian', 'Yeah', 'Sure', 'The', 'great', 'thing', 'about', 'restaurant
```

In [147]:
```
# Tokenization
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\abhinav4259\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\abhinav4259\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[147]:
```
True
```

In [148]:
```
len(text_tokens)
```

Out[148]:
```
17847
```

In [149]:
```
#remove stopwords
stopwords = stopwords.words('english')

sw_list = ['\x92','rt','ye','yeah','haha','Yes','U0001F923','I']
stopwords.extend(sw_list)

no_stop_tokens = [word for word in text_tokens if not word in stopwords]
print(no_stop_tokens)
```

```
['alien', 'Ray', 'tracing', 'Cyberpunk', 'HDR', 'nextlevel', 'Have', 'tried',
'Great', 'interview', 'Doge', 'underestimated', 'Congratulations', 'Tesla',
'China', 'amazing', 'execution', 'last', 'year', 'Now', 'next', 'even', 'Happ
y', 'New', 'Year', 'Ox', 'Frodo', 'underdoge', 'All', 'thought', 'would', 'fa
il', 'Himself', 'Haha', 'thanks', 'Indeed', 'Tweets', 'definitely', 'represen
t', 'realworld', 'time', 'allocation', 'The', 'entertaining', 'outcome', 'lik
ely', 'Just', 'sent', 'Just', 'agree', 'Clubhouse', 'It', 'getting', 'real',
'Bought', 'Dogecoin', 'lil', 'X', 'toddler', 'hodler', 'He', 'definitely', 'i
ssues', 'sentencing', 'seems', 'bit', 'high', 'Thanks', 'fixing', 'Please',
'unlock', 'account', 'This', 'true', 'power', 'Any', 'crypto', 'wallet', 'giv
e', 'private', 'keys', 'avoided', 'costs', 'Your', 'app', 'sucks', 'RT', 'NAS
A', 'selected', 'Falcon', 'Heavy', 'launch', 'first', 'two', 'elements', 'lun
ar', 'Gateway', 'together', 'one', 'mission', 'Once', 'predict', 'cash', 'flo
w', 'reasonably', 'well', 'Starlink', 'IPO', 'Starlink', 'staggeringly', 'dif
ficult', 'technical', 'economic', 'endeavor', 'However', 'SpaceX', 'needs',
'pass', 'deep', 'chasm', 'negative', 'cash', 'flow', 'next', 'year', 'Lowes
t', 'cost', 'per', 'ton', 'carbon', 'sequestered', 'net', 'value', 'product',
'made', 'Must', 'scalable', 'g', 'It', 'meant', 'price', 'countries', 'Only',
'difference', 'taxes', 'shipping', 'This', 'intended', 'Earth', 'may', 'idea
```

In [150]:
```
# Normalize the data
lower_words = [Text.lower() for Text in no_stop_tokens]
print(lower_words[100:200])
```

```
['once', 'predict', 'cash', 'flow', 'reasonably', 'well', 'starlink', 'ipo', 's
tarlink', 'staggeringly', 'difficult', 'technical', 'economic', 'endeavor', 'ho
wever', 'spacex', 'needs', 'pass', 'deep', 'chasm', 'negative', 'cash', 'flow',
'next', 'year', 'lowest', 'cost', 'per', 'ton', 'carbon', 'sequestered', 'net',
'value', 'product', 'made', 'must', 'scalable', 'g', 'it', 'meant', 'price', 'c
ountries', 'only', 'difference', 'taxes', 'shipping', 'this', 'intended', 'eart
h', 'may', 'ideas', 'apply', 'mars', 'xprize', 'team', 'manage', '100m', 'carbo
n', 'capture', 'prize', 'everyone', 'tesla', 'receives', 'stock', 'my', 'comp',
'stock', 'options', 'take', 'table', 'that', 'missing', 'back', 'work', 'go',
'does', 'seem', 'bit', 'high', 'doge', 'appears', 'inflationary', 'meaningfull
y', 'fixed', 'coins', 'per', 'unit', 'time', 'whereas', 'wow', '1', 'orbital',
'launch', 'tower', 'stack', '2', 'enough', 'raptors', 'orbit', 'booster']
```

In [151]:
```
from nltk.stem import PorterStemmer
ps =PorterStemmer()
stemmed_tokens = [ps.stem(word) for word in lower_words]
print(stemmed_tokens[100:200])
```

```
['onc', 'predict', 'cash', 'flow', 'reason', 'well', 'starlink', 'ipo', 'starli
nk', 'staggeringli', 'difficult', 'technic', 'econom', 'endeavor', 'howev', 'sp
acex', 'need', 'pass', 'deep', 'chasm', 'neg', 'cash', 'flow', 'next', 'year',
'lowest', 'cost', 'per', 'ton', 'carbon', 'sequest', 'net', 'valu', 'product',
'made', 'must', 'scalabl', 'g', 'it', 'meant', 'price', 'countri', 'onli', 'dif
fer', 'tax', 'ship', 'thi', 'intend', 'earth', 'may', 'idea', 'appli', 'mar',
'xprize', 'team', 'manag', '100m', 'carbon', 'captur', 'prize', 'everyon', 'tes
la', 'receiv', 'stock', 'my', 'comp', 'stock', 'option', 'take', 'tabl', 'tha
t', 'miss', 'back', 'work', 'go', 'doe', 'seem', 'bit', 'high', 'doge', 'appea
r', 'inflationari', 'meaning', 'fix', 'coin', 'per', 'unit', 'time', 'wherea',
'wow', '1', 'orbit', 'launch', 'tower', 'stack', '2', 'enough', 'raptor', 'orbi
t', 'booster']
```

In [152]:
```
nlp = spacy.load('en_core_web_sm')
doc = nlp(' '.join(lower_words))
print(doc)
```

```
alien ray tracing cyberpunk hdr nextlevel have tried great interview doge und
erestimated congratulations tesla china amazing execution last year now next
even happy new year ox frodo underdoge all thought would fail himself haha th
anks indeed tweets definitely represent realworld time allocation the enterta
ining outcome likely just sent just agree clubhouse it getting real bought do
gecoin lil x toddler hodler he definitely issues sentencing seems bit high th
anks fixing please unlock account this true power any crypto wallet give priv
ate keys avoided costs your app sucks rt nasa selected falcon heavy launch fi
rst two elements lunar gateway together one mission once predict cash flow re
asonably well starlink ipo starlink staggeringly difficult technical economic
endeavor however spacex needs pass deep chasm negative cash flow next year lo
west cost per ton carbon sequestered net value product made must scalable g i
t meant price countries only difference taxes shipping this intended earth ma
y ideas apply mars xprize team manage 100m carbon capture prize everyone tesl
a receives stock my comp stock options take table that missing back work go d
oes seem bit high doge appears inflationary meaningfully fixed coins per unit
time whereas wow 1 orbital launch tower stack 2 enough raptors orbit booster
3 improve ship booster mass back work tonight ð ðogecoin instructional video
the people spoken so cute u0001f495 extremely misleading image reflect true t
```

In [153]:
```
lemmas = [token.lemma_ for token in doc]
print(lemmas)
```

```
e', 'stock', 'my', 'comp', 'stock', 'option', 'take', 'table', 'that', 'mis
s', 'back', 'work', 'go', 'do', 'seem', 'bit', 'high', 'doge', 'appear', 'inf
lationary', 'meaningfully', 'fix', 'coin', 'per', 'unit', 'time', 'whereas',
'wow', '1', 'orbital', 'launch', 'tower', 'stack', '2', 'enough', 'raptor',
'orbit', 'booster', '3', 'improve', 'ship', 'booster', 'mass', 'back', 'wor
k', 'tonight', 'ð', 'ðogecoin', 'instructional', 'video', 'the', 'people', 's
peak', 'so', 'cute', 'u0001f495', 'extremely', 'misleading', 'image', 'reflec
t', 'true', 'time', 'cost', 'people', 'rain', 'pain', 'u0001f3b6', 'who', 'le
t', 'doge', 'u0001f3b6', 'hodl', 'rainforest', 'so', 'finally', 'come', 'it',
'fun', 'crypto', 'its', 'simplicity', 'genius', 'yup', 'true', 'not', 'easy',
'this', 'two', 'decade', 'intense', 'work', 'have', 'look', 'old', 'note', 'e
mail', 'text', 'lesson', 'learn', 'of', 'earth', 'mar', 'time', 'tell', 'stor
y', 'tesla', 'spacex', 'have', 'read', 'it', 'great', 'the', 'second', 'las
t', 'kingdom', 'haven', 'hear', 'name', 'year', 'dogecake', 'yolt', 'that',
'damian', 'yeah', 'sure', 'the', 'great', 'thing', 'restaurant', 'get', 'han
g', 'stranger', 'sjm', 'the', 'future', 'currency', 'earth', 'just', 'scratc
h', 'much', 'wow', 'u0001f5a4', 'destiny', 'franz', 'essential', 'that', 'sa
y', 'ship', 'landing', 'burn', 'clear', 'solution', 'my', 'greate', 'will',
'still', 'use', 'hot', 'gas', 'maneuvering', 'rcs', 'thruster', 'high', 'is
```

In [154]:
```
clean_tweets = ' '.join(lemmas)
clean_tweets
```

```
ga berlin progress neuralink work super hard ensure implant safety close comm
unication rt this mission enable access everyday people dream go space rt ann
ounce first commercial astronaut mission orbit earth aboard dragon u2192 if w
ork advanced wearable phone robot skill need feel weird helping make hopefull
y good version cyberpunk come true please consider work neuralink shortterm s
olve brain spine injury longterm human ai symbiosis latte on clubhouse tonigh
t 10 pm la time tom great story experim rt launch alert u0001f680 target earl
y april 20 launch second crew rotation mis he become big fan methane that sou
nd correct tom certainly deserve lot cre tom great instrumental develop early
ver t w 15 accelerate unusually fast high t w important reusable v no escapin
g read whole article warm sunny day snowy mountain what beautiful day la yeah
dr frankensteen never use guy he give talk spacex only halo sure hope true te
sla spacex cryoproof install engine starship sn9 sn10 in retrospect inevitabl
e great shot live sword die sword entropy buy hold company make good produce
service love earth small small still u0001f440 with cyberpunk even hotfixe li
terally hotfixe great game the dollar short indeed shopify great spacex use i
ndeed the economy \x97 make useful product provide great service \x97 actuall
y matter tanstaafl would better small fee fee latter make robin here come sho
rty apologist give respect get shorty u sell house u u sell car u u sell stoc
```

In [155]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv =CountVectorizer()
tweetscv = cv.fit_transform(lemmas)
print(cv.vocabulary_)
```

```
2195, 'flameout': 1157, 'risk': 2465, 'la': 1632, 'foolish': 1174, 'we': 330
8, 'start': 2771, 'immediately': 1475, 'land': 1638, 'become': 379, 'meme': 1
833, 'destroyer': 834, 'short': 2609, 'might': 1858, 'actually': 152, 'happe
n': 1341, 'sandstorm': 2512, 'masterpiece': 1799, 'dumb': 923, 'pull': 2314,
'method': 1848, 'lowkey': 1752, 'loki': 1731, 'no': 1973, 'highs': 1401, 'gig
achad': 1271, 'ur': 3223, 'welcome': 3322, 'sea': 2541, 'wind': 3359, 'toug
h': 3013, 'watch': 3300, 'u2192': 3157, 'off': 2028, 'twitter': 3070, 'fill':
1133, 'graffiti': 1299, 'art': 278, 'u043c': 3147, 'u044b': 3153, 'u0441': 31
52, 'u0434': 3142, 'u0435': 3143, 'u043b': 3146, 'u0430': 3139, 'giga': 1270,
'berlin': 394, 'progress': 2286, 'neuralink': 1958, 'super': 2842, 'hard': 13
45, 'ensure': 999, 'implant': 1478, 'safety': 2505, 'close': 607, 'communicat
ion': 640, 'enable': 986, 'access': 132, 'everyday': 1036, 'dream': 912, 'spa
ce': 2720, 'announce': 232, 'commercial': 637, 'astronaut': 292, 'aboard': 11
2, 'dragon': 908, 'if': 1463, 'advanced': 162, 'wearable': 3312, 'phone': 216
2, 'robot': 2474, 'skill': 2651, 'feel': 1121, 'weird': 3320, 'helping': 138
4, 'hopefully': 1425, 'version': 3257, 'consider': 676, 'shortterm': 2612, 's
olve': 2702, 'brain': 463, 'spine': 2745, 'injury': 1520, 'longterm': 1736,
'human': 1442, 'ai': 183, 'symbiosis': 2877, 'latte': 1652, 'on': 2037, '10':
2, 'pm': 2191, 'tom': 2995, 'experim': 1065, 'alert': 190, 'u0001f680': 3128,
'target': 2899, 'early': 934, 'april': 263, '20': 26, 'crew': 739, 'rotatio
```

In [156]:
```
print(cv.get_feature_names()[100:200])
```

```
['74', '78', '7th', '90', '9007', '922', '948', '95', '99', 'aber', 'able', 'ab
o', 'aboard', 'abort', 'about', 'above', 'absence', 'absolute', 'absolutely',
'absorb', 'absorption', 'absurd', 'absurdly', 'ac', 'academia', 'accel', 'accel
era', 'accelerate', 'acceleration', 'accelerator', 'accept', 'acceptable', 'acc
ess', 'accessible', 'accident', 'accidental', 'accommodate', 'account', 'accur
a', 'accuracy', 'accurate', 'ace', 'achieve', 'achievement', 'achy', 'acquisiti
on', 'across', 'action', 'active', 'activity', 'actual', 'actuall', 'actually',
'actuary', 'adagio', 'add', 'additive', 'address', 'administer', 'adult', 'adva
nc', 'advance', 'advanced', 'advantage', 'adventure', 'advertise', 'advice', 'a
dvise', 'aero', 'afb', 'affair', 'affect', 'affordable', 'africa', 'after', 'af
ternoon', 'age', 'ago', 'agony', 'agree', 'ah', 'ahead', 'ahem', 'ai', 'aim',
'air', 'aircraft', 'airplane', 'ak', 'aka', 'alert', 'alexander', 'algo', 'algo
rithm', 'alien', 'align', 'all', 'allocati', 'allocation', 'allow']
```

In [157]:
```
print(tweetscv.toarray()[100:200])
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

In [158]:
```
print(tweetscv.toarray().shape)
```

```
(11486, 3421)
```

In [159]:
```
## countvectorizer with N-grams
```

In [160]:
```
ngram_range = CountVectorizer(analyzer='word',ngram_range = (1,3),max_features=10
bow_matrix_ngram = ngram_range.fit_transform(lemmas)
```

In [161]:
```
print(ngram_range.get_feature_names())
print(bow_matrix_ngram.toarray())
```

```
['actually', 'ai', 'also', 'back', 'big', 'booster', 'car', 'come', 'cool', 'co
uld', 'crew', 'day', 'design', 'do', 'dragon', 'earth', 'engine', 'even', 'eve
r', 'exactly', 'falcon', 'first', 'flight', 'fsd', 'future', 'get', 'go', 'goo
d', 'great', 'haha', 'hard', 'high', 'if', 'it', 'just', 'land', 'launch', 'lif
e', 'like', 'look', 'lot', 'love', 'make', 'many', 'mar', 'maybe', 'mission',
'model', 'much', 'need', 'new', 'next', 'no', 'not', 'ok', 'one', 'part', 'peop
le', 'point', 'pretty', 'probably', 'production', 'right', 'rocket', 'rt', 'sa
y', 'seem', 'soon', 'space', 'spacex', 'starlink', 'starship', 'still', 'supe
r', 'sure', 'take', 'tesla', 'test', 'thank', 'that', 'the', 'there', 'they',
'think', 'this', 'time', 'true', 'try', 'ufe0f', 'use', 'way', 'we', 'week', 'w
ell', 'will', 'work', 'would', 'yeah', 'year', 'you']
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

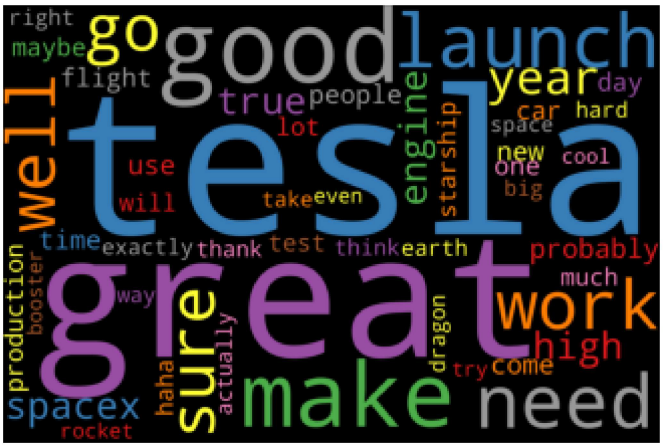
In [162]:
```
## TF-IDF Vectorizer
```

In [163]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidfv_ngram_max_features = TfidfVectorizer(analyzer = 'word',ngram_range = (1,3)
tfidf_matrix_ngram = tfidfv_ngram_max_features.fit_transform(lemmas)
```

In [164]:
```
print(tfidfv_ngram_max_features.get_feature_names())
print(tfidf_matrix_ngram.toarray())
```

```
['actually', 'ai', 'also', 'back', 'big', 'booster', 'car', 'come', 'cool', 'co
uld', 'crew', 'day', 'design', 'do', 'dragon', 'earth', 'engine', 'even', 'eve
r', 'exactly', 'falcon', 'first', 'flight', 'fsd', 'future', 'get', 'go', 'goo
d', 'great', 'haha', 'hard', 'high', 'if', 'it', 'just', 'land', 'launch', 'lif
e', 'like', 'look', 'lot', 'love', 'make', 'many', 'mar', 'maybe', 'mission',
'model', 'much', 'need', 'new', 'next', 'no', 'not', 'ok', 'one', 'part', 'peop
le', 'point', 'pretty', 'probably', 'production', 'right', 'rocket', 'rt', 'sa
y', 'seem', 'soon', 'space', 'spacex', 'starlink', 'starship', 'still', 'supe
r', 'sure', 'take', 'tesla', 'test', 'thank', 'that', 'the', 'there', 'they',
'think', 'this', 'time', 'true', 'try', 'ufe0f', 'use', 'way', 'we', 'week', 'w
ell', 'will', 'work', 'would', 'yeah', 'year', 'you']
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

In [165]:
```
def plot_cloud(wordcloud):
    plt.imshow(wordcloud)
    plt.axis('off')

STOPWORDS.add('pron')
STOPWORDS.add('rt')
STOPWORDS.add('yeah')
wordcloud=WordCloud(width=3000,height=2000,background_color='black',max_words=50,
plot_cloud(wordcloud)
```



NER (named entity recognition)

In [166]:
```
nlp = spacy.load('en_core_web_sm')

one_block = clean_tweets
doc_block = nlp(one_block)
spacy.displacy.render(doc_block,style = 'ent',jupyter = True)
```

alien ray trace cyberpunk hdr nextlevel have try great interview doge underestimate congratulation tesla china GPE amazing execution last year DATE now next even happy new year DATE ox frodo underdoge all thought would fail himself haha thank indeed tweet definitely represent realworld time allocation the entertaining outcome likely just send just agree clubhouse it get real buy dogecoin lil x toddler PRODUCT hodler he definitely issue sentencing seem bit high thank fix please unlock account this true power any crypto wallet give private key avoid cost your app suck rt nasa ORG select falcon heavy launch first ORDINAL two CARDINAL element lunar gateway together one CARDINAL mission once predict cash flow reasonably well starlink ipo starlink staggeringly difficult technical economic

In [167]:
```
for token in doc_block[100:200]:
    print(token,token.pos_)
```

```
once ADV
predict VERB
cash NOUN
flow NOUN
reasonably ADV
well ADV
starlink VERB
ipo NOUN
starlink NOUN
staggeringly ADV
difficult ADJ
technical ADJ
economic ADJ
endeavor NOUN
however ADV
spacex VERB
need NOUN
pass VERB
deep ADJ
chasm NOUN
negative ADJ
cash NOUN
flow NOUN
next ADJ
year NOUN
low ADJ
cost NOUN
per ADP
ton NOUN
carbon NOUN
sequester NOUN
net ADJ
value NOUN
product NOUN
make VERB
must AUX
scalable VERB
g ADV
it PRON
mean VERB
price NOUN
country NOUN
only ADV
difference NOUN
taxis NOUN
ship NOUN
this PRON
intend ADJ
earth NOUN
may AUX
idea VERB
apply VERB
mars PROPN
xprize PROPN
```

```
team NOUN
manage VERB
100 NUM
m VERB
carbon NOUN
capture NOUN
prize NOUN
everyone PRON
tesla ADV
receive VERB
stock NOUN
my PRON
comp NOUN
stock NOUN
option NOUN
take VERB
table NOUN
that PRON
miss VERB
back ADJ
work NOUN
go NOUN
do AUX
seem VERB
bit ADV
high ADJ
doge PROPN
appear VERB
inflationary ADJ
meaningfully ADV
fix VERB
coin NOUN
per ADP
unit NOUN
time NOUN
whereas SCONJ
wow INTJ
1 NUM
orbital ADJ
launch NOUN
tower NOUN
stack VERB
2 NUM
enough ADJ
raptor NOUN
orbit NOUN
```

In [168]:
```
# Filtering the nouns and verbs only
nouns_verbs = [token.text for token in doc_block if token.pos_ in ('NOUN','VERB')]
print(nouns_verbs[100:200])
```

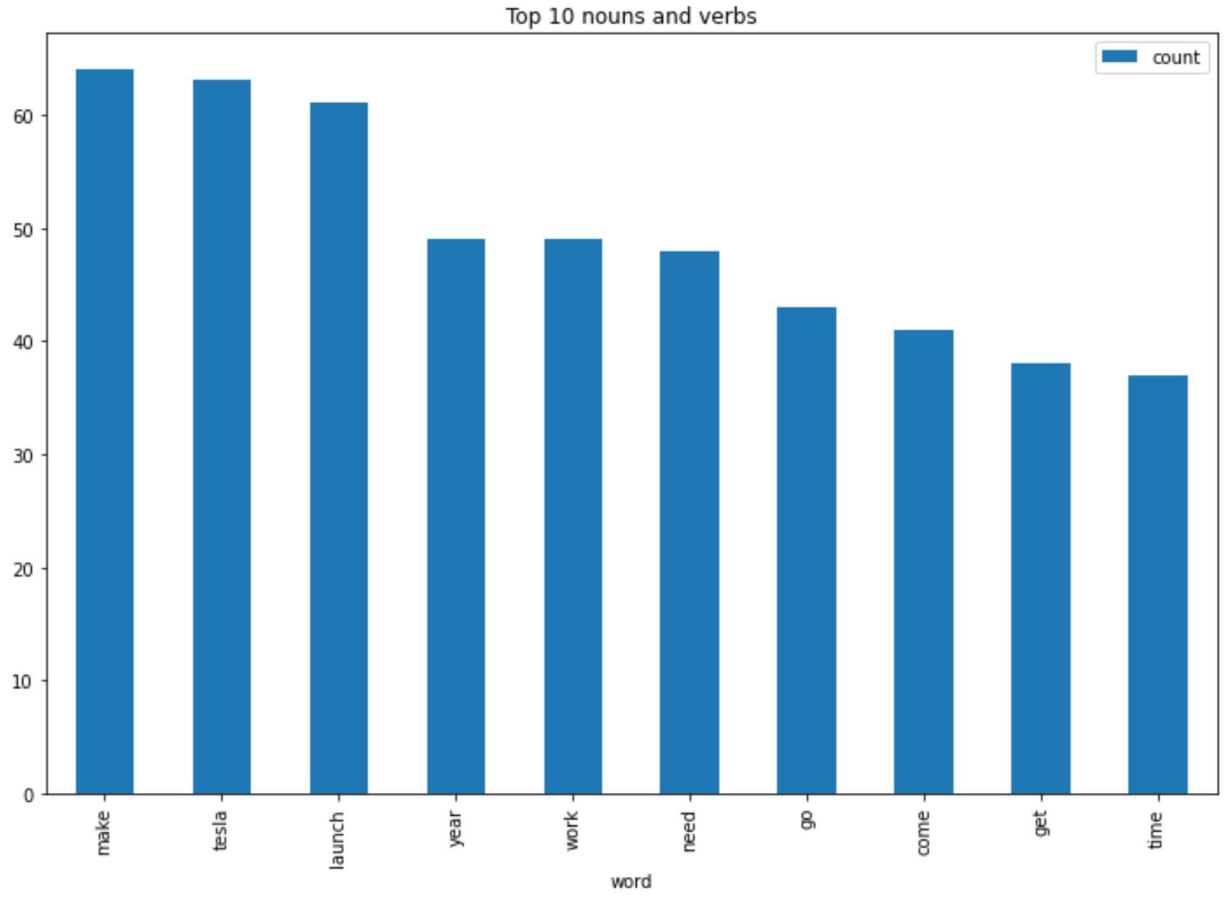
```
['time', 'launch', 'tower', 'stack', 'raptor', 'orbit', 'booster', 'improve',
'ship', 'booster', 'mass', 'work', 'tonight', 'ðogecoin', 'video', 'people', 's
peak', 'cute', 'image', 'reflect', 'time', 'cost', 'people', 'rain', 'pain', 'u
0001f3b6', 'let', 'come', 'fun', 'crypto', 'simplicity', 'genius', 'yup', 'deca
de', 'work', 'look', 'note', 'email', 'text', 'lesson', 'learn', 'earth', 'tim
e', 'tell', 'story', 'tesla', 'spacex', 'read', 'hear', 'name', 'year', 'dogeca
ke', 'yolt', 'thing', 'restaurant', 'get', 'hang', 'stranger', 'sjm', 'currenc
y', 'earth', 'scratch', 'u0001f5a4', 'destiny', 'franz', 'say', 'ship', 'landin
g', 'burn', 'solution', 'greate', 'use', 'gas', 'maneuvering', 'rcs', 'thruste
r', 'seem', 'turbopumpfed', 'raptor', 'falcon', 'launch', 'collect', 'galaxy',
'explore', 'launch', 'starlink', 'satellite', 'orbit', 'mission', 'pad', '39a',
'deck', 'default', 'engine', 'lever', 'arm', 'shut', 'engine', 'min', 'throttl
e']
```

In [169]:
```
# Counting the noun & verb tokens
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()

X = cv.fit_transform(nouns_verbs)
sum_words = X.sum(axis=0)

words_freq = [(word,sum_words[0,idx]) for word,idx in cv.vocabulary_.items()]
words_freq = sorted(words_freq, key = lambda x: x[1],reverse=True)

wd_df = pd.DataFrame(words_freq)
wd_df.columns=['word','count']
wd_df[0:10] # viewing top ten results
```

Out[169]:

	word	count
0	make	64
1	tesla	63
2	launch	61
3	year	49
4	work	49
5	need	48
6	go	43
7	come	41
8	get	38
9	time	37

In [170]:
```
wd_df[0:10].plot.bar(x='word',figsize=(12,8),title = 'Top 10 nouns and verbs');
```



In [171]:
```
from nltk import tokenize
sentences = tokenize.sent_tokenize(' '.join(data))
sentences
```

```
 'Now on to the next for even more!!',
 'Happy New Year of the Ox!',
 'https://t.co/9WFKMYu2oj Frodo was the underdoge,\nAll thought he would fai
l,\nHimself most of all.',
 'https://t.co/zGxJFDzzrM @OwenSparks_ @flcnhvy @anonyx10 Haha thanks :) @flc
nhvy @anonyx10 Indeed!',
 'Tweets definitely do not represent real-world time allocation.',
 'The most entertaining outcome is the most likely @GiveDirectly Just sent so
me Just agree to do Clubhouse with @kanyewest https://t.co/3rWE9uHSTS (http
s://t.co/3rWE9uHSTS) @geoffkeighley @UnrealEngine It\x92s getting real Bought
some Dogecoin for lil X, so he can be a toddler hodler @JoshManMode He defini
tely has issues, but the sentencing seems a bit high @freewalletorg Thanks fo
r fixing @freewalletorg Please unlock my account @AstroJordy <U+0001F923><U+0
001F923> This is true power haha https://t.co/Fc9uhQSd7O (https://t.co/Fc9uhQ
Sd7O) @freewalletorg Any crypto wallet that won\x92t give you your private ke
ys should be avoided at all costs @freewalletorg Your app sucks RT @SpaceX: N
ASA has selected Falcon Heavy to launch the first two elements of the lunar G
ateway together on one mission!',
 'https://t.co/3pWt @ajtourville Yes @BLKMDL3 @RationalEtienne @Adamklotz_ On
ce we can predict cash flow reasonably well, Starlink will IPO @RationalEtien
```

In [172]:
```
sent_df = pd.DataFrame(sentences,columns = ['sentence'])
sent_df
```

Out[172]:

	sentence
0	@kunalb11 I m an alien @ID_AA_Carmack Ray trac...
1	Have you tried it?
2	@joerogan @Spotify Great interview!
3	@gtera27 Doge is underestimated @teslacn Congr...
4	Now on to the next for even more!!
...	...
919	@kenyanwalstreet Not actually a payout, just a...
920	It may never pay out, as the stock can t b ht...
921	Details Aug 28.
922	AI symbiosis while u wait @vistacruiser7 @flcn...
923	@TeslaGong @PPathole Samwise Gamgee @PPathole ...

924 rows × 1 columns

In [173]:
```
affin = pd.read_csv('Afinn.csv',sep=',',encoding='Latin-1')
affin
```

Out[173]:

	word	value
0	abandon	-2
1	abandoned	-2
2	abandons	-2
3	abducted	-2
4	abduction	-2
...	...	...
2472	yucky	-2
2473	yummy	3
2474	zealot	-2
2475	zealots	-2
2476	zealous	2

2477 rows × 2 columns

In [174]:
```
affinity_scores = affin.set_index('word')['value'].to_dict()
affinity_scores
```

```
 'absentee': -1,
 'absentees': -1,
 'absolve': 2,
 'absolved': 2,
 'absolves': 2,
 'absolving': 2,
 'absorbed': 1,
 'abuse': -3,
 'abused': -3,
 'abuses': -3,
 'abusive': -3,
 'accept': 1,
 'accepted': 1,
 'accepting': 1,
 'accepts': 1,
 'accident': -2,
 'accidental': -2,
 'accidentally': -2,
 'accidents': -2,
```

In [175]:
```
nlp = spacy.load('en_core_web_sm')
sentiment_lexicon = affinity_scores

def calculate_sentiment(text:str=None):
    sent_score=0
    if text:
        sentence=nlp(text)
        for word in sentence:
            sent_score+=sentiment_lexicon.get(word.lemma_,0)
    return sent_score
```

In [176]:
```
calculate_sentiment(text = 'great')
```

Out[176]:
```
3
```

In [180]:
```
#calculating sentiment value for each sentence
sent_df['sentiment_value'] = sent_df['sentence'].apply(calculate_sentiment)
sent_df['sentiment_value']
```

Out[180]:
```
0      0
1      0
2      3
3      3
4      0
      ..
919    0
920   -4
921    0
922   -2
923    0
Name: sentiment_value, Length: 924, dtype: int64
```

In [181]:
```
# how many words are there in a sentence?
sent_df['word_count'] = sent_df['sentence'].str.split().apply(len)
sent_df['word_count']
```

Out[181]:
```
0      13
1       4
2       4
3      13
4       8
       ..
919    11
920    31
921     3
922    47
923    15
Name: word_count, Length: 924, dtype: int64
```

In [182]:
```
sent_df.sort_values(by='sentiment_value')
```

Out[182]:

	sentence	sentiment_value	word_count
647	Very ba https://t.co/tJsh1Exz1Q @justpaulinel...	-8	60
64	Also, the road to hell is mostly paved with ba...	-7	11
837	Cool Model 3 review by @iamjamiefoxx https://t...	-7	61
611	Then static fire, checkouts, static fire, fly ...	-4	12
920	It may never pay out, as the stock can t b ht...	-4	31
...	...	...	...
81	@teslaownersSV This is a good one @MrBeastYT I...	13	38
585	The open areas https://t.co/rabjKrtQlw @Sav...	14	138
719	We just haven t observed the https://t.co/mez...	15	72
36	@ajtourville @Erdayastronaut @SpaceX Yes, but ...	16	231
105	@Erdayastronaut @SpaceX Was also thinking that...	16	94

924 rows × 3 columns

In [183]:
```
# sentiment score of the whole review
sent_df['sentiment_value'].describe()
```

Out[183]:
```
count    924.000000
mean       1.345238
std        2.684749
min       -8.000000
25%        0.000000
50%        0.000000
75%        3.000000
max       16.000000
Name: sentiment_value, dtype: float64
```

In [184]:
```
# negative sentiment score of the whole review
sent_df[sent_df['sentiment_value']<=0]
```

Out[184]:

	sentence	sentiment_value	word_count
0	@kunalb11 I m an alien @ID_AA_Carmack Ray trac...	0	13
1	Have you tried it?	0	4
4	Now on to the next for even more!!	0	8
5	Happy New Year of the Ox!	0	6
6	https://t.co/9WFKMYu2oj Frodo was the underdog...	-2	14
...	...	...	...
919	@kenyanwalstreet Not actually a payout, just a...	0	11
920	It may never pay out, as the stock can t b ht...	-4	31
921	Details Aug 28.	0	3
922	AI symbiosis while u wait @vistacruiser7 @flcn...	-2	47
923	@TeslaGong @PPathole Samwise Gamgee @PPathole ...	0	15

496 rows × 3 columns

In [185]:
```
# positive sentiment score of the whole review
sent_df[sent_df['sentiment_value']>0]
```

Out[185]:

	sentence	sentiment_value	word_count
2	@joerogan @Spotify Great interview!	3	4
3	@gtera27 Doge is underestimated @teslacn Congr...	3	13
7	https://t.co/zGxJFDzzrM @OwenSparks_ @flcnhvy ...	2	10
9	The most entertaining outcome is the most like...	3	109
17	Back to work I go @CapybaraSurfer @MattWallace...	4	38
...	...	...	...
911	He was one of the very best.	3	7
913	@Ali_Afshari In general, we need to improve ho...	4	87
915	@burakaydik True Wow, IHOP & GitHub are cl...	3	15
917	This is both great & terrifying.	3	6
918	Everything we ve ever sensed or thought has be...	3	17

428 rows × 3 columns

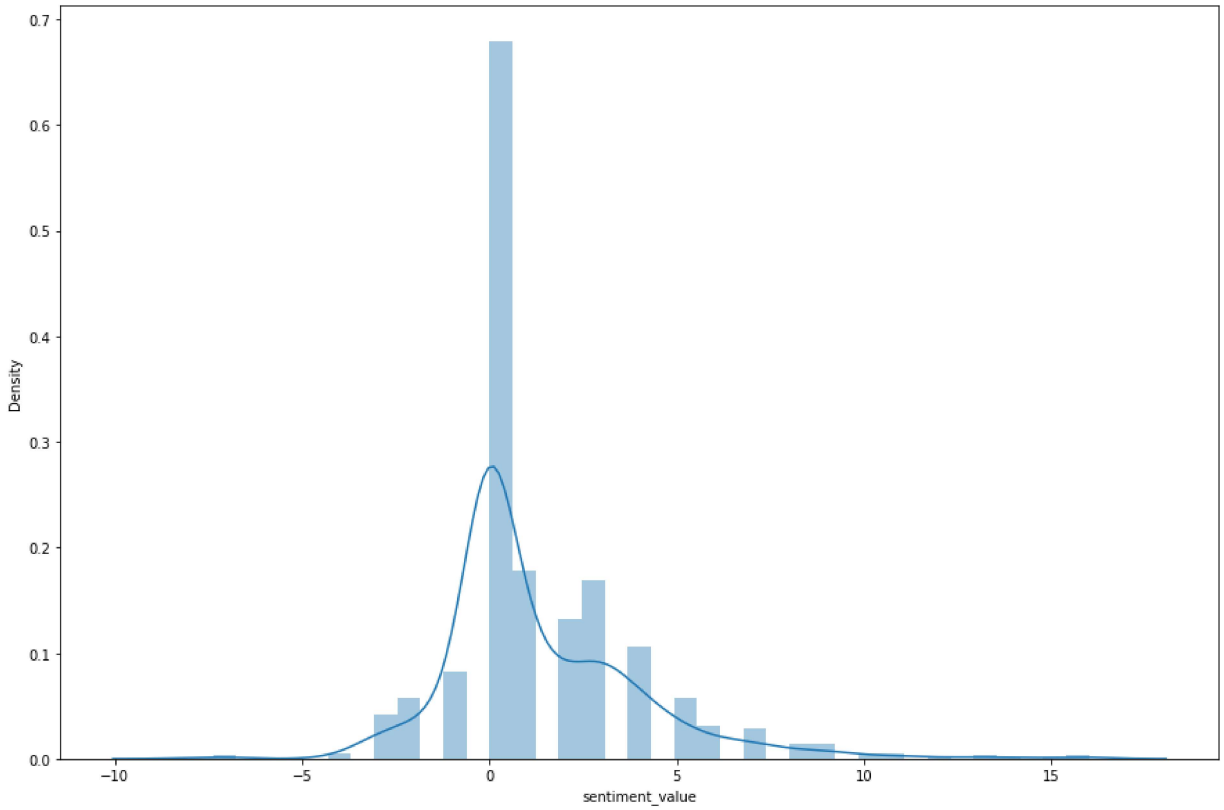
In [189]:
```
# Adding index column
sent_df['index'] = range(0,len(sent_df))
sent_df
```

Out[189]:

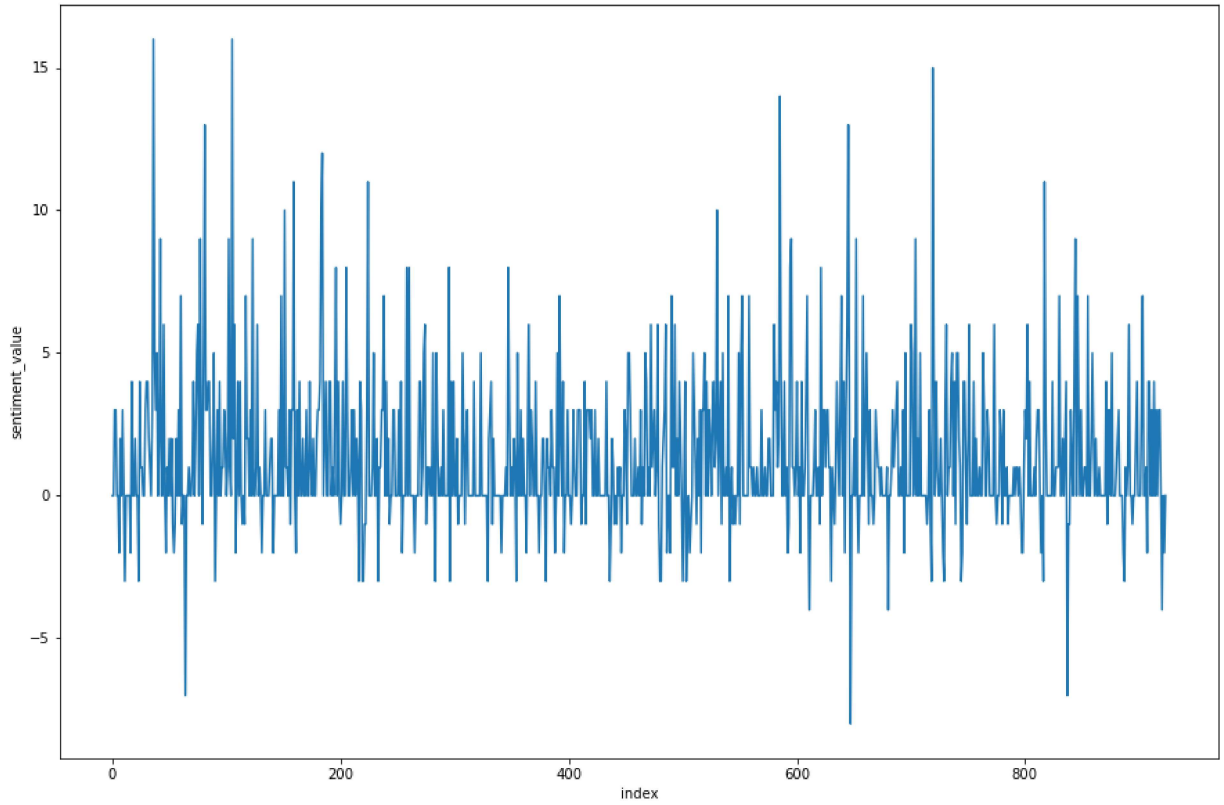
	sentence	sentiment_value	word_count	index	index
0	@kunalb11 I m an alien @ID_AA_Carmack Ray trac...	0	13	0	0
1	Have you tried it?	0	4	1	1
2	@joerogan @Spotify Great interview!	3	4	2	2
3	@gtera27 Doge is underestimated @teslacn Congr...	3	13	3	3
4	Now on to the next for even more!!	0	8	4	4
...	...	...	...	...	...
919	@kenyanwalstreet Not actually a payout, just a...	0	11	919	919
920	It may never pay out, as the stock can t b ht...	-4	31	920	920
921	Details Aug 28.	0	3	921	921
922	AI symbiosis while u wait @vistacruiser7 @flcn...	-2	47	922	922
923	@TeslaGong @PPathole Samwise Gamgee @PPathole ...	0	15	923	923

924 rows × 5 columns

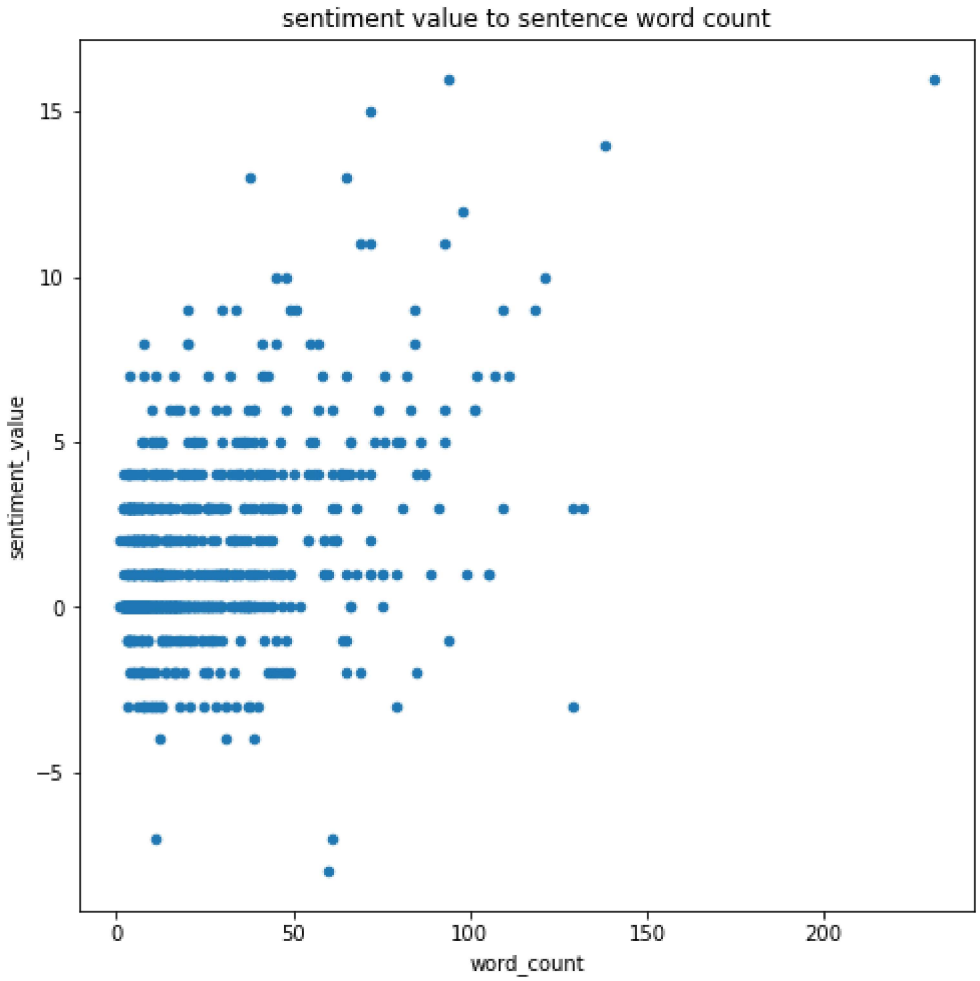
In [187]:
```
# plotting the sentiment value for whole review
import seaborn as sns
plt.figure(figsize=(15,10))
sns.distplot(sent_df['sentiment_value'])
plt.show()
```



In [190]:
```
plt.figure(figsize=(15,10))
sns.lineplot(y='sentiment_value',x='index',data=sent_df)
plt.show()
```



In [191]:
```
sent_df.plot.scatter(x='word_count',y='sentiment_value',figsize=(8,8),title='sent
plt.show()
```



In [192]:
```
Sentiment_Class'] = pd.cut(x = sent_df['sentiment_value'],bins=[-8,-1,0,17],labels
```

In [193]:
```
sent_df.sample(10)
```

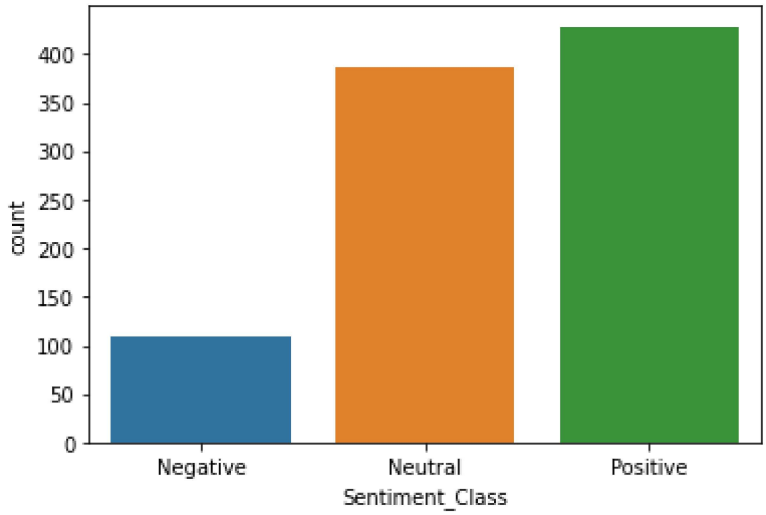
Out[193]:

	sentence	sentiment_value	word_count	index	index	Sentiment_Class
254	They stole Apple s code too.	-2	5	254	254	Negative
777	<U+0001F1FA><U+0001F1F8> returned.	0	2	777	777	Neutral
384	@engineers_feed Haha Very close to actual expe...	0	8	384	384	Neutral
246	Tesla is definitely not the only good company,...	3	44	246	246	Positive
544	Five ye https://t.co/FY4nwWbx56 @flcnhvy @Cat...	-1	21	544	544	Negative
195	@Erdayastronaut @SpaceX SN8 did great!	3	5	195	195	Positive
81	@teslaownersSV This is a good one @MrBeastYT I...	13	38	81	81	Positive
287	https://t.co/SW5RBm1sRB @JohnnaCrider1 @timmer...	0	10	287	287	Neutral
190	Seems pretty good.	4	3	190	190	Positive
601	Smartwatches & phones are yesterday s tech...	0	10	601	601	Neutral

In [194]:
```
sent_df['Sentiment_Class'].value_counts()
```

Out[194]:
```
Positive    428
Neutral     386
Negative    109
Name: Sentiment_Class, dtype: int64
```

In [195]:
```
sns.countplot(x = 'Sentiment_Class',data = sent_df)
plt.show()
```



In [ ]:
```

```