In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from textblob import TextBlob
```

In [2]:

```
data = pd.read_csv('Elon_musk.csv',encoding="latin-1")
data.head(10)
```

Out[2]:

	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...
5	6	Happy New Year of the Ox! https://t.co/9WFKMYu2oj
6	7	Frodo was the underdoge,\nAll thought he would...
7	8	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)
8	9	@flcnhvy @anonyx10 Indeed! Tweets definitely d...
9	10	The most entertaining outcome is the most likely

In [3]:

```
#Number of Words in single tweet
data['word_count'] = data['Text'].apply(lambda x: len(str(x).split(" ")))
data[['Text','word_count']].head(10)
```

Out[3]:

	Text	word_count
0	@kunalb11 I m an alien	4
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	13
2	@joerogan @Spotify Great interview!	4
3	@gtera27 Doge is underestimated	4
4	@teslacn Congratulations Tesla China for amazi...	17
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	7
6	Frodo was the underdoge,\nAll thought he would...	12
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	6
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	11
9	The most entertaining outcome is the most likely	8

In [4]:

```
#Number of characters in single tweet
data['char_count'] = data['Text'].str.len()
data[['Text','char_count']].head(10)
```

Out[4]:

	Text	char_count
0	@kunalb11 I m an alien	22
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	82
2	@joerogan @Spotify Great interview!	35
3	@gtera27 Doge is underestimated	31
4	@teslacn Congratulations Tesla China for amazi...	104
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	49
6	Frodo was the underdoge,\nAll thought he would...	96
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	46
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	89
9	The most entertaining outcome is the most likely	48

In [5]:

```
#Number of characters in single tweet
data['char_count'] = data['Text'].str.len()
data[['Text','char_count']].head(10)
```

Out[5]:

	Text	char_count
0	@kunalb11 I m an alien	22
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	82
2	@joerogan @Spotify Great interview!	35
3	@gtera27 Doge is underestimated	31
4	@teslacn Congratulations Tesla China for amazi...	104
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	49
6	Frodo was the underdoge,\nAll thought he would...	96
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	46
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	89
9	The most entertaining outcome is the most likely	48

In [6]:

```
def avg_word(sentence):
  words = sentence.split()
  return (sum(len(word) for word in words)/len(words))

data['avg_word'] = data['Text'].apply(lambda x: avg_word(x))
data[['Text','avg_word']].head(10)
```

Out[6]:

	Text	avg_word
0	@kunalb11 I m an alien	4.750000
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	5.384615
2	@joerogan @Spotify Great interview!	8.000000
3	@gtera27 Doge is underestimated	7.000000
4	@teslacn Congratulations Tesla China for amazi...	5.176471
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	6.142857
6	Frodo was the underdoge,\nAll thought he would...	5.928571
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	6.833333
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	7.181818
9	The most entertaining outcome is the most likely	5.125000

In [7]:

```
#number of stop words
import nltk
nltk.download('stopwords')

stop = stopwords.words('english')

data['stopwords'] = data['Text'].apply(lambda x: len([x for x in x.split() if x in stop]))
data[['Text','stopwords']].head(10)
```

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

Out[7]:

	Text	stopwords
0	@kunalb11 I m an alien	1
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	4
2	@joerogan @Spotify Great interview!	0
3	@gtera27 Doge is underestimated	1
4	@teslacn Congratulations Tesla China for amazi...	5
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	2
6	Frodo was the underdoge,\nAll thought he would...	5
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	0
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	2
9	The most entertaining outcome is the most likely	4

In [8]:

```
#number of special characters
data['hastags'] = data['Text'].apply(lambda x: len([x for x in x.split() if x.startswith('@
data[['Text','hastags']].head(10)
```

Out[8]:

	Text	hastags
0	@kunalb11 I m an alien	1
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	1
2	@joerogan @Spotify Great interview!	2
3	@gtera27 Doge is underestimated	1
4	@teslacn Congratulations Tesla China for amazi...	1
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	0
6	Frodo was the underdoge,\nAll thought he would...	0
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	3
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	2
9	The most entertaining outcome is the most likely	0

In [9]:

```
# no of numerical values
data['numerics'] = data['Text'].apply(lambda x: len([x for x in x.split() if x.isdigit()]))
data[['Text','numerics']].head(10)
```

Out[9]:

	Text	numerics
0	@kunalb11 I m an alien	0
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	0
2	@joerogan @Spotify Great interview!	0
3	@gtera27 Doge is underestimated	0
4	@teslacn Congratulations Tesla China for amazi...	0
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	0
6	Frodo was the underdoge,\nAll thought he would...	0
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	0
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	0
9	The most entertaining outcome is the most likely	0

In [10]:

```
data['upper'] = data['Text'].apply(lambda x: len([x for x in x.split() if x.isupper()]))
data[['Text','upper']].head(10)
```

Out[10]:

	Text	upper
0	@kunalb11 I m an alien	0
1	@ID_AA_Carmack Ray tracing on Cyberpunk with H...	1
2	@joerogan @Spotify Great interview!	0
3	@gtera27 Doge is underestimated	0
4	@teslacn Congratulations Tesla China for amazi...	0
5	Happy New Year of the Ox! https://t.co/9WFKMYu2oj	0
6	Frodo was the underdoge,\nAll thought he would...	0
7	@OwenSparks_ @flcnhvy @anonyx10 Haha thanks :)	0
8	@flcnhvy @anonyx10 Indeed! Tweets definitely d...	0
9	The most entertaining outcome is the most likely	0

In [11]:

```
data['Text'] = data['Text'].apply(lambda x: " ".join(x.lower() for x in x.split()))
data['Text'].head()
```

Out[11]:

```
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...
Name: Text, dtype: object
```

In [12]:

```
#removing punctuation
data['Text'] = data['Text'].str.replace('[^\w\s]','')
data['Text'].head()
```

```
C:\Users\SOWMYA~1\AppData\Local\Temp/ipykernel_12696/3292434683.py:2: Future
Warning: The default value of regex will change from True to False in a futu
re version.
  data['Text'] = data['Text'].str.replace('[^\w\s]','')
```

Out[12]:

```
0                               kunalb11 im an alien
1    id_aa_carmack ray tracing on cyberpunk with hd...
2                    joerogan spotify great interview
3                      gtera27 doge is underestimated
4    teslacn congratulations tesla china for amazin...
Name: Text, dtype: object
```

In [13]:

```
#removing stop words
stop = stopwords.words('english')
data['Text'] = data['Text'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
data['Text'].head()
```

Out[13]:

```
0                                    kunalb11 im alien
1    id_aa_carmack ray tracing cyberpunk hdr nextle...
2                      joerogan spotify great interview
3                           gtera27 doge underestimated
4    teslacn congratulations tesla china amazing ex...
Name: Text, dtype: object
```

In [14]:

```
#removing common words
freq = pd.Series(' '.join(data['Text']).split()).value_counts()[:10]
freq
```

Out[14]:

```
spacex           239
amp              218
tesla            166
erdayastronaut   142
rt               127
ppathole         123
flcnhvy          114
yes               86
great             76
teslaownerssv     73
dtype: int64
```

In [15]:

```
freq = list(freq.index)
data['Text'] = data['Text'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))
data['Text'].head()
```

Out[15]:

```
0                                    kunalb11 im alien
1    id_aa_carmack ray tracing cyberpunk hdr nextle...
2                            joerogan spotify interview
3                           gtera27 doge underestimated
4    teslacn congratulations china amazing executio...
Name: Text, dtype: object
```

In [16]:

```
#removing rare words
freq = pd.Series(' '.join(data['Text']).split()).value_counts()[-10:]
freq
```

Out[16]:

```
nyquil               1
musk                 1
negati               1
httpstco6ohta09s5l   1
carousel             1
joeingeneral         1
andrewbogut          1
typical              1
unusual              1
altho                1
dtype: int64
```

In [17]:

```
freq = list(freq.index)
data['Text'] = data['Text'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))
data['Text'].head()
```

Out[17]:

```
0                                    kunalb11 im alien
1    id_aa_carmack ray tracing cyberpunk hdr nextle...
2                           joerogan spotify interview
3                          gtera27 doge underestimated
4    teslacn congratulations china amazing executio...
Name: Text, dtype: object
```

In [18]:

```
data['Text'][:5].apply(lambda x: str(TextBlob(x).correct()))
```

Out[18]:

```
0                                    kunalb11 in alien
1    id_aa_carmack ray tracing cyberpunk her nextle...
2                           joerogan specify interview
3                          gtera27 done underestimated
4    teslacn congratulations china amazing executio...
Name: Text, dtype: object
```

In [19]:

```
import nltk
nltk.download('punkt')

TextBlob(data['Text'][1]).words
```

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

Out[19]:

```
WordList(['id_aa_carmack', 'ray', 'tracing', 'cyberpunk', 'hdr', 'nextleve
l', 'tried'])
```

In [20]:

```
from nltk.stem import PorterStemmer
st = PorterStemmer()
data['Text'][:5].apply(lambda x: " ".join([st.stem(word) for word in x.split()]))
```

Out[20]:

```
0                                   kunalb11 im alien
1    id_aa_carmack ray trace cyberpunk hdr nextleve...
2                           joerogan spotifi interview
3                              gtera27 doge underestim
4    teslacn congratul china amaz execut last year ...
Name: Text, dtype: object
```

In [21]:

```
from textblob import Word

import nltk
nltk.download('wordnet')

data['Text'] = data['Text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.
data['Text'].head()
```

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

Out[21]:

```
0                                   kunalb11 im alien
1    id_aa_carmack ray tracing cyberpunk hdr nextle...
2                           joerogan spotify interview
3                         gtera27 doge underestimated
4    teslacn congratulation china amazing execution...
Name: Text, dtype: object
```

In [22]:

```
TextBlob(data['Text'][0]).ngrams(2)
```

Out[22]:

```
[WordList(['kunalb11', 'im']), WordList(['im', 'alien'])]
```

In [23]:

```
tf1 = (data['Text'][1:2]).apply(lambda x: pd.value_counts(x.split(" "))).sum(axis = 0).rese
tf1.columns = ['words','tf']
tf1
```

Out[23]:

	words	tf
0	id_aa_carmack	1
1	ray	1
2	tracing	1
3	cyberpunk	1
4	hdr	1
5	nextlevel	1
6	tried	1

In [24]:

```
for i,word in enumerate(tf1['words']):
  tf1.loc[i, 'idf'] = np.log(data.shape[0]/(len(data[data['Text'].str.contains(word)])))

tf1
```

Out[24]:

	words	tf	idf
0	id_aa_carmack	1	4.166415
1	ray	1	5.035453
2	tracing	1	7.600402
3	cyberpunk	1	5.115496
4	hdr	1	6.907255
5	nextlevel	1	6.907255
6	tried	1	5.808643

Term Frequency-Inverse Document Frequency (TF-IDF)

In [25]:

```
tf1['tfidf'] = tf1['tf'] * tf1['idf']
tf1
```

Out[25]:

	words	tf	idf	tfidf
0	id_aa_carmack	1	4.166415	4.166415
1	ray	1	5.035453	5.035453
2	tracing	1	7.600402	7.600402
3	cyberpunk	1	5.115496	5.115496
4	hdr	1	6.907255	6.907255
5	nextlevel	1	6.907255	6.907255
6	tried	1	5.808643	5.808643

In [26]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(max_features=1000, lowercase=True, analyzer='word',
 stop_words= 'english',ngram_range=(1,1))
vect = tfidf.fit_transform(data['Text'])
vect
```

Out[26]:

```
<1999x1000 sparse matrix of type '<class 'numpy.float64'>'
        with 7374 stored elements in Compressed Sparse Row format>
```

Bag of Words

In [27]:

```
from sklearn.feature_extraction.text import CountVectorizer
bow = CountVectorizer(max_features=1000, lowercase=True, ngram_range=(1,1),analyzer = "word
data_bow = bow.fit_transform(data['Text'])
data_bow
```

Out[27]:

```
<1999x1000 sparse matrix of type '<class 'numpy.int64'>'
        with 8020 stored elements in Compressed Sparse Row format>
```

Sentiment Analysis

In [28]:

```
data['Text'][:5].apply(lambda x: TextBlob(x).sentiment)
```

Out[28]:

```
0                                  (-0.25, 0.75)
1                                     (0.0, 0.0)
2                                     (0.0, 0.0)
3                                     (0.0, 0.0)
4    (0.20000000000000004, 0.32222222222222224)
Name: Text, dtype: object
```

In [29]:

```
data['sentiment'] = data['Text'].apply(lambda x: TextBlob(x).sentiment[0] )
data[['Text','sentiment']].head(10)
```

Out[29]:

	Text	sentiment
0	kunalb11 im alien	-0.250000
1	id_aa_carmack ray tracing cyberpunk hdr nextle...	0.000000
2	joerogan spotify interview	0.000000
3	gtera27 doge underestimated	0.000000
4	teslacn congratulation china amazing execution...	0.200000
5	happy new year ox httpstco9wfkmyu2oj	0.468182
6	frodo underdoge thought would fail httpstcozgx...	-0.500000
7	owensparks_ anonyx10 haha thanks	0.200000
8	anonyx10 indeed tweet definitely represent rea...	0.000000
9	entertaining outcome likely	0.250000

In [ ]:

```

```