In [1]:
```
#Spashtunyar Week 3 Text Homework
```

In [2]:
```
import pandas as pd
```

In [3]:
```
TrainData = pd.read_csv('labeledTrainData.tsv', sep='\t')
```

In [4]:
```
#Part 1 Q1 setting up the data set
TrainData.head()
```

Out[4]:

	id	sentiment	review
0	5814_8	1	With all this stuff going down at the moment w...
1	2381_9	1	\The Classic War of the Worlds\" by Timothy Hi...
2	7759_3	0	The film starts with a manager (Nicholas Bell)...
3	3630_4	0	It must be assumed that those who praised this...
4	9495_8	1	Superbly trashy and wondrously unpretentious 8...

In [5]:
```
#understanding dataset
TrainData.shape
```

Out[5]:
```
(25000, 3)
```

In [6]:
```
TrainData.size
```

Out[6]:
```
75000
```

In [7]:
```
TrainData['sentiment'].value_counts()
```

Out[7]:
```
1    12500
0    12500
Name: sentiment, dtype: int64
```

In [8]:
```
#we only have scores of 0 and 1 for our sentiment scores it appears
#there are 12500 good and 12500 bad reviews within this data set, it is split down the
```

In [9]:
```
pip install -U textblob
```

```
Requirement already satisfied: textblob in c:\users\spashtunyar\anaconda3\lib\site-pa
ckages (0.17.1)
Requirement already satisfied: nltk>=3.1 in c:\users\spashtunyar\anaconda3\lib\site-p
ackages (from textblob) (3.7)
Requirement already satisfied: joblib in c:\users\spashtunyar\anaconda3\lib\site-pack
ages (from nltk>=3.1->textblob) (1.1.0)
Requirement already satisfied: tqdm in c:\users\spashtunyar\anaconda3\lib\site-packag
es (from nltk>=3.1->textblob) (4.64.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\spashtunyar\anaconda3\lib
\site-packages (from nltk>=3.1->textblob) (2022.3.15)
Requirement already satisfied: click in c:\users\spashtunyar\anaconda3\lib\site-packa
ges (from nltk>=3.1->textblob) (8.0.4)
Requirement already satisfied: colorama in c:\users\spashtunyar\anaconda3\lib\site-pa
ckages (from click->nltk>=3.1->textblob) (0.4.4)
Note: you may need to restart the kernel to use updated packages.
```

In [10]:
```
import nltk
```

In [11]:
```
!pip3 install textblob
```

```
Requirement already satisfied: textblob in c:\users\spashtunyar\anaconda3\lib\site-pa
ckages (0.17.1)
Requirement already satisfied: nltk>=3.1 in c:\users\spashtunyar\anaconda3\lib\site-p
ackages (from textblob) (3.7)
Requirement already satisfied: joblib in c:\users\spashtunyar\anaconda3\lib\site-pack
ages (from nltk>=3.1->textblob) (1.1.0)
Requirement already satisfied: click in c:\users\spashtunyar\anaconda3\lib\site-packa
ges (from nltk>=3.1->textblob) (8.0.4)
Requirement already satisfied: tqdm in c:\users\spashtunyar\anaconda3\lib\site-packag
es (from nltk>=3.1->textblob) (4.64.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\spashtunyar\anaconda3\lib
\site-packages (from nltk>=3.1->textblob) (2022.3.15)
Requirement already satisfied: colorama in c:\users\spashtunyar\anaconda3\lib\site-pa
ckages (from click->nltk>=3.1->textblob) (0.4.4)
```

In [12]:
```
from textblob import TextBlob
```

In [13]:
```
text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating flesh on contact.
Snide comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the grey goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''

blob = TextBlob(text)
blob.tags           # [('The', 'DT'), ('titular', 'JJ'),
                    #  ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases   # WordList(['titular threat', 'blob',
                    #            'ultimate movie monster',
                    #            'amoeba-like mass', ...])

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
```

```
0.06000000000000001
-0.3416666666666673
```

In [14]:
```
#Textblob is installed and tested above using the sample code provided from thier pack
#It is showing the correct response so we can believe the installation occured success
```

In [15]:
```
#getting sentiment analysis
TrainData['sentiment'] = TrainData['review'].apply(lambda tweet: TextBlob(tweet).senti
```

In [16]:
```
#I have two values and want to split them
TrainData['sentiment'].head
```

Out[16]:
```
<bound method NDFrame.head of 0          (0.001276742581090417, 0.6067460317460317)
1            (0.2563492063492064, 0.5311111111111111)
2          (-0.05394123606889564, 0.5629331306990881)
3            (0.1347530864197531, 0.4929012345679012)
4        (-0.024841720779220786, 0.45981782106782115)
                              ...
24995       (0.10208333333333333, 0.5428571428571428)
24996       (0.09081262939958591, 0.4623706004140787)
24997      (0.14525641025641026, 0.48410256410256414)
24998                   (0.065625, 0.5045138888888889)
24999        (0.2392948717948718, 0.7358974358974358)
Name: sentiment, Length: 25000, dtype: object>
```

In [17]:
```
#subjectivity function
def getSubjectivity(text):
   return TextBlob(text).sentiment.subjectivity
```

In [18]:
```
 #Create a function to get the polarity
 def getPolarity(text):
   return TextBlob(text).sentiment.polarity
```

In [19]:
```
TrainData['TextBlob_Subjectivity'] = TrainData['review'].apply(getSubjectivity)
```

In [20]:
```
TrainData['TextBlob_Polarity'] = TrainData['review'].apply(getPolarity)
```

In [21]:
```
#Validating that I was able to split the two
TrainData.head
```

Out[21]:
```
<bound method NDFrame.head of            id                                      sent
iment  \
0       5814_8    (0.001276742581090417, 0.6067460317460317)
1       2381_9      (0.2563492063492064, 0.5311111111111111)
2       7759_3    (-0.05394123606889564, 0.5629331306990881)
3       3630_4      (0.1347530864197531, 0.4929012345679012)
4       9495_8  (-0.024841720779220786, 0.45981782106782115)
...        ...                                          ...
24995   3453_3     (0.10208333333333333, 0.5428571428571428)
24996   5064_1     (0.09081262939958591, 0.4623706004140787)
24997  10905_3    (0.14525641025641026, 0.48410256410256414)
24998  10194_3                (0.065625, 0.5045138888888889)
24999   8478_8      (0.2392948717948718, 0.7358974358974358)

                                                  review  \
0      With all this stuff going down at the moment w...
1      \The Classic War of the Worlds\" by Timothy Hi...
2      The film starts with a manager (Nicholas Bell)...
3      It must be assumed that those who praised this...
4      Superbly trashy and wondrously unpretentious 8...
...                                                  ...
24995  It seems like more consideration has gone into...
24996  I don't believe they made this film. Completel...
24997  Guy is a loser. Can't get girls, needs to buil...
24998  This 30 minute documentary Buñuel made in the ...
24999  I saw this movie as a child and it broke my he...

       TextBlob_Subjectivity  TextBlob_Polarity
0                   0.606746           0.001277
1                   0.531111           0.256349
2                   0.562933          -0.053941
3                   0.492901           0.134753
4                   0.459818          -0.024842
...                      ...                ...
24995               0.542857           0.102083
24996               0.462371           0.090813
24997               0.484103           0.145256
24998               0.504514           0.065625
24999               0.735897           0.239295

[25000 rows x 5 columns]>
```

In [22]:
```python
#Moving to a negative or positive analysis
def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
```

In [23]:
```python
#positive or negative polarity views
TrainData['TextBlob_Analysis'] = TrainData['TextBlob_Polarity'].apply(getAnalysis)
```

In [24]:
```python
#we are able to get the counts of wether the value is positive or negative
TrainData['TextBlob_Analysis'].value_counts()
```

Out[24]:
```
Positive    19000
Negative     5983
Neutral        17
Name: TextBlob_Analysis, dtype: int64
```

In [25]:
```
print(TrainData['review'].iloc[0])
```

```
With all this stuff going down at the moment with MJ i've started listening to his mu
sic, watching the odd documentary here and there, watched The Wiz and watched Moonwal
ker again. Maybe i just want to get a certain insight into this guy who i thought was
really cool in the eighties just to maybe make up my mind whether he is guilty or inn
ocent. Moonwalker is part biography, part feature film which i remember going to see
at the cinema when it was originally released. Some of it has subtle messages about M
J's feeling towards the press and also the obvious message of drugs are bad m'kay.<br
/><br />Visually impressive but of course this is all about Michael Jackson so unless
you remotely like MJ in anyway then you are going to hate this and find it boring. So
me may call MJ an egotist for consenting to the making of this movie BUT MJ and most
of his fans would say that he made it for the fans which if true is really nice of hi
m.<br /><br />The actual feature film bit when it finally starts is only on for 20 mi
nutes or so excluding the Smooth Criminal sequence and Joe Pesci is convincing as a p
sychopathic all powerful drug lord. Why he wants MJ dead so bad is beyond me. Because
MJ overheard his plans? Nah, Joe Pesci's character ranted that he wanted people to kn
ow it is he who is supplying drugs etc so i dunno, maybe he just hates MJ's music.<br
/><br />Lots of cool things in this like MJ turning into a car and a robot and the wh
ole Speed Demon sequence. Also, the director must have had the patience of a saint wh
en it came to filming the kiddy Bad sequence as usually directors hate working with o
ne kid let alone a whole bunch of them performing a complex dance scene.<br /><br />B
ottom line, this movie is for people who like MJ on one level or another (which i thi
nk is most people). If not, then stay away. It does try and give off a wholesome mess
age and ironically MJ's bestest buddy in this movie is a girl! Michael Jackson is tru
ly one of the most talented people ever to grace this planet but is he guilty? Well,
with all the attention i've gave this subject....hmmm well i don't know because peopl
e can be different behind closed doors, i know this for a fact. He is either an extre
mely nice but stupid guy or one of the most sickest liars. I hope he is not the latte
r.
```

In [26]:
```
#above validation looks positive and the score is positive, also reading this takes wo
print(TrainData.iloc[0])
```

```
id                                                          5814_8
sentiment                  (0.001276742581090417, 0.6067460317460317)
review                  With all this stuff going down at the moment w...
TextBlob_Subjectivity                                        0.606746
TextBlob_Polarity                                            0.001277
TextBlob_Analysis                                            Positive
Name: 0, dtype: object
```

In [27]:
```
print(TrainData['review'].iloc[750])
```

```
If this is the first of the \Nemesis\" films that you have seen, then I strongly urge
you to proceed no further. The sequels to \"Nebula\" prove to be no better...hard to
believe considering this entry is bottom-of-the-barrel. This movie tries, but it's ju
st not worth your time, folks. Take a nap instead."
```

In [28]:
```
#This one appears negative but gets a marginal passing score
print(TrainData.iloc[750])
```

```
id                                                          3796_1
sentiment                  (0.08666666666666667, 0.4133333333333334)
review                 If this is the first of the \Nemesis\" films t...
TextBlob_Subjectivity                                       0.413333
TextBlob_Polarity                                           0.086667
TextBlob_Analysis                                           Positive
Name: 750, dtype: object
```

In [29]:
```
print(TrainData['review'].iloc[1527])
```

```
Cary Grant, Douglas Fairbanks Jr. and Victor McLaglen are three soldiers in 19th Cent
ury India who, with the help of a water boy (Sam Jaffe) rid the area of the murderous
thuggee cult. The chemistry between the actors helps make this one of the most entert
aining movies of all time. Sam Jaffe is exceptional as the outcast water boy who is m
istreated by all and still wants to be accepted as a soldier in the company. Loosely
based on Rudyard Kipling's poem. A must see by anyone who enjoys this type of movie.
```

In [30]:
```
#This is very positive and appears very positive as well
print(TrainData.iloc[1527])
```

```
id                                                         5418_10
sentiment                  (0.3579487179487179, 0.5138461538461538)
review                 Cary Grant, Douglas Fairbanks Jr. and Victor M...
TextBlob_Subjectivity                                       0.513846
TextBlob_Polarity                                           0.357949
TextBlob_Analysis                                           Positive
Name: 1527, dtype: object
```

In [31]:
```
#I think after the validation I can see that this is useful because no one has the tim
#It gets close to accurate results but I am still weary. Id say it beats random guessi
```

In [32]:
```
pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in c:\users\spashtunyar\anaconda3\lib\s
ite-packages (3.3.2)Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: requests in c:\users\spashtunyar\anaconda3\lib\site-pa
ckages (from vaderSentiment) (2.27.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\spashtunyar\anaconda3\l
ib\site-packages (from requests->vaderSentiment) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\spashtunyar\anaconda
3\lib\site-packages (from requests->vaderSentiment) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\users\spashtunyar\anaconda3\lib\sit
e-packages (from requests->vaderSentiment) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\spashtunyar\anac
onda3\lib\site-packages (from requests->vaderSentiment) (2.0.4)
```

In [33]:
```
#import vader
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

In [34]:
```
#taking the vader steps from the info site
def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)
```

```
    print("Overall sentiment dictionary is : ", sentiment_dict)
    print("sentence was rated as ", sentiment_dict['neg']*100, "% Negative")
    print("sentence was rated as ", sentiment_dict['neu']*100, "% Neutral")
    print("sentence was rated as ", sentiment_dict['pos']*100, "% Positive")

    print("Sentence Overall Rated As", end = " ")

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        print("Positive")

    elif sentiment_dict['compound'] <= - 0.05 :
        print("Negative")

    else :
        print("Neutral")
```

In [ ]:
```
#Creating new column for Vader
TrainData['VADER_Analysis'] = sentiment_scores(TrainData['review'])
```

In [ ]:
```
#VaderAnalysis
TrainData['VADER_Analysis']
```

In [38]:
```
#Lowercase
TrainData['text_lower'] = TrainData['review'].str.lower()
```

In [39]:
```
#Validation
print(TrainData.iloc[0])
```

```
id                                                          5814_8
sentiment                  (0.001276742581090417, 0.6067460317460317)
review                 With all this stuff going down at the moment w...
TextBlob_Subjectivity                                       0.606746
TextBlob_Polarity                                           0.001277
TextBlob_Analysis                                           Positive
text_lower             with all this stuff going down at the moment w...
Name: 0, dtype: object
```

In [69]:
```
#remove puncation
TrainData["new_column"] = TrainData['text_lower'].str.replace('[^\w\s]','')
```

```
C:\Users\spashtunyar\AppData\Local\Temp\ipykernel_34644\4248447268.py:2: FutureWarnin
g: The default value of regex will change from True to False in a future version.
  TrainData["new_column"] = TrainData['text_lower'].str.replace('[^\w\s]','')
```

In [44]:
```
#validation
print(TrainData['new_column'].iloc[2])
```

```
the film starts with a manager nicholas bell giving welcome investors robert carradin
e to primal park  a secret project mutating a primal animal using fossilized dna like
jurassik park and some scientists resurrect one of natures most fearsome predators th
e sabretooth tiger or smilodon  scientific ambition turns deadly however and when the
high voltage fence is opened the creature escape and begins savagely stalking its pre
y  the human visitors  tourists and scientificmeanwhile some youngsters enter in the
restricted area of the security center and are attacked by a pack of large prehistori
cal animals which are deadlier and bigger  in addition  a security agent stacy haiduk
and her mate brian wimmer fight hardly against the carnivorous smilodons the sabretoo
ths themselves  of course are the real star stars and they are astounding terrifyingl
y though not convincing the giant animals savagely are stalking its prey and the grou
p run afoul and fight against one natures most fearsome predators furthermore a third
sabretooth more dangerous and slow stalks its victimsbr br the movie delivers the goo
ds with lots of blood and gore as beheading hairraising chillsfull of scares when the
sabretooths appear with mediocre special effectsthe story provides exciting and stirr
ing entertainment but it results to be quite boring the giant animals are majority ma
de by computer generator and seem totally lousy middling performances though the play
ers reacting appropriately to becoming foodactors give vigorously physical performanc
es dodging the beasts runningbound and leaps or dangling over walls  and it packs a r
idiculous final deadly scene no for small kids by realisticgory and violent attack sc
enes  other films about sabretooths or smilodon are the following  sabretooth2002by j
ames r hickox with vanessa angel david keith and john rhys davies and the much better
10000 bc2006 by roland emmerich with with steven strait cliff curtis and camilla bell
e this motion picture filled with bloody moments is badly directed by george miller a
nd with no originality because takes too many elements from previous films miller is
an australian director usually working for television tidal wave journey to the cente
r of the earth and many others and occasionally for cinema  the man from snowy river
zeus and roxannerobinson crusoe  rating  below average bottom of barrel
```

In [50]:
```
from nltk.tokenize import word_tokenize
```

In [45]:
```
from nltk.corpus import stopwords
```

In [46]:
```
import nltk
```

In [48]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\spashtunyar\Anaconda3\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[48]:
```
True
```

In [52]:
```
stop_words = stopwords.words('english')
```

In [53]:
```
print(stop_words)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

In [54]:
```
TrainData['removedstops'] = TrainData['new_column'].apply(lambda x: ' '.join([word for
```

In [57]:
```
#new column with removed stop words
TrainData['removedstops']
```

Out[57]:
```
0        stuff going moment mj ive started listening mu...
1        classic war worlds timothy hines entertaining ...
2        film starts manager nicholas bell giving welco...
3        must assumed praised film greatest filmed oper...
4        superbly trashy wondrously unpretentious 80s e...
                               ...
24995    seems like consideration gone imdb reviews fil...
24996    dont believe made film completely unnecessary ...
24997    guy loser cant get girls needs build picked st...
24998    30 minute documentary buñuel made early 1930s ...
24999    saw movie child broke heart story unfinished e...
Name: removedstops, Length: 25000, dtype: object
```

In [58]:
```
from nltk.stem.porter import PorterStemmer
```

In [64]:
```
porter = PorterStemmer()
```

In [65]:
```
def stem_sentences(sentence):
    tokens = sentence.split()
    stemmed_tokens = [porter.stem(token) for token in tokens]
    return ' '.join(stemmed_tokens)
```

In [66]:
```
TrainData['StemPort'] = TrainData['removedstops'].apply(stem_sentences)
```

In [68]:
```
#stemmed data set
TrainData['StemPort']
```

Out[68]:
```
0        stuff go moment mj ive start listen music watc...
1        classic war world timothi hine entertain film ...
2        film start manag nichola bell give welcom inve...
3        must assum prais film greatest film opera ever...
4        superbl trashi wondrous unpretenti 80 exploit ...
                              ...
24995    seem like consider gone imdb review film went ...
24996    dont believ made film complet unnecessari firs...
24997    guy loser cant get girl need build pick strong...
24998    30 minut documentari buñuel made earli 1930 on...
24999    saw movi child broke heart stori unfinish end ...
Name: StemPort, Length: 25000, dtype: object
```

In [70]:
```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
```

In [71]:
```
count = CountVectorizer()
```

In [74]:
```
bag_of_words = count.fit_transform(TrainData['StemPort'])
```

In [75]:
```
bag_of_words
```

Out[75]:
```
<25000x92532 sparse matrix of type '<class 'numpy.int64'>'
        with 2439335 stored elements in Compressed Sparse Row format>
```

In [76]:
```
#the bag of words shows 25000 rows which is the same as the original data frame
```

In [77]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [78]:
```
tfidf = TfidfVectorizer()
```

In [81]:
```
feature_matrix = tfidf.fit_transform(TrainData['StemPort'])
```

In [82]:
```
feature_matrix
```

Out[82]:
```
<25000x92532 sparse matrix of type '<class 'numpy.float64'>'
        with 2439335 stored elements in Compressed Sparse Row format>
```

In [83]:
```
#feature matrix shows the same number of rows
```

In [ ]:
```

```