In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
ml = pd.read_csv("C:/Users/saiku/OneDrive/ML Projects/model_building.csv")
```

In [4]:

```
ml.head()
```

Out[4]:

	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominato
0	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
1	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	
2	B000UA0QIQ	A395BORC6FGVXV	Karl	3	
3	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	
4	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapennything	0	

Sorting the dataframe according to 'Time' feature

In [5]:
```
ml.sort_values(['Time'], ascending=True, inplace=True)
ml.head()
```

Out[5]:

	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominat
150522	0006641040	ACITT7DI6IDDL	shari zychinski	0	
150499	0006641040	AJ46FKXOVC7NR	Nicholas A Mesiano	2	
451854	B00004CXX9	AIUWLEQ1ADEG5	Elizabeth Medina	0	
374357	B00004CI84	A344SMIA5JECGM	Vincent P. Ross	1	
451876	B00004CXX9	A344SMIA5JECGM	Vincent P. Ross	1	

Dropping the unwanted columns from our data frame.

In [6]:
```
ml.drop(['ProductId', 'ProfileName','HelpfulnessNumerator','HelpfulnessDenominatc
```

In [7]:
```
ml.head()
```

Out[7]:

	UserId	Score	ReviewSummary	ReviewText
150522	ACITT7DI6IDDL	5	EVERY book is educational	this witty little book makes my son laugh at ...
150499	AJ46FKXOVC7NR	5	This whole series is great way to spend time ...	I can remember seeing the show when it aired ...
451854	AIUWLEQ1ADEG5	5	Entertainingl Funny!	Beetlejuice is a well written movie ..... eve...
374357	A344SMIA5JECGM	5	A modern day fairy tale	A twist of rumplestiskin captured on film, st...
451876	A344SMIA5JECGM	5	A modern day fairy tale	A twist of rumplestiskin captured on film, st...

Score less and greater than 3 equal to negative and postive class

In [8]:
```
score = []
for i in ml['Score']:
    if i < 3:
        score.append('0')
    else:
        score.append('1')
ml['Score'] = score
```

In [9]:
```
ml.head()
```

Out[9]:

	UserId	Score	ReviewSummary	ReviewText
150522	ACITT7DI6IDDL	1	EVERY book is educational	this witty little book makes my son laugh at ...
150499	AJ46FKXOVC7NR	1	This whole series is great way to spend time ...	I can remember seeing the show when it aired ...
451854	AIUWLEQ1ADEG5	1	EntertainingI Funny!	Beetlejuice is a well written movie ..... eve...
374357	A344SMIA5JECGM	1	A modern day fairy tale	A twist of rumplestiskin captured on film, st...
451876	A344SMIA5JECGM	1	A modern day fairy tale	A twist of rumplestiskin captured on film, st...

Model Building

In [10]:
```
from sklearn.model_selection import train_test_split
```

In [11]:
```
total_size=len(ml)
train_size=int(0.70*total_size)
train=ml.head(train_size)
test=ml.tail(total_size - train_size)
```

In [12]:
```
train.Score.value_counts()
```

Out[12]:
```
1    342750
0     55167
Name: Score, dtype: int64
```

Removing all rows where 'Score' is equal to 3

In [13]:
```
train = train[train.Score != 3]
test = test[test.Score != 3]
```

In [14]:
```
print(train.shape)
print(test.shape)
```

```
(397917, 4)
(170536, 4)
```

In [15]:
```
train['Score'].value_counts()
```

Out[15]:
```
1    342750
0     55167
Name: Score, dtype: int64
```

In [16]:
```
test['Score'].value_counts()
```

Out[16]:
```
1    143666
0     26870
Name: Score, dtype: int64
```

Text Preprocessing

In [17]:
```
lst_text = train['ReviewText'].tolist()
lst_summary = train['ReviewSummary'].tolist()
```

In [18]:
```
lst_text
```

```
hole once they are caught. I hope you find this product as easy to  use as I
did, Good luck.',
 ' This are so much easier to use than the Wilson paste colors.  Colors are
vibrant, and do not taint the frosting like some colors can.  These are  simp
le to use, and do not make a mess.  My only complaint is that I did not  find
these years ago.  This is a must have if you decorate often!',
 ' These are easy to use, they do not make a mess, and offer vibrant colors.
They do not taint what you are decorting as some colors can.  I would  highly
recommend these to anyone to likes to decorate.',
 " This is such a great film, I don't even know how to sum it up. First of  a
ll, it is completely original and it is unlike any film I have ever seen  bef
ore. Second of all, it's a great comedy with kind of a spooky, weird  feel to
it, which is something all of Tim Burton's films have. The look of  the film

is probably what I like the best. Art Director Bo Welch and Tim  Burton show
us a world unlike anything seen in a movie.  This is a great  film, and I wou
ld recommend it to anyone looking for an enjoyable,  entertaining film that i
s original and inventive.",
 " This is such a great film, I don't even know how to sum it up. First of  a
ll, it is completely original and it is unlike any film I have ever seen  bef
ore. Second of all, it's a great comedy with kind of a spooky, weird  feel to
```

In [19]:
```
lst_summary
```

```
 ' A Afterlife Success',
 ' Make your own Martha Stewart style cakes and cookies',
 " What's the Catch?",
 ' CASPER IS THE GHOST WITH THE MOST',
 ' Great movie, terrible DVD',
 ' Great movie, terrible DVD',
 ' Great movie, terrible DVD',
 ' Beetlejuice is a greatmovie, but they cheated you on the dvd',
 ' Beetlejuice is a greatmovie, but they cheated you on the dvd',
 ' Beetlejuice is a greatmovie, but they cheated you on the dvd',
 ' Nice, bright colors!',
 ' Beetlejuice! Beetlejuice! Beatlejuice!',
 ' Beetlejuice! Beetlejuice! Beatlejuice!',
 ' It Was a favorite!',
 ' A little piece of heaven.',
 ' A little piece of heaven.',
 ' A little piece of heaven.',

 ' Beetlejuice - Great Fun for Everyone!',
 ' Beetlejuice - Great Fun for Everyone!',
 ' Beetlejuice - Great Fun for Everyone!'
```

In [20]:
```
test_text = test['ReviewText'].tolist()
test_text
```

Out[20]:
```
[" I have to say i bought this item with some apprehension but when it arrive
d I was happy. The item doesn't come in any packaging except a plastic bag, h
mmm, bit odd, but its not made of plastic itself its actually metal, very stu
rdy. Now my main concern was that I ordered it and my wife said how does it f
ix to the units, I replied by sticky pads. Oh she said you do know we are rep
lacing our kitchen units soon, oh I said, well have no fear it came with scre
ws as well, thank you. It fits up pretty easy although you do need three hand
s just because of its shape, design and you need to hold it underneath a kitc
hen unit. But it works perfectly and does exactly what it says. Very, very pl
eased. Highly recommend this for your K-cups.",
 ' I wonder if someone started listening to the reviews on here of dented can
s? Mine was shipped in a box with air-filled bags inside of another box with
a ton more air-filled bags. No dents.<br /><br />I purchased this oil for two
reasons. One, homemade healthy mayonnaise. Two, high smoking point. I used it
last night to deep fry some soft corn tortillas to make hard taco shells. Wor
ked great. No gross after taste like you get with canola. I also deep fried s
ome flour tortillas for my husband (I have a wheat allergy), and he LOVED the
m. The great part is I can deep fry and not have to feel guilty about serving
up rancid, omega-6 filled food. Very happy.',
```

Converting to Lower-case

In [21]:
```
lst_text = [str(item).lower() for item in lst_text]
lst_summary = [str(item).lower() for item in lst_summary]
```

In [22]:
```
test_text = [str(item).lower() for item in test_text]
```

Removing HTML Tags from strings

Removing HTML Tags from strings

In [23]:
```
import re
def striphtml(data):
    p = re.compile(r'<.*?>')
    return p.sub('', data)

for i in range(len(lst_text)):
    lst_text[i] = striphtml(lst_text[i])
    lst_summary[i] = striphtml(lst_summary[i])
```

In [24]:
```
for i in range(len(test_text)):
    test_text[i] = striphtml(test_text[i])
```

In [25]:
```
lst_text[0:5]
```

Out[25]:
```
[" this witty little book makes my son laugh at loud. i recite it in the car as
we're driving along and he always can sing the refrain. he's learned about whal
es, india, drooping roses",
 " i can remember seeing the show when it aired on television years ago, when i
was a child.  my sister later bought me the lp (which i have to this day,  i'm
thirty something).i used this series of books &amp; songs when i did my  studen
t teaching for preschoolers &amp; turned the whole school on to it.  i am now p
urchasing it on cd, along with the books for my children 5 &amp;  2.  the tradi
tion lives on!",
 ' beetlejuice is a well written movie ..... everything about it is excellent!
from the acting to the special effects you will be delighted you chose to view
this movie.',
 " a twist of rumplestiskin captured on film, starring michael keaton and geena
davis in their prime.  tim burton's masterpiece, rumbles with absurdity, and is
wonderfully paced to the point where there is not a dull  moment.",
 " a twist of rumplestiskin captured on film, starring michael keaton and geena
davis in their prime.  tim burton's masterpiece, rumbles with absurdity, and is
wonderfully paced to the point where there is not a dull  moment."]
```

In [26]:
```
lst_summary[0:6]
```

Out[26]:
```
[' every book is educational',
 ' this whole series is great way to spend time with your child',
 ' entertainingl funny!',
 ' a modern day fairy tale',
 ' a modern day fairy tale',
 ' a modern day fairy tale']
```

Removing Special Characters from strings

In [27]:
```
for i in range(len(lst_text)):
    lst_text[i] = re.sub(r'[^A-Za-z]+', ' ', lst_text[i])
    lst_summary[i] = re.sub(r'[^A-Za-z]+', ' ', lst_summary[i])
```

In [28]:
```
for i in range(len(test_text)):
    test_text[i] = re.sub(r'[^A-Za-z]+', ' ', test_text[i])
```

In [29]:
```
lst_text[0:5]
```

Out[29]:
```
[' this witty little book makes my son laugh at loud i recite it in the car as
we re driving along and he always can sing the refrain he s learned about whale
s india drooping roses',
 ' i can remember seeing the show when it aired on television years ago when i
was a child my sister later bought me the lp which i have to this day i m thirt
y something i used this series of books amp songs when i did my student teachin
g for preschoolers amp turned the whole school on to it i am now purchasing it
on cd along with the books for my children amp the tradition lives on ',
 ' beetlejuice is a well written movie everything about it is excellent from th
e acting to the special effects you will be delighted you chose to view this mo
vie ',
 ' a twist of rumplestiskin captured on film starring michael keaton and geena
davis in their prime tim burton s masterpiece rumbles with absurdity and is won
derfully paced to the point where there is not a dull moment ',
 ' a twist of rumplestiskin captured on film starring michael keaton and geena
davis in their prime tim burton s masterpiece rumbles with absurdity and is won
derfully paced to the point where there is not a dull moment ']
```

Removing Stop Words

In [30]:
```
import nltk
nltk.download('stopwords')
```

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

Out[30]:
```
True
```

In [31]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

In [32]:
```
%time
stop_words = set(stopwords.words('english'))
for i in range(len(lst_text)):
    text_filtered = []
    summary_filtered = []
    text_word_tokens = []
    summary_word_tokens = []
    text_word_tokens = lst_text[i].split()
    summary_word_tokens = lst_summary[i].split()
    for r in text_word_tokens:
        if not r in stop_words:
            text_filtered.append(r)
    lst_text[i] = ' '.join(text_filtered)
    for r in summary_word_tokens:
        if not r in stop_words:
            summary_filtered.append(r)
    lst_summary[i] = ' '.join(summary_filtered)
```

```
Wall time: 0 ns
```

In [33]:

```
for i in range(len(test_text)):
    text_filtered = []
    text_word_tokens = []
    text_word_tokens = test_text[i].split()
    for r in text_word_tokens:
        if not r in stop_words:
            text_filtered.append(r)
    test_text[i] = ' '.join(text_filtered)
```

In [ ]:

```
```

Stemming

Stem for each word

In [34]:

```
%time
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("english")
for i in range(len(lst_text)):
    text_filtered = []
    summary_filtered = []
    text_word_tokens = []
    summary_word_tokens = []
    text_word_tokens = lst_text[i].split()
    summary_word_tokens = lst_summary[i].split()
    for r in text_word_tokens:
        text_filtered.append(str(stemmer.stem(r)))
    lst_text[i] = ' '.join(text_filtered)
    for r in summary_word_tokens:
        summary_filtered.append(str(stemmer.stem(r)))
    lst_summary[i] = ' '.join(summary_filtered)
```

```
Wall time: 0 ns
```

In [35]:

```
for i in range(len(test_text)):
    text_filtered = []
    text_word_tokens = []
    text_word_tokens = test_text[i].split()
    for r in text_word_tokens:
        if not r in stop_words:
            text_filtered.append(str(stemmer.stem(r)))
    test_text[i] = ' '.join(text_filtered)
```

In [36]:
```
lst_text[0:5]
```

Out[36]:
```
['witti littl book make son laugh loud recit car drive along alway sing refrain
learn whale india droop rose',
 'rememb see show air televis year ago child sister later bought lp day thirti
someth use seri book amp song student teach preschool amp turn whole school pur
chas cd along book children amp tradit live',
 'beetlejuic well written movi everyth excel act special effect delight chose v
iew movi',
 'twist rumplestiskin captur film star michael keaton geena davi prime tim burt
on masterpiec rumbl absurd wonder pace point dull moment',
 'twist rumplestiskin captur film star michael keaton geena davi prime tim burt
on masterpiec rumbl absurd wonder pace point dull moment']
```

In [37]:
```
test_text[0:5]
```

Out[37]:
```
['say bought item apprehens arriv happi item come packag except plastic bag hmm
m bit odd made plastic actual metal sturdi main concern order wife said fix uni
t repli sticki pad oh said know replac kitchen unit soon oh said well fear came
screw well thank fit pretti easi although need three hand shape design need hol
d underneath kitchen unit work perfect exact say pleas high recommend k cup',
 'wonder someon start listen review dent can mine ship box air fill bag insid a
noth box ton air fill bag dent purchas oil two reason one homemad healthi mayon
nais two high smoke point use last night deep fri soft corn tortilla make hard
taco shell work great gross tast like get canola also deep fri flour tortilla h
usband wheat allergi love great part deep fri feel guilti serv rancid omega fil
l food happi',
 'huge fan mrs bridg lemon curd ounc jar pack like keep stock pantri refriger u
se biscuit scone fresh fruit excel qualiti tast',
 'good stuff like drink wuyi oolong tea total relax job sever week drink produc
t notic hungri last year lost approxim lbs count product help lose weight cut b
ack',
 'expect good product product let big time worst macaroni chees ever eaten']
```

Converting Text to Numerical vectors - BOW Representation

In [38]:
```
from sklearn.feature_extraction.text import CountVectorizer
vocab = CountVectorizer()
train_bow = vocab.fit_transform(lst_text)
```

In [39]:
```
train_bow
```

Out[39]:
```
<397917x65770 sparse matrix of type '<class 'numpy.int64'>'
        with 11485842 stored elements in Compressed Sparse Row format>
```

In [40]:
```
X_test_dtm = vocab.transform(test_text)
X_test_dtm
```

Out[40]:
```
<170536x65770 sparse matrix of type '<class 'numpy.int64'>'
        with 5047211 stored elements in Compressed Sparse Row format>
```

Multinomial Naive Bayes

In [41]:

```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
nb.fit(train_bow, train.Score)
```

Out[41]:

```
MultinomialNB()
```

In [42]:

```
y_pred_class_nb = nb.predict(X_test_dtm)
```

In [43]:

```
from sklearn import metrics
metrics.accuracy_score(test.Score, y_pred_class_nb)
```

Out[43]:

```
0.8801484730496787
```

Logistic Regression

In [44]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(train_bow, train.Score)
```

Out[44]:

```
LogisticRegression()
```

In [45]:

```
y_pred_class_logistic = classifier.predict(X_test_dtm)
```

In [46]:

```
metrics.accuracy_score(test.Score, y_pred_class_logistic)
```

Out[46]:

```
0.8962389172960548
```